Interacting With Dynamic Real Objects in Virtual Environments

by

Benjamin Chak Lum Lok

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2002

Approved by:

_____
Advisor: Dr. Frederick P. Brooks, Jr.

_____
Reader: Prof. Mary C. Whitton

_____
Reader: Dr. Gregory Welch

_____
Dr. Edward S. Johnson

_____
Dr. Anselmo Lastra

ABSTRACT

Benjamin Chak Lum Lok: Interacting With Dynamic Real Objects in Virtual Environments

(Under the direction of Frederick P. Brooks, Jr.)

Suppose one has a virtual model of a car engine and wants to use an immersive virtual environment (VE) to determine whether both a large man and a petite woman can readily replace the oil filter. This real world problem is difficult to solve efficiently with current modeling, tracking, and rendering techniques. Hybrid environments, systems that incorporate real and virtual objects within the VE, can greatly assist in studying this question.

We present algorithms to generate virtual representations, *avatars*, of dynamic real objects at interactive rates. Further, we present algorithms to allow virtual objects to interact with and respond to the real-object avatars. This allows dynamic real objects, such as the user, tools, and parts, to be visually and physically incorporated into the VE. The system uses image-based object reconstruction and a volume-querying mechanism to detect collisions and to determine plausible collision responses between virtual objects and the real-time avatars. This allows our system to provide the user natural interactions with the VE and visually faithful avatars.

But is incorporating real objects even useful for VE tasks? We conducted a user study that showed that for spatial cognitive manual tasks, hybrid environments provide a significant improvement in task performance measures. Also, participant responses show promise of our improving sense-of-presence over customary VE rendering and interaction approaches.

Finally, we have begun a collaboration with NASA Langley Research Center to apply the hybrid environment system to a satellite payload assembly verification task. In an informal case study, NASA

LaRC payload designers and engineers conducted common assembly tasks on payload models. The results suggest that hybrid environments could provide significant advantages for assembly verification and layout evaluation tasks.

ACKNOWLEDGEMENTS

I would like to acknowledge:

Dr. Frederick P Brooks, Jr. for being my advisor and for his guidance in my academic and personal development over the years.

Professor Mary C. Whitton for her support and seemingly endless patience in the course of developing this research. This work would not have been possible without her assistance and belief in me.

Drs. Gregory F. Welch, **Anselmo Lastra**, and **Edward S. Johnson**, my doctoral dissertation committee members. Their ideas, support, and encouragement were invaluable and have strongly shaped my development as a researcher.

Samir Naik for his excellent collaboration to bring this research to fruition.

Danette Allen, of NASA Langley Research Center, for our collaboration on applying this research to a real problem.

Samir Naik, **Sharif Razzaque**, **Brent Insko**, **Michael Meehan**, **Mark Harris**, **Paul Zimmons**, **Jason Jerald**, and the entire Effective Virtual Environments and Tracker project teams for their invaluable assistance, ideas, software and user study support, **Andrei State** and **Bill Baxter** for code support, **Dr. Henry Fuchs**, **David Harrison**, **John Thomas**, **Kurtis Keller**, and **Stephen Brumback** for equipment support, and **Tim Quigg**, **Paul Morris**, and **Janet Jones** for administrative support.

My study participants for their contributions to this work.

Dr. Sujeet Shenoi, my undergraduate advisor at the University of Tulsa, for encouraging me to pursue graduate studies in computer graphics.

The UNC Department of Computer Science, The LINK Foundation, The National Science Foundation, the National Institutes of Health National Center for Research Resources (Grant Number P41 RR 02170) for financial and equipment support used in this work.

And most importantly, I would like to acknowledge my parents **Michael Hoi Wai** and **Frances**, my brother **Jonathan**, sister **Melissa**, my best friend **Laura**, and my extended family for their love and support through the years. You truly are my foundation.

The system design and evaluation described in this dissertation was my own personal project, and I am fully responsible for those decisions. The system was built on a virtual environment

infrastructure developed by many faculty members and students at UNC-Chapel Hill. Moreover, the members of the Effective Virtual Environments project gave much assistance, especially in the user study.

We is used throughout this dissertation to describe this combination of personal design and team support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Introduction and Summary

1.1 Driving Issues

Motivation. Conducting design evaluation and assembly feasibility evaluation tasks in immersive virtual environments (VEs) enables designers to evaluate and validate multiple alternative designs more quickly and cheaply than if mock-ups are built and more thoroughly than can be done from drawings. Design review has become one of the major productive applications of VEs [Brooks99]. Virtual models can be used to study the following important design questions:

- Can an artifact readily be assembled?
- Can repairers readily service it?

Ideal. The ideal VE system would have the participant fully believe he was actually performing a task. Every component of the task would be fully replicated; the environment would be visually identical to the real task. Further, the participants would hear accurate sounds, smell identical odors, and when they reached out to touch an object, they would be to feel it. In the assembly verification example, the ideal system would present an experience identical to actually performing the assembly task. Parts and tools would have mass, feel real, and handle appropriately. The participant would interact with every object as he would if he were doing the task. The virtual objects would in turn respond to the participant's action appropriately. Training and simulation would be optimal [Sutherland65]. This is similar to the fictional Holodeck from the futuristic science fiction Star Trek universe, where participants were fully immersed in a computer-generated environment. In the mythos, the environments and objects were so real, if a person were shot with a virtual bullet, he would be physically killed.

Current VE Methods. Obviously, current VEs are far from that ideal system. Indeed, not interacting with every object as if it were real has distinct advantages, as in the bullet example. In current VEs, almost all objects in the environment are virtual. But both assembly and servicing are hands-on tasks, and the principal drawback of virtual models — that there is nothing there to feel, nothing to give manual affordances, and nothing to constrain motions — is a serious one for these applications. Using a six degree-of-freedom (DOF) wand to simulate a wrench, for example, is far from realistic, perhaps too far to be useful. Imagine trying to simulate a task as basic as unscrewing an oil filter from a car engine in such a VE!

Interacting with purely virtual objects imposes two limiting factors on VEs. First, since fully modeling and tracking the participant and other real objects is difficult, virtual objects cannot easily respond to them. Second, since the VE typically has limited information on the shape, appearance, and motion of the participant and other real objects, the visual representation of these objects within the VE is usually stylized and not necessarily visually faithful to the object itself.

The participant is represented within the virtual environment as an avatar. Avatars are typically represented by stylized human models, such as those provided in the commercial packages EDS's Jack [Ward01] or Curious Lab's Poser 4 [Simone99]. Although these models contain a substantial amount of detail, they do not visually match each specific participant's appearance. Previous research hypothesizes that this misrepresentation of self is so detrimental to VE effectiveness, it will reduce how much a participant believed he was "in" the virtual world, his sense-of-presence [Slater93, Welch96r, Heeter92].

We extend our definition of an *avatar* to include a virtual representation of any real object, not just the participant. The *real-object avatar* is registered with the real object and ideally has the same shape, appearance and motion as the real object.

Getting shape, appearance, motion, and actions from real objects, such as the participant's hand, specialized tools, or parts, requires specific technology and development for modeling, tracking, and interaction. For

example, in developing the purely virtual condition for our user study, we wanted to allow the participants to pick up and manipulate virtual blocks. This required developing software to incorporate tracked pinch gloves to control an avatar, interaction mechanisms among all the virtual objects, and models for the avatar and the blocks. Every possible input, action, and model for all objects, virtual and real, had to be defined, developed, and implemented. The resulting system also established very specific ways the participant could interact with the blocks. Further, any changes to the VE required substantial modifications to the software or technology base.

