FingARtips – Gesture Based Direct Manipulation in Augmented Reality

Volkert Buchmann[†]
HIT Lab NZ

Stephen Violich[‡]
University of
Canterbury, NZ

Mark Billinghurst[*]
HIT Lab NZ

Andy Cockburn[§]
University of
Canterbury, NZ

Abstract

This paper presents a technique for natural, fingertip-based interaction with virtual objects in Augmented Reality (AR) environments. We use image processing software and finger- and hand-based fiducial markers to track gestures from the user, stencil buffering to enable the user to see their fingers at all times, and fingertip-based haptic feedback devices to enable the user to feel virtual objects. Unlike previous AR interfaces, this approach allows users to interact with virtual content using natural hand gestures. The paper describes how these techniques were applied in an urban planning interface, and also presents preliminary informal usability results.

CR Categories: H.5.1 [Information Systems]: Multimedia Information Systems – Artificial, augmented, and virtual realities; H.5.2 [Information Systems]: User Interfaces – Haptic I/O; I.3.6 [Computer Graphics]: Methodology and Techniques – Interaction techniques

Keywords: Augmented Reality, gesture interaction, occlusion

1 Introduction

Augmented Reality (AR) interfaces typically involve the overlay of virtual imagery onto the real world. The goal is to blend reality and virtuality in a seamless manner. Although the first AR interfaces were demonstrated nearly 40 years ago [Sutherland 1968], the field has been primarily concerned with "visual augmentations" [Ishii 1997]. While great advances have been made in AR display technologies and tracking techniques, few systems provide tools that let the user interact with and modify AR content. Furthermore, even basic interaction tasks, such as manipulation, copying, annotating, and deleting virtual objects have often been poorly addressed.

Augmented Reality interaction techniques need to be as intuitive as possible to be accepted by end users. They may also need to be customized to cater for different application needs. In the past, a variety of interaction methods have been presented, including the use of tracked objects [Kiyokawa 1999], mouse and keyboard input, and pen and tablet [Schmalstieg 2000]. However, there has been little research on the use of free hand interaction. Since hands are our main means of interaction with objects in real life, AR interfaces should also allow for free hand interaction with virtual objects. This would not only enable natural and intuitive interaction with virtual objects, but it would also help to ease the transition between interaction with real and virtual objects at the same time. In this paper we present techniques for free hand interaction with AR content.

We can distinguish between two different interaction spaces for hand based interaction: near space and far space. Near objects are within arm's reach and can be manipulated intuitively. Far objects are no longer within arm's reach and cannot be manipulated with a hand in real life. There are other range-based AR and Virtual Reality (VR) techniques that are more appropriate for selection and manipulation of far objects, such as ray based selection. The FingARtips project is concerned with near objects that can be manipulated by natural gestures such as grabbing, pressing, dragging and releasing. Currently, the main domain for this type of interaction is a personal workplace or a conference table

One example application where gesture interaction could be applied is for urban planning. This is where architects, city council members and interest groups meet and discuss building alternatives. An AR model of city blocks can easily be modified and animated, and it provides the opportunity for rich interactive experiences, such as immersive visits into the virtual content. A key requirement for this application is fast and easy interaction. In order to communicate the architectural design, interaction has to be natural and intuitive. Users should be able to express their vision by simply dragging and dropping houses just like real boxes. Non-experts such as politicians and ordinary citizens must be able to use the system without lengthy training. Another potential requirement is mobility of the system. Architects should be able to travel to their customers and be able to set up the system easily. Clearly, a system that satisfies these requirements will be applicable in many areas, not just urban planning.

Gesture-based interaction with AR content has two important characteristic problems: incorrect occlusion of the real hand and the lack of haptic feedback. In many AR interfaces, virtual objects are overlaid on top of video of the real world. This means that they will occlude real objects that should appear in front of them. This is especially problematic for hand-based interaction where the relative depth ordering of real and virtual objects is crucial. In real life, we rely heavily on haptic feedback when we manipulate objects with our hands. However, virtual objects cannot provide such feedback. FingARtips uses simple and inexpensive techniques to successfully address these two problems.

FingARtips uses visual tracking and haptic feedback to implement a simple set of gestures which enable the user to directly manipulate virtual objects. These capabilities are demonstrated in an AR urban planning application, in which users manipulate virtual buildings, roads and viewpoints. In the remainder of this paper we first review related work and then describe the

[†] e-mail: oakley.buchmann@hitlabnz.org
[‡] e-mail: ssv10@student.canterbury.ac.nz
[*] e-mail: mark.billinghurst@hitlabnz.org
[§] e-mail: andy@cosc.canterbury.ac.nz

FingARtips system. Finally, we describe the results of our informal usability tests, lessons learned and future work.

2 Related Work

There are a number of previous research projects that have explored hand- and finger-based interaction in Augmented and Virtual Reality interfaces. Their approaches vary widely.

In early immersive Virtual Environments gesture based interaction was common. For example, the NASA VIEW system used a VPL Dataglove [Wise 1990] to enable users to navigate through the environment and select virtual menu options. In the *Rubber Rocks* virtual environment, users could pick up virtual rocks by making a fist gesture and throw and release them with a flat hand gesture [Codella, 1992]. The GIVEN virtual environment (Gesture-driven Interactions in Virtual Environments), used a neural network to recognize up to twenty static and dynamic gestures [Väänänen and Böhm, 1993]. These include pointing gestures for flying, fist gestures for grabbing and other whole hand gesture for releasing objects or returning back to the starting point in the virtual environment. These interfaces and others show that gestures can be an extremely intuitive method for interacting with virtual objects.

Although gesture input has been extensively used in VR interfaces, it is far less common in AR applications. Our work was partially inspired by the Tinmith project [Piekarski 2003], a wearable outdoor AR system. Tinmith uses special gloves for interaction with the system and the environment. Fiducial markers on the user's thumbs are visually tracked by a camera mounted on a head mounted display (HMD). Their system also allows for data input when the gloves are not visible [Thomas and Piekarski 2002]. The gloves are used as the sole input devices for the Tinmith system. But while two-handed interaction is supported, the markers are only used for 3 degree of freedom (DOF) input. The Tinmith system is also explicitly centered on interaction with far away objects and does not explore near space interaction techniques such as grabbing.