The additional development effort required, coupled with the difficulties of object tracking and modeling, lead designers to use few real objects in most VEs. Further, there are also restrictions on the types of real objects that can readily be incorporated into a VE. For example, highly deformable objects, such as a bushy plant, are especially difficult to model and track.

Working with virtual objects could hinder training and performance in tasks that require haptic feedback and natural affordances. For example, training with real tools would understandably be more effective than training with virtual approximations.

Incorporating Real Objects. We believe a system that could incorporate *dynamic real objects* would improve interactivity as well as provide visually faithful real-object avatars. We define dynamic objects as real objects that can change shape and appearance. Examples include a socket wrench set, clothing, and the human hand. For a substantial class of VEs, incorporating dynamic real objects has the potential to benefit task performance and presence. In assembly verification tasks, the participant, tools, and parts are typically dynamic in shape, motion, and appearance.

We define *incorporating real objects* as being able to handle, feel, and use real objects while simultaneously seeing registered representations of the real objects. One also wants virtual objects to react tot eh virtual representations of real objects. The challenges are visualizing the real objects within the VE and managing the interactions between the real and the virtual objects.

By having the real objects interacting with a virtual model, designers can see if there is enough space to reach a certain location or train people in assembling a device at different stages of construction, all while using real parts, using real tools, and accounting for the variability among assemblers. Today, neither standard tracking technologies nor modeling techniques are able to do this task at interactive rates.

Dynamic Real Objects. Incorporating dynamic real objects requires capturing both the shape and appearance and inserting this information into the VE. We present a system that generates approximate virtual models of dynamic real objects in real time. The shape information is calculated from multiple outside-looking-in cameras. The real-object appearance is captured from a camera that has a similar line of sight as the participant.

The advantages of interacting with real objects could allow applying VEs to tasks that are hampered by using all virtual objects. Specifically, we feel that spatial cognitive manual tasks would benefit with increased task performance from incorporating real objects. These tasks require problem solving through manipulating and orientating objects while maintaining mental relationships among them. These are common skills required in simulation and training VEs.

Slater, *et al*., have shown that VE participants develop a stronger sense-of-presence when they see even a highly stylized avatar representing themselves as opposed to no avatar [Slater93, Slater94]. Heeter suggests, "Perhaps it would feel even more like being there if you saw your real hand in the virtual world [Heeter92]." Video capture of real object appearance has this potential advantage — enhanced visual realism. When participants move one of their arms into the field of view, we want to show an accurately lit, pigmented, and clothed arm. Generating virtual representations of the participant in real time would allow the system to render a visually faithful avatar. Our system, which uses video capture of real object appearance, enables a test of Heeter's hypothesis.

1.2 Thesis Statement

We started off to prove the following:

Naturally interacting with real objects in immersive virtual environments improves task performance and sense-of-presence in cognitive tasks.

Our study results showed a significant task performance improvement, but did not show a significant difference in sense-of-presence.

1.3 Overview of Approach

Generating Virtual Representations of Real Objects. To demonstrate the truth of this thesis statement, we have developed a *hybrid environment* system that uses image-based object reconstruction algorithms to generate real-time virtual representations, avatars, of real objects. The participant sees both himself and any real objects introduced into the scene incorporated into the VE. *Incorporation of real objects* includes having VE subsystems, such as lighting, rendering, and physics simulations, be aware of and react to real objects. We use an image-based algorithm that does not require prior modeling and can support dynamic real objects, both of which are critical in assembly-design tasks.

Our system uses commodity graphics-acceleration hardware to accelerate computing a virtual approximation, the visual hull, of real objects. Current graphics hardware has a limited set of operations (compared to a general CPU), but can execute those operations very quickly. For example, the nVidia GeForce4 can calculate 3-D transformations and lighting for rendering 3-D triangles at over 75 million triangles a second. It can also draw over 1.2 billion pixels on the screen per second [Pabst02]. We use these same computations along with the associated common graphics memory buffers, such as the frame buffer and the stencil buffer, to generate virtual representations of real scene objects from arbitrary views in real time. The system discretizes the 3-D visual hull problem into a set of 2-D problems that can be solved by the graphics hardware.

To generate a virtual representation of a real object, we first capture the real object's shape and appearance. Then we render the virtual representation in the VE. Finally, the virtual representation is tested for collisions with other virtual objects.

From a set of live multiple, live-camera images, the system derives the visual hull for real objects in the scene. The visual hull is textured with the image from a HMD-mounted camera with a line of sight similar to that of the participant, such that he sees a virtual representation of himself that is faithful in appearance. The virtual representations are combined with virtual objects with correct obscuration. The results are computed at interactive rates, and thus the real-object avatars also have accurate representations of all joint motions and shape deformations.

Interactions with Virtual Representations of Real Objects. We developed algorithms to use the real-object avatars in virtual lighting and in physically based mechanics simulations. This includes new collision-detection and collision-response algorithms that exploit graphics hardware for computing results in real time. For example, they can be lit by virtual lights, shadowed by virtual objects, and cast shadows onto virtual objects. Also, we can detect when the real-object avatars collide with virtual objects, and provide plausible collision responses for the virtual objects. This type of interaction allows the real-object avatars to affect simulations such as particle systems, cloth simulations, and rigid-body dynamics.

In our oil filter example, we can thus detect if the real oil filter the user is carrying intersects the virtual engine model, can have the user's hand cast a shadow onto the virtual engine, and can enable the user's hand to brush a virtual wire aside as he tries to reach a specific area. In a sense we are merging two spaces, a physical space with real objects, and a virtual space with corresponding virtual objects.

User Studies of Interacting with Real Objects. Given this system, we wanted to explore the effects of haptics and visual fidelity of avatars on task performance and presence. For cognitive tasks:

- Will task performance significantly improve if participants interact with real objects instead of purely virtual objects?

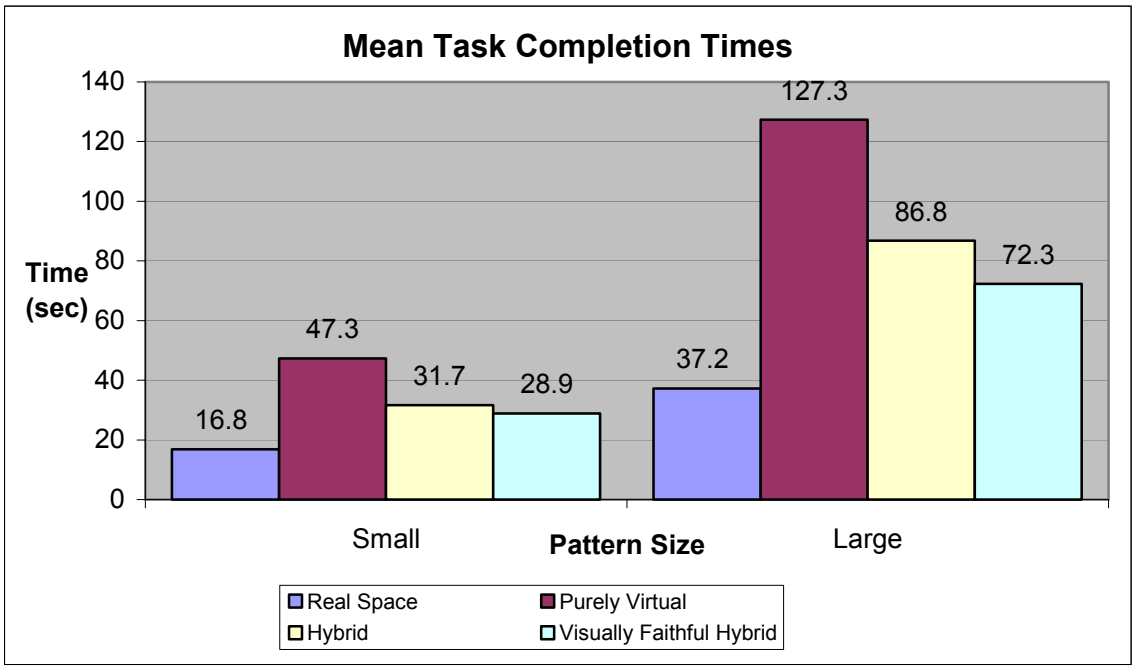
- Will sense-of-presence significantly improve when participants are represented by visually faithful self-avatars?

As opposed to perceptual motor tasks (e.g., pick up a pen), cognitive tasks require problem-solving decisions on actions (e.g., pick up a *red* pen). Most design verification and training tasks are cognitive. Studies suggest assembly planning and design are more efficient with immersive VEs, as opposed to when blueprints or even 3-D graphics models on desktop monitors are used [Banerjee99].

To test both hypotheses, we conducted a user study on a block arrangement task. We compared a purely virtual task system and two hybrid task systems that differed in level of visual fidelity. In all three cases, we used a real-space task system as a baseline.

For task performance, we compared the time it took for participants to complete the task in the VE condition to their time in performing the task in real space. The task performance measures are summarized in Figure 1. The study results show a statistically significant improvement in task performance measures for interacting with real objects within a VE compared to interacting with virtual objects.

Figure 1 – Task Performance in VEs with different interaction conditions. The Real Space Environment (RSE)was the baseline condition. The Purely Virtual (PVE) had participants manipulating virtual objects. Both the Hybrid (HE) and Visually Faithful Hybrid (VFHE) had participants manipulating real objects.

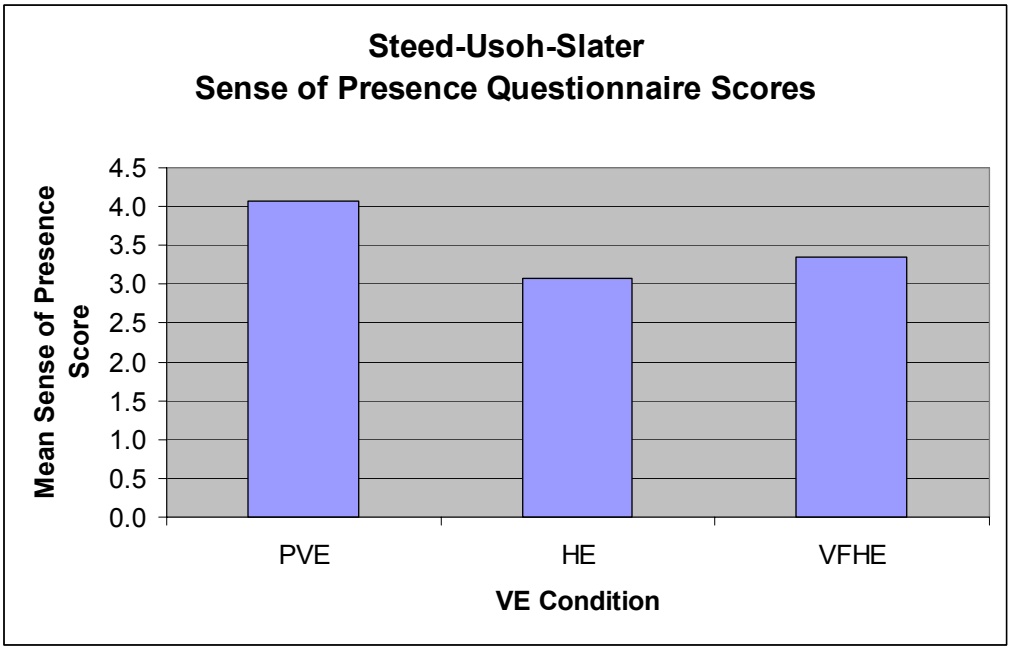


For sense-of-presence comparison, we used the following explicit definition of *presence* from Slater and Usoh [Slater93]:

"The extent to which human participants in a virtual environment allow themselves to be convinced while experiencing the effects of a computer-synthesized virtual environment that they are somewhere other than where they physically are – that 'somewhere' being determined by the image, sounds, and physical sensations provided by the computer-synthesized virtual environment to their senses."

We administered a presence questionnaire and interviewed participants after they completed the experience. We compared responses between the VEs that presented generic avatars to the VE that presented a personalized avatar. The results did not show a statistically significant (Figure 2) difference in self-reported sense-of-presence for participants presented with visually faithful avatars compared to those presented with generic avatars.

Figure 2 – Mean Sense-of-Presence Scores for the different VE conditions. VFHE had visually faithful avatars, while HE and PVE had generic avatars.



Application to an Assembly Verification Task. We wanted to apply the system to a real world problem to evaluate the potential utility of this technology. We began a collaboration with a payload-design group at NASA Langley Research Center (NASA LaRC) in Hampton, Virginia. In an exploratory study, four experts in payload design used the hybrid system to evaluate an abstracted version of a payload integration task.

The design problem was to establish the distance between two virtual payload models to allow the participant to attach a real pipe and connect a real power cable to a physical connector. The connector was registered with a fixed virtual model of one of the payload objects. Collision detection of the user, real tools, and real parts were done against the multiple virtual payloads.

The participants' experiences with the system anecdotally showed the positive effect of handling real objects when interacting with virtual objects. The NASA LaRC engineers were surprised at the layout issues they encountered, even in the simplified example we had created. In debriefing, they concurred that

early detection and correction of the errors detected while using the hybrid system would have saved them a substantial amount of money and time costs in correcting and refining their design.

1.4 Innovations

The work described in this dissertation investigates incorporating dynamic real objects into virtual environment: the methods, their usefulness, and an application. We developed an algorithm for generating virtual representations of real objects at interactive rates. The algorithm uses graphics hardware to reconstruct a visual hull of real objects using a novel volume-querying technique, and textures the visual hull with the images of the object. We developed hardware-accelerated collision-detection and collision-response algorithms for managing interactions between real and virtual objects. It is to our understanding that this is the first system that allows for the incorporation of arbitrary dynamic real objects into a VE.

We wanted to see if these methods for incorporating real objects were advantageous for cognitive tasks. We conducted studies to examine whether the effects of interaction modality and avatar fidelity on task performance and sense-of-presence. We found that interacting with real objects significantly improves task performance for spatial cognitive tasks. We did not find a significant difference in reported sense-of-presence due to avatar visual fidelity.

We have begun applying our system to a NASA LaRC assembly verification task. Initial trials with payload designers show promise in the effectiveness of the reconstruction system to aid in payload development.

2. Previous Work

Our work builds on the research areas of, and presents new algorithms for, incorporating real objects into VEs, human avatars in VEs, and interaction techniques in VEs. We discuss prior research in each area in turn.