Gordon et al. use a dense stereo range device to enable users to use their fingers or pens as pointing devices in AR [Gordon et al. 2002]. Their algorithm takes as input dense depth data and visual images from a real time rangefinder attached to an HMD. The contour of a finger is filtered against the background. Its tip and direction are calculated using a simple and straightforward approach. Unlike Tinmith, there is no requirement for special tracking markers or other enhancements to be added to the user's hands. However, noise in the data means that the system can reliably only be used for 3 DOF position input. An informal evaluation of their system showed that incorrect occlusion of the user's hand caused frustration.

Walairacht et al. describe a system that allows two-handed interaction in AR with a force feedback device [Walairacht et al. 2002]. Four fingers from each hand are connected to three wires each, which are suspended from the edges of a cubic frame and kept tense by motors. The finger position is calculated from the length of the wires, while force feedback is provided by the motors pulling the wires. The system also registers the hands correctly in 3D – including the depth of each finger – resulting in a realistic blend of virtual objects and real hands. Our rendering of the user's fingers was inspired by this system. However, the frame greatly reduces the mobility of the system and confines interaction to a small space within the frame. The wire-based tracking also requires tedious calibration.

Dorfmüller-Ulhaas and Schmalstieg present a finger tracking system that features a marked glove, a stereoscopic computer vision tracking system, and a kinematic 3D model of the human finger [Dorfmüller-Ulhaas and Schmalstieg 2001]. It enables the user to grab, translate, rotate, and release objects in a natural way. The system tracks the position and orientation of the joints of the user's index finger. The user wears a glove, the index finger of which is fitted with an `exoskeleton' of small, retro-reflective balls. These balls are displaced from the joints by a small piece of wire, thereby greatly reducing occlusion of the markers. However, since only the index finger is tracked, no natural grabbing is supported. Also, the hand may be incorrectly occluded by virtual content.

Researchers have found that adding force or haptic feedback significantly increases performance in gesture-based virtual environments. DiZio and Lackner report that in VR, cutaneous contact cues "contribute to perception and adaptation of limb position and force" [DiZio and Lackner 2002]. They found that the accuracy with which participants could repeatedly touch virtual objects improved significantly when haptic feedback was added. They show that haptic and visual information should be combined to learn the position of an object. The findings also suggest that haptic feedback is necessary for exact pointing.

Klatzky and Lederman [2003] propose that vibrotactile stimulation could be the best approach for cutaneous feedback in virtual environments, because it is easy and cheap to generate. Vibration can easily be produced using small electromagnets, and frequency and intensity can easily be manipulated.

A related piece of work, the TactilePointer, described by Hauber, is a 6 DOF pointer with tactile feedback [Hauber 2002]. While this system does not allow for natural interaction such as grabbing, it explores haptic feedback using buzzers, and in a second version also piezoelectric bending actuators. Hauber found that buzzers are a cheap and effective solution for haptic feedback.

In addition to haptic input, occlusion is an important cue for relative depth information [Braunstein et al. 1982]. Cutting found that occlusion in Virtual Environments provided the best depth information over other cues such as relative size, height in the visual field and aerial perspective [Cutting 1997]. Furmanski et al. showed for AR that virtual objects that were rendered as an overlay of a real scene were perceived to be on top of the real objects even though they were fixed on a position behind a real object [Furmanski et al. 2002]. Motion perspective should have been able to reveal the correct depth ordering, but participants gave information from occlusion a higher priority. Livingston et al. evaluated different visual cues for AR depth ordering [Livingston et al. 2003] and concluded that occlusion might be the primary cue for depth ordering.

These results suggest than for effective gesture interaction an interface should support both some form of tactile or haptic feedback, provide occlusion cues and allow multi-fingered input. In the next section we describe the FingARtips system and show how it provides all of these features.

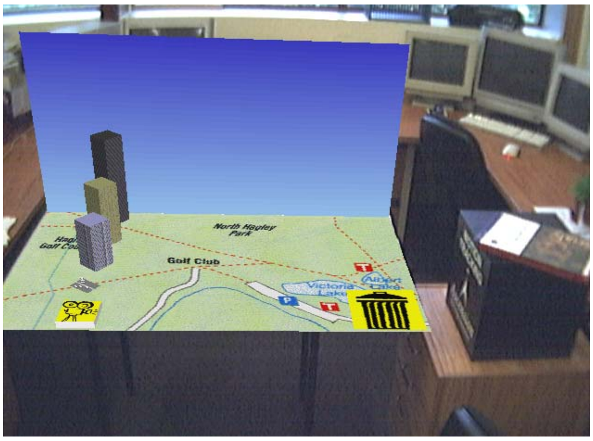

Figure 1: The urban planning workspace.

3 FingARtips

In this section, we give an in-depth description of the FingARtips system, beginning with an overview of an urban planning application that uses the FingARtips gesture input.

3.1 The Urban Planning Workspace

The urban planning workspace was implemented to explore the possibilities of the FingARtips system. Four different types of gesture interaction are implemented: grabbing, pointing, navigating and command gestures. The application does not take any input other than from a single gloved hand. This shows that even simple gesture interaction provides for very versatile input.