2.1 Incorporating Real Objects into VEs

Overview. Our goal is to populate a VE with virtual representations of dynamic real objects. We focus on the specific problem of: given a real object, generating a virtual representation of it. Once we have this representation, we seek to incorporate that into the VE.

We define *incorporation of real objects* as having VE subsystems, such as lighting, rendering, and physics simulations, be aware of and react to real objects. This involves two primary components: capturing object information, and having virtual systems interact with the captured data. We review current algorithms for capturing this information, then look at methods for using the captured data as part of a virtual system.

Applications that incorporate real objects seek to capture the shape, surface appearance, and motion of the real objects. Object material properties and articulation may also be of interest.

The requirements for incorporation of real objects are application-specific. Does it have to be done in real time? Are the objects dynamic, i.e. move, change shape, change appearance, change other properties? What is the required accuracy? How will the rest of the VE use these representations?

Prebuilt, catalog models are usually not available for specific real objects. Making measurements and then using a modeling package is tedious and laborious for complex static objects, and near impossible for

capturing the degrees of freedom and articulation of dynamic objects. We give three example applications that require capturing information about specific, complex real objects.

Creating a virtual version of a real scene has many applications in movie making, computer games, and generating 3-D records, e.g., as capturing models of archeological sites, sculptures [Levoy00], or crime scenes. These models can then be viewed in VEs for education, visualization, and exploration. Using traditional tape measure, camera, and CAD approaches for these tasks is extremely time-consuming. These applications benefit greatly from automated highly accurate shape and appearance capture of scenes and objects, usually static ones. Static scenes were among the first real objects for which automated capture has been used.

Techniques for viewing recorded or live events from novel viewpoints are enhancing entertainment and analysis applications. They have enabled experts to analyze golf swings and sportscasters to present television viewers dynamic perspective of plays. Kanade's (CMU) Eye Vision system debuted at Superbowl XXXV and generated novel views of the action from image data generated from a ring of cameras mounted in the stadium [Baba00]. This allowed commentators to replay an event from different perspectives, letting the audience see the action from the quarterback or wide receiver's perspective. This required capturing within a short amount of time the shape, motion, and appearance information for a large scene populated with many objects.

Tele-immersion applications aim to extend videoconferencing's 2-D approach to provide 3-D perception. Researchers hypothesize that interpersonal communication will be improved through viewing the other party with all the proper 3-D cues. The UNC Office of the Future project, described by Raskar, generates 3-D models of participants from multiple camera images [Raskar98]. It then transmits the novel view of the virtual representation of the person to a distant location. The communicating parties are highly dynamic real objects.

Each of these applications requires generating virtual representations of real objects. We examine current approaches for modeling real objects, tracking real objects, and incorporating the virtual representation of real objects.

Modeling Real Objects. Many commercial packages are available for creating virtual models. Creating virtual models of real objects is a specific subset of the problem called *object* or *scene* reconstruction. A common distinction between the two is that object reconstruction focuses primarily on capturing data on a specific set of objects whereas scene reconstruction focuses on capturing data for an entire location.

Applications often have specific requirements for the virtual representation of a real object, and different algorithms are uniquely suited for varying classes of problems. The primary characteristics of model-generation methods for a real object are:

Accuracy – How close to the real object is the virtual representation? Some applications, such as surgery planning, have very strict requirements on how closely the virtual representation needs to correspond to the real object.

Error Type – Is the resulting virtual representation conservative (the virtual volume fully contains the real object) or speculative (there exists points in the real object not within the virtual volume)? What are the systematic and random errors of the system?

Time – Is the approach designed for real-time model generation or is it limited to, and optimized for, models of static objects? For real-time approaches, what are the sampling rates and latency?

Active/Passive – Does the capture of object information require instrumenting the real objects, such as attaching trackers, or touching the object with tracked pointers? Some objects such as historical artifacts could be irreversibly damaged by physical interactions. Camera- or laser-based methods are better approaches for capturing delicate objects' data, as in Levoy's capture of Michelangelo's *David* [Levoy00].

Non-Real-Time Modeling of Real Objects. The tradeoff between accuracy and computation divides reconstruction algorithms into those suitable for real-time applications and those suitable only for off-line applications. Non-real-time algorithms capture camera images, laser range data, or tracker readings of real objects and hence can emphasize generating accurate geometric models.

One of the first methods for capturing object shape was to track a device, typically a stylus, and move the stylus on the surface of the object and record the reported tracker position. The resulting set of surface points was then typically converted into a polygonal mesh. Some commercial products, such as the Immersion Microscribe 3-D, provide high-precision mechanical tracking [http://www.immersion.com/products/3-D/capture/msinfo.shtml].

Commercial products are available that sweep lasers across a surface and measure the time of flight for the beam to reflect to a sensor. Given these distances to points on real objects, algorithms can generate point clouds or polygonal meshes of a real environment [Turk94]. Scene digitizers are useful for modeling real objects or environments, such as a crime scene or a movie set.

Image-based scene reconstruction is a subcategory of a large class of camera-based model-generation techniques called Shape from X algorithms. Examples of Shape from X include shape from texture, shape from shading, shape from silhouettes, and shape from motion. These techniques generate virtual models of real objects' shape and appearance by examining changes in input camera images caused by the light interacting with scene objects [Faugeras93a].

The generic problem in Shape from X is to find 3-D coordinates of scene objects from multiple 2-D images. One common approach, correlation-based stereo, is to look for pixels in one image and search for pixels in other images that correspond to the same point on a real object [Faugeras93b]. The multiple sightings of a point establish the point's position in the scene. The Virtualized Reality work by Kanade et al. uses a dense-stereo algorithm to find correspondences. Forty-nine cameras, connected to seventeen

computers, record events in a room [Baba00]. Offline, to generate a view, the nearest five cameras to a virtual camera's pose are used, and baseline stereo is used to generate volumetric representations of real-objects in the scene.

Object reconstruction algorithms generate a volumetric, polygonal, or point cloud representations of objects in the scene. Volumetric approaches to model generation divide space into discrete volume elements called *voxels*. The algorithms partition or carve the volume into voxels that contain real objects and those that do not based on the established correspondences [Carr98, Chien86, Potmesil87]. Algorithms that calculate real object surfaces output surface points as a point cloud or compute connectivity information to generate polygonal models representations of the real objects [Edelsbrunner94].

Real-Time Modeling of Real Objects. Real-time algorithms simplify the object reconstruction by restricting the inputs, making simplifying assumptions, or accepting output limitations. This allows the desired model to be computed at interactive rates.

For example, the 3-D Tele-Immersion reconstruction algorithm by Daniilidis, *et al*., restricts the reconstructed volume size so that usable results can be computed in real time using their dense-stereo algorithm [Daniilidis00]. The camera images, numbering five to seven in their current implementation, are reconstructed on the capture side and a depth image is sent across the Internet to the display side.

For some applications, precise models of real objects are not necessary. One simplification is to compute approximations of the objects' shapes, such as the visual hull. A shape-from-silhouette concept, the *visual hull*, for a set of objects and set of *n* cameras, is the tightest volume that can be obtained by examining only the object silhouettes, as seen by the cameras [Laurentini94].

At SIGGRAPH 2000, Matusik, *et al*., presented an image-based visual hull algorithm, "Image Based Visual Hulls" (IBVH), that uses image-based rendering (IBR) algorithms to calculate novel views of visual hulls at interactive rates [Matusik00]. First, silhouette boundaries are calculated for all newly introduced real

objects through image subtraction. Each pixel in the novel-view image-plane maps to an epipolar line in each source camera image. To determine if the visual hull projects onto a pixel in the novel view, the source images are examined along these epipolar lines to see if silhouette spans overlap. Overlaps indicate the visual hull points that project onto the novel-view pixel. Further, the IBVH algorithm computes visibility and coloring of the visual hull.

The IBVH system uses four cameras connected to four PCs on a dedicated network to capture images and a quad-processor PC to compute the reconstruction. Their work also provides methods for converting the visual hull surface into polygonal meshes [Matusik01]. Matusik's algorithm has an $O(n^2)$ work complexity, where n is sampling resolution. Our algorithm for recovering real object shape is similar but with substantial differences in both application and functionality.

First, our approach, $O(n^3)$, is a graphics hardware-accelerated algorithm that benefits from the rapid performance and functionality upgrades that commodity graphics hardware provides. Second, our visual hull algorithm is well suited for first-person VE rendering with specific algorithms for coloring of the visual hull. Third, our volume-querying algorithm, discussed in detail in Chapter 3 and Chapter 4, provide efficient mechanisms for collision detection and different types of intersection queries with the visual hull. Finally, our algorithm is not sensitive to the number or complexity of the real objects we wish to reconstruct, and the reconstruction and collision detection work complexity scales linearly with the number of cameras.

Registering Virtual Representations with the Real Objects. We enforce a registration of the virtual representation and the real object. For dynamic real objects, this means that capturing motion and deformation in addition to shape and appearance. Defining the motion of a real object requires capturing its position and orientation. To do this, we must consider the following issues:

- Application Requirements – The performance requirements. For example, head tracking for head-mounted VE systems must return data with minimal latency and high precision. Inability to satisfy these requirements will result in simulator sickness during prolonged

exposures. Medical and military training applications have high accuracy and low latency requirements for motion information.

- Real Object Types – Are the objects rigid bodies, articulated rigid bodies, or deformable bodies? Does the object topology change?
- Available Systems – The speed, latency, accuracy, and precision of available tracking systems

Tracking systems, which report the motion of real objects, can be divided into two major groups, active and passive tracking. We define *active tracking* as physically attaching devices to an object for capturing motion information. In contrast, *passive tracking* uses outside-looking-in devices, such as lasers or cameras, to capture information without augmenting the objects.

Active tracking is the most common method for tracking real objects. Devices, that use magnetic fields, acoustic ranging, optical readings, retro-reflectors or gyros, are attached to the object. These devices, either alone, or in combination with an additional sensor source, return location and/or orientation in relation to some reference point. The tracker readings are then used to place and orient virtual models. The goal in hybrid systems is to register a virtual model with a real object. For example, Hoffman, *et al*., attached a magnetic tracker to a real plate to register a virtual model of a plate [Hoffman98]. This allowed the participant to pick up a real plate where the virtual model appeared. Other products include the CyberGlove from Immersion Corporation, which has twenty-two sensors that report joint angles for human hands and fingers [http://www.immersion.com/products/3-D/interaction/cyberglove.shtml], and Measurand's ShapeTape [Butcher00], a flexible curvature-sensing device that continually reports its form.

Active tracking has the following advantages:

- Commonly used,
- Well understood,
- Easily implemented,
- Generates very accurate and robust results for rigid bodies.

Active tracking has the following disadvantages:

- Imposes physical restrictions – the tracking devices, mounting locations, and any associated wires could restrict natural object motion.
- Imposes system restrictions – each tracking device typically reports motion information for a single point, usually the device's position and orientation. This limited input is inefficient for objects with substantial motion information, such as a human body.
- Limited applicability for tracking highly deformable bodies.

As opposed to the augmenting approach of adding trackers, image-based algorithms use cameras that passively observe the real objects. These algorithms capture object motion through generating new object representations from new camera images.

Camera-based approaches have the following advantages over tracker-based methods for capturing object motion:

- Allow for a greater range of object topologies,
- No *a priori* modeling, hence flexibility and efficiency,
- Non-rigid bodies can be more readily supported,
- Fewer physical restrictions.

Camera-based approaches have the following disadvantages compared to tracker-based methods for capturing object motion:

- Limited number of views of the scene reduces tracking precision.
- Limited in dealing with object occlusions and complex object topologies.
- Camera resolution, camera calibration, and image noise can drastically effect tracking accuracy.
- Limited information about the real objects being tracked. The object reconstruction algorithms in particular can only determine whether a volume is or is not occupied, and

not necessarily *what* the object that occupies the volume is. For example, if the user is holding a tool, the system cannot disambiguate between the two objects, and the volume is treated as one object.

Collision Detection. Detecting and resolving collisions between moving objects is a fundamental issue in physical simulations. If we are to incorporate real objects into the VE, then we must be able to detect when real and virtual objects intersect so as not to create cue conflicts because of interpenetration. From this we can proceed to determine how to resolve the intersection.

Collision detection between virtual objects is an area of *vast* previous and current research. The applicability of current algorithms depends on virtual object representation, object topology, and application requirements. We review a few image based, graphics-hardware accelerated, and volumetric algorithms for collision detection to which our algorithm is most related.

Our virtual representations of real objects are not geometric models and do not have motion information such as velocity and mass. This imposes unique requirements on detecting and dealing with collisions. Collision detection between polygonal objects, splines, and algebraic surfaces can be done with highly efficient and accurate packages such as Swift++ [Ehmann01]. Hoff and Baciu's techniques use commodity graphics-hardware's accelerated functions to solve for collisions and generate penetration information [Hoff01, Baciu99]. Boyles and Fang proposed an algorithm for collision detection between volumetric representations of objects, common in medical applications [Boyles00]. Other work on collision detection between real and virtual objects focused on first creating geometric models of the rigid-body real objects, and then detecting and resolving collision between the models [Breen95].

2.2 Avatars in VEs

Overview. An *avatar* is an embodiment of an ideal or belief. It is derived from a Sanskrit phrase meaning "he descends" and "he crosses over" referring to a god taking a human form on earth. In VEs, avatars are the participant's self-representation within the virtual environment. This review focuses on algorithms for

generating and controlling, the self-avatar, and on research into the effects of the self-avatar on the immersive VE experience. In the previous section, we used the term *avatar* to represent the virtual representation of any real object. In this section, we limit our discussion of avatars to the visual representation of the participant.

Current Avatar Approaches. Existing VE systems provide the participant with either choices of an avatar from a library of representations, a generic avatar (each participant has the same avatar), or no avatar at all. From our survey of the VE research, the most common approach is to provide a generic avatar – literally, one size fits all.

Researchers believe that providing generic avatars substantially improves sense-of-presence over providing no avatar [Slater93, Heeter92, Welch96r]. In our own experience with the Walking Experiment demo, we have noted some interesting user comments that have led us to hypothesize that a realistic avatar will improve presence over a generic avatar. Usoh concludes, "Substantial potential presence gains can be had from tracking all limbs and customizing avatar appearance [Usoh99]."

Providing realistic avatars requires capturing the participant's motion and rendering the participant's form and appearance. Further, we often desire the avatar to be the primary mechanism through which the user interacts with the VE.