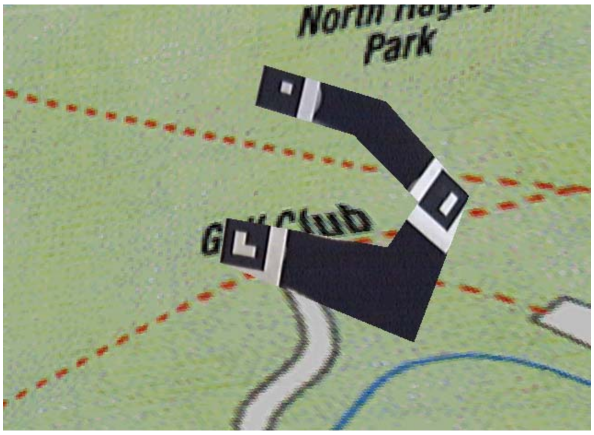

Figure 2: The user's thumb and index finger as they appear in the workspace

The urban planning workspace is based on a map of a park as shown in figure 1. On the left hand side, a menu offers three types of skyscrapers, a street and an immersive view button. On the right hand side, a trash can icon is displayed. A sky texture is seen at the rear of the workspace. The user is able to place buildings, draw streets and switch back and forth to an immersive view anywhere on the map. Figure 2 shows how the user's hand is rendered in the urban planning workspace. The hand model only includes the index finger and thumb.

Such an application could be used to collaboratively discuss and sketch the development of a built-up area. The application comes with an unlimited number of buildings in all sizes. The streets carry animated cars, which further increases the realism of the virtual development area. Finally, the immersive mode allows the users to enter the virtual scene and watch from the eyes of a future inhabitant. These features provide important advantages over a real model.

The glove is the sole input device for the urban planning workspace. The application recognizes a set of gestures such as grabbing, releasing, pointing, pressing and navigating, which we will describe in the remainder of this section.

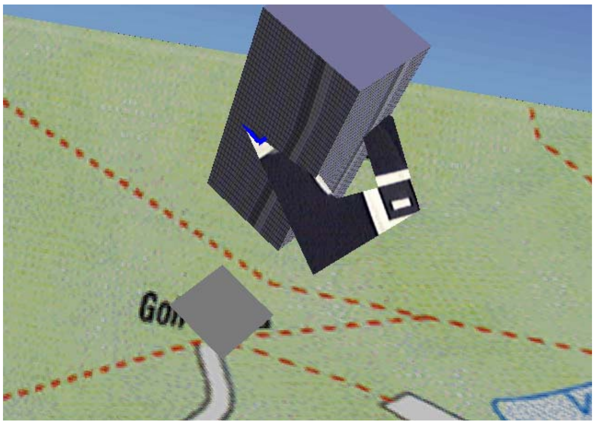

Figure 3: Shadows help the user to judge a lifted building's position and orientation.

3.1.1 Grabbing

Grabbing gestures are used to manipulate the buildings. Using this gesture, buildings can be created, destroyed, translated, rotated and resized.

To create a new building, the user simply grabs a building from the menu. This will duplicate the building model, resulting in the user holding a new building in their hand (figure 3).

The building can now be dragged to the desired position. The building is translated and rotated according to the hand's movements to maintain the illusion of grabbing. A shadow is displayed on the map to help the user judge the building's position and orientation, as seen in figure 3. To release the building, the user simply spreads their fingers. The application ensures that buildings fall to the ground and remain upright at all times. Dropped buildings can, of course, be picked up again. To change the height of a building, the user grabs the top 3 cm of the building, and then drags it up or down. To destroy a building, the user picks it up and drops it onto the trash icon. While a fingertip is inside or on a building, a buzzer mounted on that fingertip activates, providing haptic feedback.

3.1.2 Pointing

Pointing gestures are used to create and manipulate virtual streets. A pointing gesture is recognized when the user touches their finger to the ground.

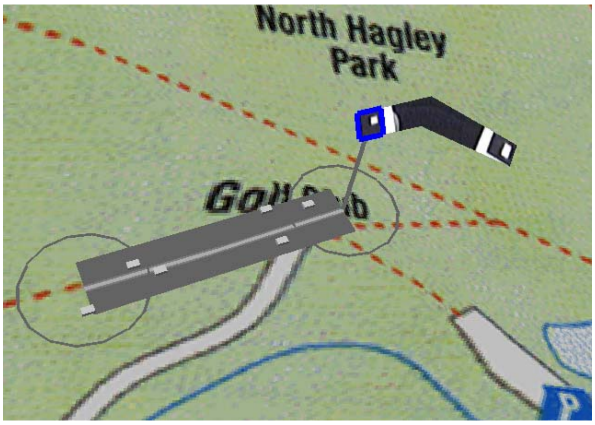

Figure 4: Streets are drawn by pointing at the start and ending point.

To create a new street, the user touches their index finger on the street icon and then points on the ground plane where they would like the street to start. A virtual street model will appear with one end at the start point and the other at the user's current pointing finger position. The currently selected street is highlighted by circles around its start and end points, and it is drawn on the ground rather than between the start point and the index finger. A vertical line between the index finger and the end of the street is drawn to create a visual connection between them as shown in figure 4. The user can stretch the street to the desired end point simply by moving their pointing finger. To manipulate existing streets, the user points at the current end of the road and then at the new desired end point. To discard a street, the user places the end point on the trash icon.

3.1.3 Pressing

Pressing gestures are a modification of pointing gestures and are used to interact with a virtual button. This can be pressed just like a real button, providing visual feedback by changing its height (see figure 5), and also haptic feedback. Button input is used to change to the immersive VR viewing mode.

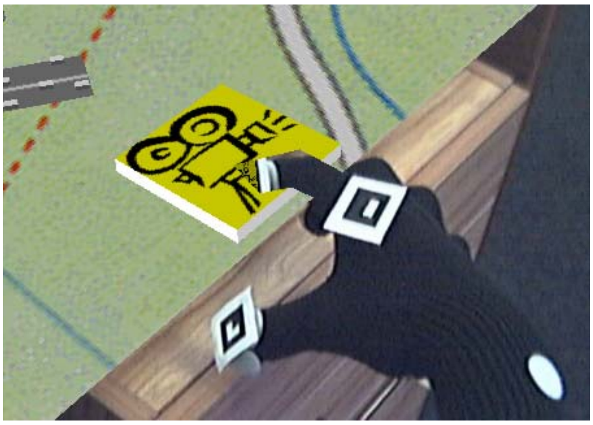

Figure 5: Pressing a virtual button.

3.1.4 Navigation

The user can easily change between AR and immersive VR viewing modes. The immersive mode is entered by pressing the immersive mode button and then pointing at the place where the user wants to be immersed. The user will then be transported to that place in the virtual scene (see figure 6).

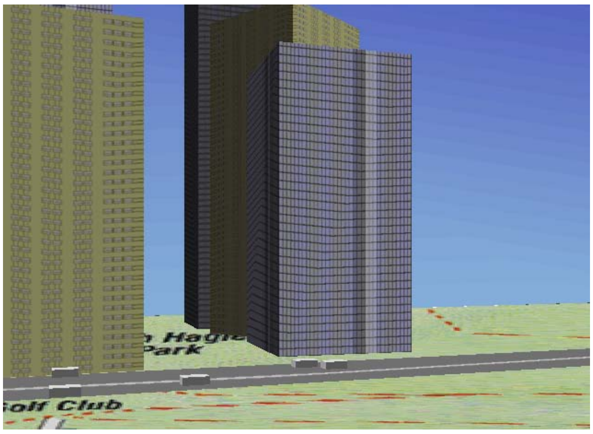

Figure 6: Immersive view of a model.

When immersed in the virtual scene the user can use their index finger to rotate their view left and right. When the user looks at their finger and then moves it to the left their viewpoint will rotate left, while moving the finger to the right will rotate the user to the right. Other functionality, such as looking up and down or changing position, has not yet been implemented. To move back to the AR view the user just has to move their thumb and index finger more than 10 cm apart. This command gesture draws from the metaphor of releasing the immersive mode.

3.2The Urban Design Setup

We use a 115 cm by 60 cm board covered with 24 markers, each 10 cm square as seen in figure 7. These markers are used by the ARToolKit computer vision based tracking library [ARToolKit 2003] to find the user's viewpoint in the world coordinate frame. A 120 cm by 70 cm virtual map of a park is registered on this board.

The glove has 2 cm square markers mounted on the tips of the thumb and the index finger, and on the hand itself. A buzzer is attached to each of these fingertips and controlled by an electronic circuit controlled through the computer's parallel port.

Users wear an i-visor HMD with a web camera attached (see figure 7). In the HMD, the user sees a video view of the real world with computer graphics superimposed. Both the camera and the HMD work at a resolution of 640 by 480 pixels. The output as seen by the participant is also displayed on the computer's monitor. The computer, an Athlon XP 2000+ with an nVidia GeForce4 Ti video card, runs the application at 20 frames per second. However, the current implementation is not speed optimized.

3.3Tracking

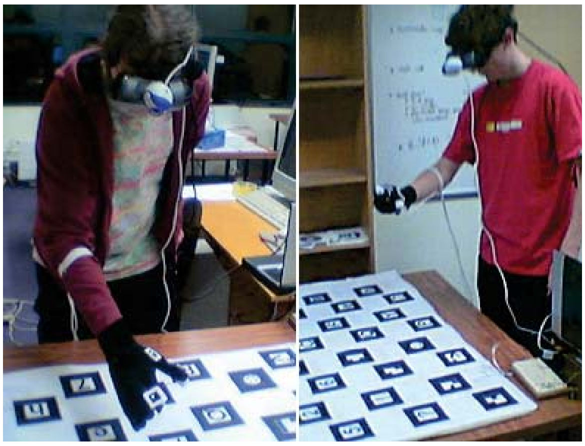

Figure 7: The urban planning demonstration setup.

In order to capture hand input, a gesture tracking system is needed. We use the ARToolKit computer vision tracking library [ARToolKit 2003]. This open source library will calculate camera position and orientation relative to one or more black square fiducial markers. These markers are attached to the user's hand and to the real environment to provide hand and world coordinate tracking. This same software was used by Piekarski and Thomas in their Tinmith system [Piekarski 2003].

3.3.1 World Coordinate Tracking

In the urban design application the user's head position is tracked in world coordinates (relative to the virtual park scene). To do this we use the ARToolKit's multiple marker tracking capabilities. With 24 markers 10 cm square we are able to reliably track the user up to 180 cm away from the sheet. As long as one of the markers is in view, we are able to calculate the user's viewpoint. This means that the tracking will not fail when the user obscures some of the markers with their hand. Reliable tracking is necessary for a usable AR interface.

3.3.2 Finger Tracking

In addition to tracking the user's viewpoint, their fingers also need to be tracked. To support this we use a glove with small markers attached to two fingertips and the hand (see figure 8). We found that a marker 2 cm wide was recognized by the ARToolKit at arm's length with a camera resolution of 640 by 480 pixels. However, in this configuration, ARToolKit has problems tracking marker orientation and distance from the camera. Tracking both the users view and finger positions relative to the user's viewpoint, we are able to get the position of the index finger, thumb and hand in the world coordinate frame.

3.3.3 Hand Tracking

The visual position of the fingers is the most important and immediate cue in our system. Gordon et al. suggest that it might not be sufficient to only display correctly registered fingertips. Instead, the whole hand should be shown [Gordon et al. 2002]. However, since virtual objects are simply an overlay on the real world, they will also cover a hand when it should appear in front of them.

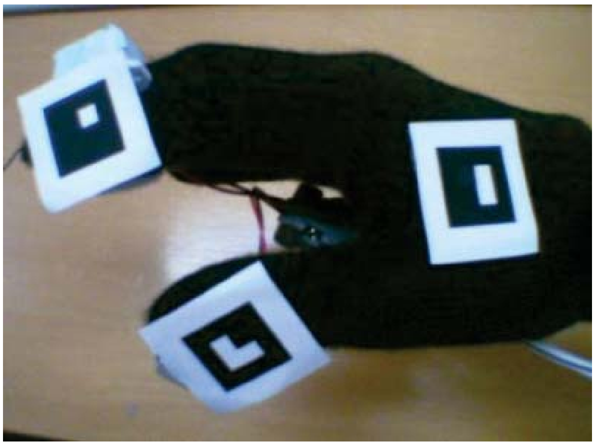

Figure 8: The glove, showing the markers used for hand tracking.