In general, existing VE systems attach extra trackers to the participant for sensing changing positions to drive an articulated stock avatar model. The human body has many degrees of freedom of movement. Further, there are large variances in shape and appearance between people. Presenting a visually accurate representation of the participant's shape and pose is difficult due to the human body's deformability. For example, observe the dramatic changes in the shape of your hand and arm as you grasp and open a twist-lid jar. Articulated models of the human form lack the required flexibility for these intricate shape changes, and developing and controlling models that have the required deformability is difficult.

Other than shape, appearance is another important characteristic of the human form for avatars. Matching the virtual look to the physical reality is difficult to do dynamically, though commercial systems are becoming available that generate a personalized avatar. With the AvatarMe™ system, participants walk into a booth where four images are taken [Hilton00]. Specific landmarks, such as the top of the head, tip of the hands, and armpits, are automatically located in the images. These points are used to deform stock avatar model geometry and then the images are mapped onto the resulting model. The personalized avatars can then be used in any VE, including interactive games and multi-user online VEs.

We have seen how important having an avatar is, but we will examine a popular VE to help identify common issues in providing good, articulated avatars.

The Walking > Virtual Walking > Flying, in Virtual Environments project, the Walking Experiment, by Usoh, *et al*., uses additional limb trackers to control the motion of a stock avatar model [Usoh99]. The avatar model in that VE was the same for all participants. It was gender and race neutral (gray in color), and it is wearing a blue shirt, blue pants, and white tennis shoes. We have observed participants comment:

- "Those are not my shoes."
- "I'm not wearing a blue shirt."
- (From an African-American teenager) "Hey, I'm not white!"

These comments sparked our investigation to see whether representing participants with a visually faithful avatar would improve the effectiveness of the VE experience.

This Walking Experiment VE has been demoed over two thousand times, yet a version with an articulated tracked avatar (tracking an additional hand or a hand and two feet) has only been shown a handful of times [Usoh99]. The reasons for this include:

- The time required to attach and calibrate the trackers for each person decreased the number of people who could experience the VE.

- The increase in system complexity required more software and hardware for both running and maintaining the VE.
- The increase in encumbrance with the wires and tethers for the trackers made the system more prone to equipment failure.

So even with a system capable of providing tracked avatars, the additional hardware might make it infeasible or undesirable to present the more elaborate experience for everyone.

Avatar Research. Current research is trying to understand the effects of avatars on the experience in a VE. Specifically:

- What makes avatars believable?
- Given that we wish the avatar to represent certain properties, what parts of avatars are necessary?

Avatars are the source of many different types of information for VE participants, and researchers are trying to identify what components of avatars are required for increased presence, communication, interaction, etc. Non-verbal communication, such as gestures, gaze direction, and pose, provide participants with as much as 60% of information gathered in interpersonal communication. What properties should one choose to have the avatar represent? Thalmann details the current state and research challenges of various avatar components, such as rendering, interaction, and tracking [Thalmann98].

Recent studies suggest that even crude avatar representations convey substantial information. In a study by Mortensen, *et al*., distributed participants worked together to navigate a maze while carrying a stretcher. The participants were represented with very low quality visual avatars that only conveyed position, orientation, a hand cursor, and speech. The study investigated how participants interacted and collaborated. Even with crude avatar representations, participants were able to negotiate difficult navigational areas and sense the mood of other participants [Mortensen02].

Slater, *et al*., have conducted studies on the effects and social ramifications of having avatars in VEs [Slater94, Maringelli01]. They are interested in how participants interact with virtual avatars and the similarities with real human interaction. One early study compared small group behavior under three conditions: fully immersive VE, desktop (computer screen), and real environments. In both the immersive VE and desktop conditions, participants navigated and interacted with other participants in a VE while being represented by crude avatars. With avatars, emotions such as embarrassment, irritation, and self-awareness could be generated in virtual meetings. Their research studies showed that having *some* representation of the participants in the environment was important for social interaction, task performance, and presence.

In Garau's study, she compared participant interaction when communicating with another person represented by: audio only, avatars with random gaze, avatars with inferred (tracked user eye motion) gaze, and high-quality audio/video. The results show a significant difference between conditions, with the inferred-gaze condition consistently and significantly outperforming the random-gaze condition in terms of participants' subjective responses [Garau01].

Slater's team is also exploring using avatars in working with public speaking phobias [Pertaub01] and distributed-users task interaction [Slater00, Mortensen02]. Their work points to the strong effect on sense-of-presence and VE interactivity of even relatively crude self-avatars.

2.3 Interactions in VEs

Overview. Interacting with the virtual environment involves providing inputs to, or externally setting variables in, a world model simulation. Some inputs are active, such as scaling an object or using a menu, and others are passive, such as casting a shadow in the environment.

Active inputs to the VE are usually accomplished by translating hardware actions, such as button pushes or glove gestures, to actions such as grasping [Zachmann01]. For example, to select an object, a participant

typically moves his avatar hand or selection icon to intersect the object, and then presses a trigger or makes a grasping or pinching gesture.

Passive inputs depend on incorporating real-object avatars as additional data objects in simulation systems running within the environment, such as rigid-body simulations, lighting and shadow rendering, and collision detection algorithms. Typically, these passive interactions cause the world to behave as expected as the participant interacts with the environment in the way he is used to.

VE Interaction Research. Human computer interaction researchers have studied taxonomies of the active inputs to VEs. Bowman's dissertation and Hand's survey on interaction techniques decompose actions into basic components, such as selection and translation [Hand97, Bowman97]. Some tasks, such as deleting or scaling an object do not have a real world equivalent.

Ideally, a participant should be able to interact with the virtual environment by natural speech and natural body motions. Ideally, the VE system would understand and react to expressions, gestures, and motion. How do we capture all this information, both for rendering images and for input to simulations? Human limbs are articulated with many segments; their surfaces are deformable.

The fundamental interaction problem is that most things are not real in a virtual environment. Of course, the other end of the spectrum – having all real objects – removes any advantages of using a VE such as quick prototyping, or training and simulation for expensive or dangerous tasks. The optimal combination of real and virtual objects depends on the application. Examples of a near perfect combination of real and virtual objects are flight simulators. In most state-of-the-art flight simulators, the entire cockpit is real, with a motion platform to provide motion sensations, and the visuals of the environment outside the cockpit are virtual. The resulting synergy is so compelling and effective it is almost universally used to train pilots.

Having everything virtual removes many of the important cues that we use to perform tasks, such as motion constraints, tactile response, and force feedback. Typically these cues are either approximated or not provided at all.

There has been previous work on the effect of interacting with real objects on VE graphical user interfaces (GUIs). Lindeman, *et al*., conducted a study that compared 2-D and 3-D GUI widgets and the presence of a physical interaction surface. The tasks were a slider task (match a number by sliding a pip) and a drag-and-drop task. The virtual GUI had different types of surfaces with which it was registered: a tracked real surface, a virtual surface, and a virtual surface that visually clamped the avatar when the avatar intersected with it. The difference in performance for two tasks between using the 2-D and 3-D widgets were mixed. The physical surface was significantly better than the clamped virtual surface, which was in turn significantly better than a purely virtual surface [Lindeman99].

Current Interaction Methods. Specialized devices are tracked and used to provide participant inputs and controls for the VE. Common commercial interaction devices include a tracked articulated glove that with gesture recognition or buttons (Immersion's Cyberglove [http://www.immersion.com/products/3-D/interaction/cyberglove.shtml]), tracked mouse (Ascension Technology's 6D Mouse [http://www.ascension-tech.com/products/6dmouse/]), or tracked joystick with multiple buttons (Fakespace's NeoWand [http://www.fakespacesystems.com/pdfs/FS_ss_NeoWand.pdf]). Interactions comprise motions and/or button presses.