In order to track the fingers accurately and provide correct occlusion cues, we added a third marker between the base knuckles of the thumb and forefinger. This part of the hand barely moves while the thumb and forefinger pinch and release. This enables us to correctly render the fingers as shown in figure 9.

3.4The Hand Model

Inspired by Walairacht et al. [2002], we use a simplified model of the hand to interpolate the position of the index finger and the thumb from three markers. Our goal is to provide a simple model of the hand that can be used for correct occlusion.

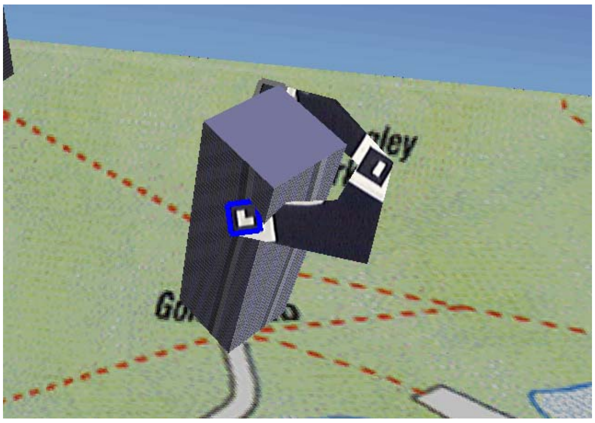

Figure 9: Correct occlusion of the fingers.

We assume that the movement of the index finger and thumb are subject to some rigid constraints. Our hand model assumes that the thumb only bends relative to the hand. The index finger is assumed to have two joints: at its base knuckle and between the other two knuckles. Thus, our hand model has three joints, each positioned relative to one marker. This allows for a simple drawing algorithm.

We use OpenGL stencil buffering to display the real image at the positions of the index finger and the thumb, as seen in figure 10.

A stencil polygon is drawn for each finger. For the index finger, the fingertip part of the polygon is fixed to the index finger marker and then connected to a line fixed to the hand marker. For the thumb, the part of the polygon that covers the hand is fixed to the hand marker and then connected to a line fixed on the thumb marker. The edges of the polygons that connect the parts fixed to the markers do not only create the illusion of bending but will also stretch and shrink to accommodate different lengths of fingers.

We decided against rendering virtual representations of the fingers in order to allow for easy interaction with both real and virtual objects. Virtual representations of the fingers would make fine interaction with real objects difficult.

3.5 Gesture recognition

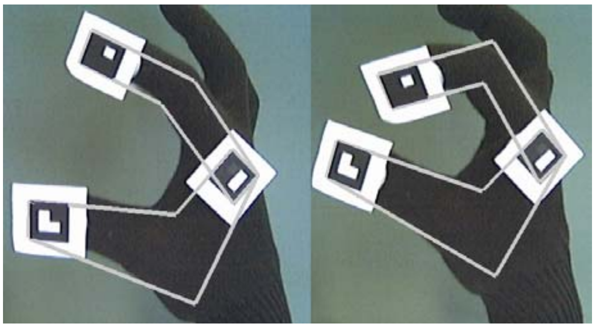

Figure 10: The simplified model of the hand.

Gesture recognition is based on the tracked position of the fingertips relative to the scene coordinate frame. Our application only requires simple gestures whose recognition can be based on one of three measurements: the position of the fingertips relative to an object, the position of the fingertips relative to the scene ground plane, or the position of the fingertips relative to each other.

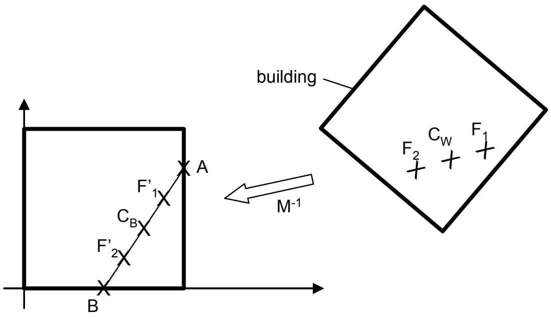

Figure 11: Relevant points for the grabbing algorithm in the building's coordinate system and the world coordinate system.

Grabbing, dragging and releasing gestures are used for manipulating buildings. Grabbing is recognized when both fingertips are found to be inside the same object. In the grabbing state, an object is translated and rotated relative to a point midway between both fingertips such that it appears to be moved by the user. Releasing is recognized when the fingertips are moved so far apart that they would no longer touch the building. The threshold distance for releasing is calculated when an object is picked up.

The corresponding algorithm consists of two parts: grabbing recognition and dragging. For each tracked position of the fingertips, one of these parts is executed, depending on which mode the system is in.

When no building is being dragged, the system tests each building to see if it is being grabbed (see figure 11):

- Use the building's inverse transformation matrix M^{-1} to translate the fingertip positions F_1 and F_2 into the building's coordinate system (F'_1 and F'_2)
- Check if F'_1 and F'_2 are inside the building.
- Continue only if both fingers are inside the building.
- Set a flag to indicate that dragging mode has been entered.
- Store the point in the middle between F_1 and F_2 (in world-coordinates, C_W and building-coordinates, C_B).
- Store the rotation α of fingers about the z-axis.
- Store the building's transformation matrix M.
- Store d, the distance between A and B

When a finger has been found to be inside a building, its buzzer is activated.

While a building is being held between the fingers, it is rotated and translated according the fingers' movement:

- If the distance between F_1 and F_2 is greater than d, the object is dropped. Otherwise, continue:
- Calculate how much the fingers have been rotated about the z-axis since the building was initially grabbed
- Rotate the building's original transformation matrix about C_B
- Calculate the current point C_W between the fingertips
- Translate the transformation matrix about the difference between current C_W and stored C_W
- Set the resulting matrix as the building's transformation matrix.

While the building is being dragged, both fingertip buzzers are activated.

Pointing is used for manipulating roads and pressing buttons. It is recognized when the index finger's tip is lower than a certain threshold above the virtual ground plane. To distinguish between two pointing gestures, for example when manipulating a street, a second threshold well above the first one was introduced. After a pointing gesture has been recognized, the user must lift the index finger above the second threshold before a new pointing gesture can be recognized. This is necessary to filter out tracking errors or unsteady hand positions. Pressing a button differs from pointing only in the threshold values.

The user can easily change between AR and immersive VR viewing modes by pressing the immersive view button. Navigation gestures are used to rotate the user in the immersive mode in our application. Rotating to the right and left is recognized when the index finger is in the right or left half of the user's view respectively. When the distance between both fingertips reaches a certain threshold, the user leaves the immersive mode. Both gestures are trivial to implement.

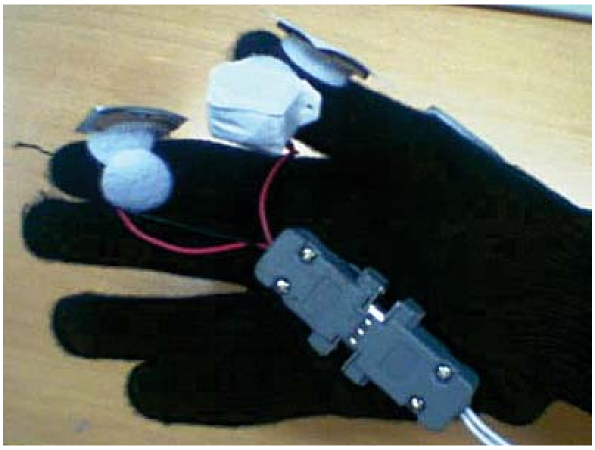

Figure 12: The buzzers mounted on the glove.

3.6Haptic Feedback

Interaction with virtual objects can be cumbersome, especially if the user only relies on visual cues. Haptic feedback is valuable for virtual object manipulation. It gives the user more confidence when manipulating objects and is an important cue when a fingertip is occluded by a virtual object.

We use small electronic buzzers to provide feedback for each tracked fingertip (figure 12). The buzzers vibrate at around 400 Hz and were muted while preserving their vibrotactile properties.

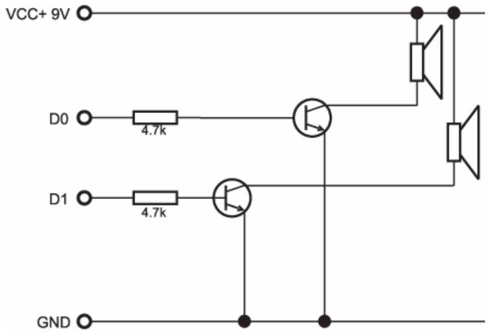

Figure 13: Circuit to control the buzzers for haptic feedback.

We found that the buzzers need not be tightly pressed onto the finger to provide valid haptic feedback, allowing for a comfortable fit of our glove. A simple transistor-switched circuit interfacing with the parallel port was used to control the buzzers (see figure 13). The current circuit only allows switching the buzzers on and off, with no support for different levels of vibration.

4 User Feedback

The FingARtips urban planning workspace has been shown at a large public demonstration where a number of visitors tried out the application as well as in more informal user studies. From these trials we have gained insight into how people use the glove, how they would like to use it and what problems we still have to solve

In general most users had no problems using the glove and were able to grab and drag buildings immediately, even when they had no prior experience with AR interfaces. Those who had difficulties were affected mainly by the constraints imposed by our currently limited optical tracking. For example, some people moved their hand so close to the camera that their hand occluded the world markers, preventing reliable tracking. Also, some users had problems moving their hand slowly and evenly and keeping it oriented toward the camera for the system to track the hand markers reliably. Most of the users managed to adjust to these constraints. However, at least one user perceived these constraints as “intimidating”, and much more reliable tracking is needed for truly intuitive interaction.

Although most users told us that interaction was easy and intuitive, many also felt fatigued after having used the interface. We believe that tracking problems caused much of the users’ workload. It is tiresome to move the hand slowly and keep it oriented toward the camera. Occasionally, changing lighting conditions would cause the workspace or the fingers to disappear for a few frames. This destroys the illusion of natural interaction and adds to the mental and physical workload.

Many users were fascinated that they could manipulate virtual objects in the same way as real objects. We often heard comments such as “Wow, I can really grab it!” and “This is so easy!” We found that when users had learned to grab and place buildings with confidence, they quickly acquired the other gestures as well. At this stage, usually under one or two minutes of practicing, many users had already mastered the interface and were concentrating on their urban model, uttering comments such as “This is much better” after resizing their first building.

Users also had no problem navigating in the immersive view. They enjoyed being immersed in the world they just created, but some felt the transition between AR view and immersive view was too abrupt. Users also sometimes exited the immersive view accidentally. In addition to testing the distance between the two tracked finger tips, a releasing movement should be recognized to make the command gesture less ambiguous.

Our experience with the FingARtips system suggests that correct occlusion of the fingers is a key component for successful hand based interaction. While implementing the system, we found that it is quite a challenge to grab a building when only the tracked finger tips are rendered correctly and the fingers themselves are incorrectly occluded by virtual objects. After we included the whole fingers in our hand model to solve the occlusion problem, we felt that it was much easier to interact with the buildings. Initially, we assumed that the thumb and index finger would never bend and shared a single joint at the hand marker. This ‘scissorhand’ was a big improvement over the finger-tips-only solution, but many participants still thought interaction was a bit awkward. They also did not know which finger was displayed when only one fingertip marker could be tracked. The hand model presented in this paper allows for satisfactory occlusion and much more intuitive interaction.

Adding haptic feedback had a similar positive effect. For most grabbing actions, the index finger is hidden behind the building and early versions of the system did not give any feedback whether a hidden finger was touching the building. Haptic feedback, simulating the natural feedback for this situation, works very well. Most importantly, haptic feedback seemed to increase the confidence with which users interacted with the buildings.