Often a device is specially engineered for a specific type of interaction. This typically improves interaction affordances, so that the participant interacts with the system in a more natural manner. Hinckley, *et al*., augmented a doll's head with sliding rods and trackers to enable doctors to more select cutting planes for visualizing MRI data of a patient's head [Hinckley94]. Military combat simulators attach special buttons and trackers to gun replicates for training. These specialized props can be very effective for improving interaction. On the other hand, the specialized engineering work is time-consuming and often usable for only a specific set of tasks.

3. Real Object Reconstruction

This algorithm was presented at the 2001 ACM Symposium on Interactive 3-D Graphics [Lok01].

3.1 Introduction

This work presents new algorithms for object reconstruction, capturing real-object shape and appearance, and for incorporating these real-object avatars with other virtual objects. In this chapter, we present an algorithm for real-time object reconstruction.

Goal. Incorporating a real object into a hybrid environment should allow the participant to hold, move and use the real object while seeing a registered virtual representation of the real object in the virtual scene.

We have two choices for generating virtual representations of the real objects: either model the real objects off-line and then track and render them on-line, or capture and render real object shape and appearance on-line. Our approach is the latter. This requires computing new virtual representations of real objects at interactive rates.

Algorithm Overview. We present a new, real-time algorithm for computing the visual hull of real objects that exploits the tremendous recent advances in graphics hardware. Along with the Image-Based Visual Hulls work [Matusik00] cited earlier, this algorithm is one of the first for real-time object reconstruction. This algorithm requires no tracking of the real objects, and can also be used for collision detection, as is discussed in Chapter 4.

Incorporating a real object into a VE involves capturing the real object's shape and appearance to generate virtual representation. We have chosen to approximate the shape of the real objects in the scene with a visual hull. The visual hull technique is a shape-from-silhouette approach. That is, it examines only the silhouettes of a real object, viewed from *n* cameras at different locations, to make a surface approximation. The projection of *n* silhouette image "carves" space into a volume that includes the real object, and a remaining volume that does not. The intersection of the projections of the *n* silhouette images approximates the object shape. The visual hull is a conservative approach that, in the absence of error, always fully circumscribes the real object. If a 3-D point is within the real object, it is within that object's visual hull. The converse is not always true.

Depending on the real object geometry, silhouettes information alone will not define an accurate surface. Concavities, such as the insides of a cup, cannot be approximated with silhouettes, even from an infinite number of external views. Because the visual hull technique uses only silhouettes, the object's color information, which might help in determining convexity, correlations, and shadows, is not used in computing real object shape.

Terminology. Throughout, we use the following special terms and phrases:

Hybrid environment – a virtual environment that incorporates both real and virtual objects

Participant – a human immersed in a virtual environment

Real object – a physical object

Dynamic real object – a physical object that can change in appearance and shape

Virtual object – a computer graphics model of an object

Object reconstruction – generating a virtual representation of a real object

Real-object avatar – virtual representation (both shape and appearance) of a real object

Image segmentation – the process of labeling each pixel in an image as corresponding to either foreground objects (objects to be reconstructed) or background objects

Background image – stored image of a static scene that is captured during startup

Object pixel – a pixel in a camera image that corresponds to a foreground object

Background pixel – a pixel in a camera image that corresponds to a background object

Object-pixel map – an array, the same size as an input cameras images, of values that identify input camera pixels as object pixels or background pixels. The map values are 1 or 0 to indicate foreground or background input camera pixels, respectively.

Novel viewpoint – a viewpoint and view-direction for viewing the foreground objects. Usually, the novel viewpoint is a participant's viewpoint, changing in real time in response to head-tracking information.

3.2 Capturing Real Object Shape

The reconstruction algorithm takes as input *n*, live, fixed-position video camera images, identifies newly introduced real objects in the scene (*image segmentation*), and then computes a novel view of the real objects' shape (*volume-querying*). The object appearance is described in Section 3.3 and the final rendering in Section 3.4.

Image Segmentation Algorithm. We assume that the scene is made up of static background objects and foreground objects that we wish to reconstruct. The goal of this stage is to identify the foreground objects in the camera images of the scene. To do this we employ the well-known image segmentation technique of image subtraction with thresholds [Castleman96]. Each camera's view of the static background scene is captured as a *background image*. We label pixels that correspond to foreground objects as *object pixels*, and pixels that represent the static background, *background pixels*. The image segmentation process results in an object-pixel map that classifies the camera image pixels into object pixels or background pixels. Simplistically, (static background scene + foreground objects) – (static background scene) = foreground objects.

Unfortunately, the input camera images contain time-varying noise – corresponding pixels in multiple images of a static scene actually vary slightly in color over time. This is due to both mechanical noise (the cameras are not perfectly still) and electrical noise. Not taking this image color variability into account would result in many pixels being identified wrongly as a part of a foreground object. One approach for managing this color variation is to use a segmentation threshold. In each new camera image, each pixel

whose color difference from its corresponding background image pixel is greater than its corresponding threshold is labeled as an object pixel. That is, *the object-pixel map* value for that pixel is set to 1. All remaining pixels are classified as background pixels, i.e. their object-pixel map value is set to 0. This process is described by Equation 1 as follows:

Equation 1 - Image Segmentation

$$\text{For pixel } j \text{ and camera } i, 1 \le i \le n, \forall i, j \quad O_{i,j} = \begin{bmatrix} |L_{i,j} - B_{i,j}| > T_{i,j} & 1 \\ |L_{i,j} - B_{i,j}| \le T_{i,j} & 0 \end{bmatrix}$$

Where

$L_{i,j}$ – Pixel j of the source image for camera i (x x y resolution) [pixels]

$O_{i,j}$ – Pixel j of the object-pixel map for camera i (x x y resolution) [pixels]

$B_{i,j}$ – Pixel j of the background image for camera i (x x y resolution) [pixels]

$T_{i,j}$ – Pixel j of the segmentation threshold map for camera i (x x y resolution) [pixels]

As the noise in a static scene can vary spatially across an image, we set segmentation threshold values on a per-pixel basis. The *segmentation threshold map* is an array of statistically-based threshold values that characterizes the noise of the background image for a camera. Background image pixels that correspond to high-frequency edges or areas in the scene will have higher variation because of camera vibration. Too high a threshold value results in missed object pixels, and so we tried to minimize high spatial frequency portions in the background images by draping dark cloth over most surfaces.

Image segmentation returns results that are sensitive to shadows, changes in lighting, and image noise. For example, altering the lighting without capturing new background images would increase errors in image segmentation. We attempted to keep the lighting constant. We did not attempt to identify or filter out real object shadows, but we used diffuse lighting so shadows would not be sharp.

Image Segmentation Implementation. At initialization, five frames of the background scene are captured for each camera. These images are averaged to compute a background image. To compute a camera's segmentation threshold map, we take the maximum deviation from the average as a segmentation threshold

value on a per-pixel basis. We found that five images of the static background were sufficient to calculate useful background images and segmentation threshold maps.