We were surprised with how well the haptic feedback assists the user. We expected that it would be too primitive to aid the user since it currently does not give information on whether the object is securely grabbed or about to be dropped. However, we received very positive feedback, such as “I can really feel it!” Nobody complained about being distracted by the haptic feedback. But while we found that most users did not have any problem with the buzzers being mounted on the fingernail, a few said that they could not feel the vibration, or they could not distinguish which buzzer was activated. This might result from the loose mount of the buzzers on the glove or the relatively high frequency of the vibration.

Drawing streets was problematic for many users. The interface did not provide enough visual feedback on the selection of the road drawing mode, and the inaccurate tracking sometimes made it impossible to recognize that the user had touched the virtual ground plane. This is due to variations in the reported height value in world coordinates of a fingertip tracking pattern. When a finger marker is lying flat on the marker board, its tracked height can vary between -2 and +5 cm. As one consequence, we turned the immersive icon into a raised button to make sure that it is not registered as lying under the physical table. The button also provides satisfactory visual and haptic feedback.

The final usability problem that we noticed is the ambiguous position of the hand relative to the buildings. No depth cue such as shadows or binocular disparity other than occlusion is provided. Thus, users cannot perceive exactly how far away their hand is from a building, leading to the feeling of “fishing” for buildings. This may be another factor that contributes to the mental and physical workload.

The feedback that we received from the informal evaluation was positive, with most criticisms being derived from tracking problems. Most users were able to confidently use the system after short periods of time and were fascinated by moving virtual objects with their hand. We believe that the correct occlusion of the fingers and haptic feedback are key features of hand-based interaction. Cues about the depth position of the hand may be another key component. These key cues should be further evaluated in a formal study, once the tracking performance is improved.

5 Lessons Learned

From our experience developing the urban planning application and implementing gesture input in an AR interface, we arrived at the following set of guidelines and lessons learned:

- *Impose constraints*. By taking advantage of the constraints of our urban planning scenario, many tracking problems could be hidden. These constraints also reduced the user’s workload by removing redundant possibilities of action. For example, we made placing the buildings easier by ensuring that they fell to the ground when being released, and by only allowing vertical rotation
- *Do not distract*. No participant had problems in understanding the objective of the urban planning workspace. Due to the simplicity and enjoyable nature of the task, participants were able to quickly use the glove. We initially intended to use a simple game as a sample application, but it soon became clear that the rules of such a game would only add to the user’s workload and distract from the glove input.
- *Do not ignore tracking errors*. We found that it was important to design the application so that it would work well despite the tracking errors. All objects had a certain diameter so that it was easy to manipulate them. We also did not try to implement high precision interaction tasks.
- *Compromise beauty for stability*. While we could have tried to render realistic virtual fingers or created a volumetric hand model, our simple approach has the advantage that the user sees a stable representation of the fingers despite relatively unreliable tracking. This is crucial for a usable system.

6 Future Work

While we are happy with the preliminary results from this early version of the FingARtips glove, there are relevant usability problems that we would like to overcome in the future.

The most important problem to be addressed is tracking inaccuracy. Only when the glove is reliably tracked can the system be thoroughly evaluated and put to use. We would like to improve the robustness of our system while maintaining its single-camera simplicity and mobility. This could be achieved by adding more markers to the fingertips and hand, enabling greater freedom of rotation to the user. An alternative solution would be to adopt the exoskeleton-based tracking system of Dorfmüller-Ulhaas and Schmalstieg [2001].

ARToolKit is a computer vision based tracking library and so is affected by lighting conditions. We found that the fingertip markers did not work well in bad lighting conditions. While the world markers were stationary relative to the light sources, the fingertip markers were constantly moving and re-orienting in space, requiring multiple light sources to work best. In order to minimize lighting effects we plan to use Pintaric’s dynamic thresholding technique [Pintaric 2003] which will increase the robustness of the tracking under changing lighting conditions.

To increase the mobility of the workspace, natural feature tracking as used by Gordon et al. [2002] could replace the marker board.

The finger visualization could be improved by assuming that the index finger could bend at two joints. Even more accurate visualization could be achieved by using more markers or by contrasting the image of the hand to the background patterns. Instead of simple polygons, we could use volumes for the stencil buffer, created from a three-dimensional hand model. This would improve realism when fingers intersect with virtual objects or when the hand is rotated.

With the current system, users can not perceive how far their hand is away from the building they want to manipulate. This could be solved by introducing a stereoscopic display. However, this would require a more expensive hardware setup. Alternatively we could explore the use of shadows as a depth cue.

It might also be valuable to add more feedback, for example audio feedback or visual feedback when a building has been grabbed. Conversely, additional multimodal input could be used, such as speech input or input from a pinch glove as used by Thomas and Piekarski [2002]. It is also compelling and easy to modify our haptic feedback device so that the buzzers could provide feedback at different intensities to indicate different levels of pressure.

To make the interface more complete, we are interested in exploring more gestures based on the thumb and index finger as well as refining the currently implemented ones. For example, grabbing and releasing should be made less ambiguous, and new gestures such as nudging a building into the correct position should be investigated.

Additionally, we would like to explore two-handed interaction. It would also be worthwhile to investigate the application of VR far space interaction in AR such as the Go-Go technique [Poupyrev 1996], and to explore the transition between near and far space interaction.

Finally, we would like to formally evaluate the FingARtips glove and compare the many different approaches in visualization and multimodal input and feedback that we have used. We are especially interested in assessing the importance of what we believe to be the three key cues for hand based interaction: correct occlusion of the fingers, haptic feedback and cues for the distance from the target.

7 Conclusion

We have built a system that enables users to interact naturally with virtual objects in AR. It allows people to use two fingers to intuitively manipulate virtual objects. The FingARtips system also provides haptic feedback and solves the problem of occlusion of the real hand by virtual objects. The system is inexpensive and easy to transport and set up.

We have implemented a small urban design application that demonstrates the versatility of our system. Gestures such as grabbing, dragging, rotating, dropping, pushing and pointing are used to construct an animated virtual city and to enter and navigate it for an immersive experience.

The demo application has been tested informally in a variety of settings, including a large public demonstration. We found that people who have never before used an AR application rapidly learned how to use our system. The evaluation also identified usability concerns, with the accuracy of glove tracking being the main problem. We will address this issue in our further work.

The availability of haptic feedback and the use of the stencil buffer appeared to greatly enhance the usability of the glove, and we believe that these are two key features for natural and intuitive manipulation of objects in AR. Conversely, the absence of fine-grain depth cues made interaction difficult, and we believe that they might be another key feature. Our future work will concentrate on empirically examining how best to implement these cues, and on scrutinizing their relative contribution to the overall usability of the system

We are pleased that our relatively simple system works so successfully and we believe that once a robust tracking solution is found hand- and finger- based manipulation will be an important part of AR applications in the future.

Acknowledgments

Special thanks to Malcolm Gordon from the Department of Electronic and Computer Engineering at the University of Canterbury, who helped us get our hardware up and running.

References

ARTOOLKIT 2003. http://www.hitl.washington.edu/artoolkit/

BRAUNSTEIN, M. AND ANDERSEN, G. AND RIEFER, D. 1982. The use of occlusion to resolve ambiguity in parallel projections. In *Perception and Psychophysics*, 31(3), 261-267

CODELLA, C. (1992) Interactive Simulation in a Multi-Person Virtual World. In *Proceedings of CHI '92*. ACM Press, New York.

CUTTING, J. 1997. How the eye measures reality and virtual reality. In *Behavior Research Methods, Instruments & Computers*, 29(1), 27-36

DIZIO, P. AND LACKNER, J. 2002. Proprioceptive Adaption and Aftereffects. In *Handbook of Virtual Environments*, 751-771

DORFMÜLLER-ULHAAS, K. AND SCHMALSTIEG, D. 2001. Finger Tracking for Interaction in Augmented Environments. In *Proceedings ISAR'01, New York, Oct. 2001*

HAUBER, J. 2002. TactilePointer - Entwicklung eines Interaktions-Devices mit moto-taktiler Unterstuetzung fuer augmentierte und virtuelle Umgebungen. *Diploma Thesis, Fachhochschule Ulm*

FURMANSKI, C. AND AZUMA, R. AND DAILY, M. 2002. Augmented-reality visualizations guided by cognition: Perceptual heuristics for combining visible and obscured information. *Proceedings of the International Symposium on Mixed and Augmented Reality.* pp. 215-224.

GORDON, G. AND BILLINGHURST, M. AND BELL, M. AND WOODFILL, J. AND KOWALIK, B. AND ERENDI, A. AND TILANDER J. 2002. The Use of Dense Stereo Range Data in Augmented Reality. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002)*, 14-23

ISHII, H.AND ULLMER , B. 1997. Tangible bits towards seamless interfaces between people, bits and atoms. *Proceedings of CHI97.* 1997. ACM. pp. 234-241.

KIYOKAWA, K. AND TAKEMURA, H. AND YOKOYA, N. 1999. A Collaboration Supporting Technique by Integrating a Shared Virtual Reality and a Shared Augmented Reality, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '99)*, Vol. VI, pp. 48 - 53.

KLATZKY, R. AND LEDERMAN, S. 2003. Touch. In *Handbook of Psychology*, vol. 4, pp. 147 - 176

LIVINGSTON, M. AND SWAN, J. AND GABBARD, J. AND HOELLERER, T. AND HIX, D. AND BAILLOT, Y. AND BROWN, D. 2003. Resolving Multiple Occluded Layers in Augmented Reality. In Proceedings of *Second IEEE and ACM International Symposium on Mixed and Augmented Reality, Tokyo, Japan,* 56-65

PIEKARSKI, W. AND THOMAS, B. H. 2002. Using ARToolKit for 3D Hand Position Tracking in Mobile Outdoor Environments. In *Proceedings of 1st Int'l Augmented Reality Toolkit Workshop*

PIEKARSKI, W. 2003. http://www.tinmith.net

PINTARIC, T. 2003. An Adaptive Thresholding Algorithm for the Augmented Reality Toolkit. in *Proceedings CD of 2nd IEEE Intl. Augmented Reality Toolkit Workshop*, Waseda Univ., Tokyo, Japan, Oct.7, 2003.

POUPYREV, I. AND BILLINGHURST M. AND WEGHORST, S. AND ICHIKAWA, T. 1996. The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation. In *VR. Proceedings of UIST 96*, Nov. 6-8 Seattle WA, 1996

SCHMALSTIEG, D. AND FUHRMANN, A. et al. 2000. Bridging multiple user interface dimensions with augmented reality systems. *ISAR'2000, IEEE.*

SUTHERLAND, I. 1968. A head-mounted three dimensional display. In *Proc. FJCC 1968,* pages 757-764, Washington, DC.

THOMAS, B. H. AND PIEKARSKI, W. 2002. Glove Based User Interaction Techniques for Augmented Reality in an Outdoor Environment. In *Virtual Reality: Research, Development, and Applications* (6) 3

VÄÄNÄNEN, K. AND BÖHM, K. 1993. Gesture Driven Interaction as a Human Factor in Virtual Environments - An Approach with Neural Networks. in *R. Earnshaw, M. Gigante, H. Jones (Eds.) Virtual Reality Systems*

WALAIRACHT, S. AND YAMADA, K. AND HASEGAWA, S. AND KOIKE, Y. AND SATO, M. 2002. 4 + 4 fingers manipulating virtual objects in mixed-reality environment. In *Presence: Teleoperators and Virtual Environments*, vol. 11, 2, 134-143. MIT Press.

WISE, S. 1990. Evaluation of a Fiber Optic Glove for Semi-automated Goniometric Measurements. *Rehabilitation Research and Development,* Vol. 27, No. 4, 1990, pp. 411-424.