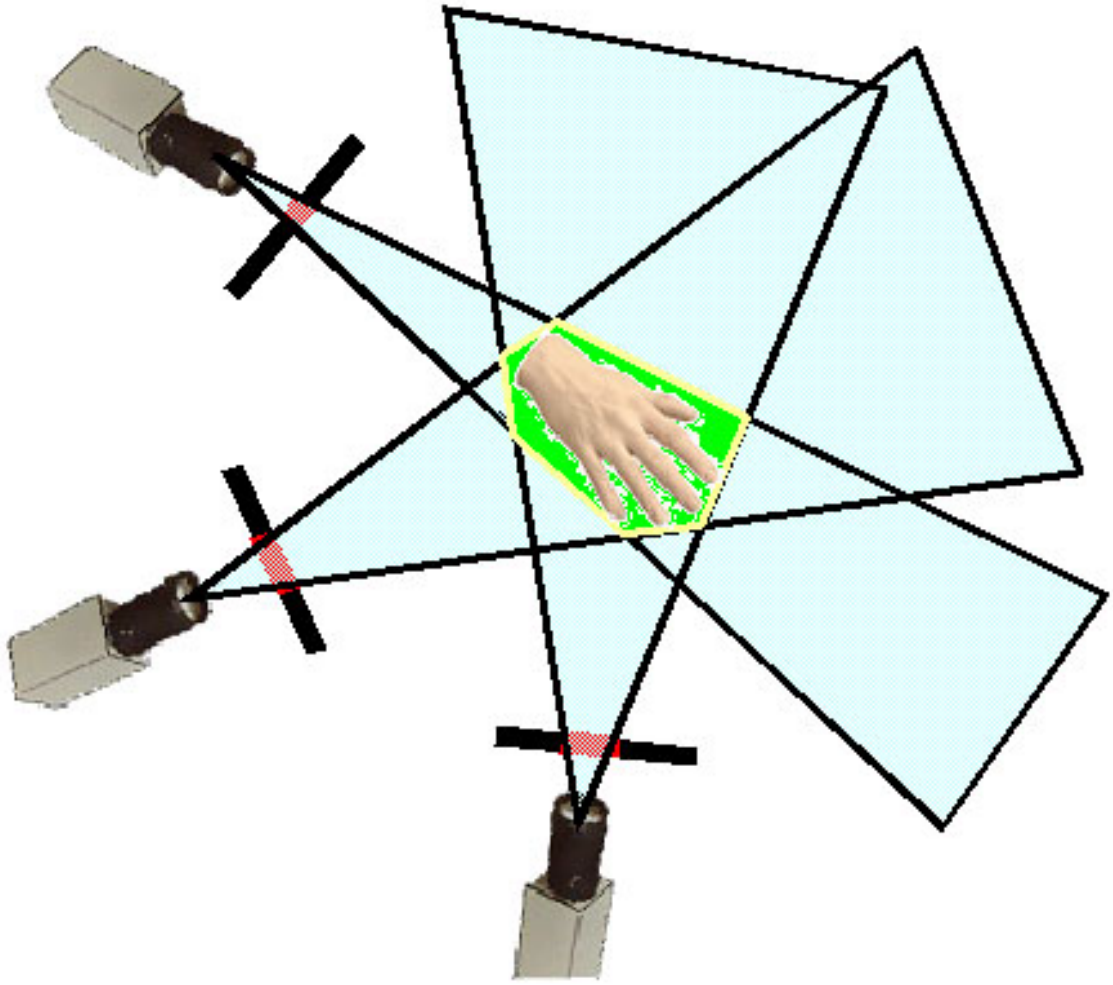
Figure 3 - Frames from the different stages in image segmentation. The difference between the current image (left) and the background image (center) is compared against a threshold to identify object pixels. The object pixel map is stored in the alpha channel. For the right image, we cleared the color component of background pixels to help visualize the object pixels.

Image segmentation with thresholds is essentially the same as Chromakeying, a standard technique for separating foreground objects from a monochromatic background, used in television and movies.

The object pixel maps are stored in the alpha channel of the corresponding camera image. Object pixels have an alpha of 1 (full opacity), and background pixels have an alpha of 0 (full transparency).

***Volume-querying* Algorithm.** Given the *n* object-pixel maps from the *n* input images, we want to compute the visual hull [Laurentini94] of the real objects. Recall that the set of object pixels for one camera represents the projection of a real object onto that camera's image plane. The projection of an object pixel is a pyramidal volume. The projection of an object pixel map is a collection of pyramidal volumes. The visual hull is the intersection of projections of all *n* object pixel maps, as shown in Figure 4. Computing the intersection requires testing the projection of each object pixel from a camera against the projection of object pixels from all the other cameras.

Figure 4 – The visual hull of an object is the intersection of the object pixel projections.



The reconstruction volume is the intersection of all the cameras' frusta, and it is the only part of each frustum's volume where object pixel pyramidal volumes can intersect. The number of intersection tests grows linearly with the number of cameras and with the square of the resolution of the cameras. For example, with 3 NTSC cameras, there could be up to $(720*486)^2 * 2 = 2.45 * 10^{11}$ pyramid-pyramid intersection tests per frame. The computational complexity is further discussed in Section 3.5.

In general we want to render the visual hull from a novel viewpoint, a view that is different from that of any of the cameras. To do this, we use a method we call *volume-querying*, a variation on standard techniques for volume definition given boundary representations [Kutulakos00].

Volume-querying asks, *Given a 3-D point (P), is it within the visual hull (VH) of a real object in the scene?* *P* is within the visual hull iff for each camera *i*, $1 \leq i \leq n$, *P* projects onto an object pixel *j* for camera *i*. This equation is represented in Equation 2 as follows.

Equation 2 – Volume-querying

$$\boldsymbol{P \ni \sim VH_{object} \text{ iff } \forall\ i, \ni\ j \text{ such that } O_{i,j} = C_{m,i} * P,\ O_{i,j} = 1}$$

Where

$\sim VH_{object}$ – (calculated) Visual hull of the real object

$O_{i,j}$ – Pixel *j* of the object-pixel map for camera *i* (*x* x *y* resolution) [pixels]

P – a 3-D point (3 x 1 vector) [meters]

and where each camera *i* defined by its extrinsic parameters {C_t translation (3 x 1 vector) and C_r rotation (3 x 3 matrix)}, its intrinsic parameters {C_d radial distortion (scalar), C_{pp} principal point (2 x 1 vector), C_f focal lengths (2 x 1 vector)}, and its C_s resolution (*x* x *y*). $C_{m,i}$ is the projection (4 x 4 vector) matrix given the camera's extrinsic and intrinsic parameters.

For rendering the visual hull from a *novel viewpoint*, we volume-query a sampling of the view frustum volume. This is in effect asking, which points in the novel view volume are within the visual hull? This sampling is done at two levels. The view frustum is sampled by a discrete set of planes swept through it from front to back, along the novel view direction. Each plane is in turn sampled at view image resolution by rendering, reprojected with standard graphics hardware, each of the cameras' object pixel maps and using a standard stencil buffer to detect *n* object pixel projection intersections, indicating that P is within the visual hull.

Accelerating Volume-querying with Graphics Hardware. We use the graphics-hardware-accelerated functions of projected textures, alpha testing, and stencil testing in conjunction with the depth buffer, stencil buffer, and frame buffer for performing intersection tests. We want to generate a view of the visual hull from the same novel view (viewpoint, view direction, and field of view) from which the virtual environment is rendered. For a *u* x *v* resolution viewport into which the visual hull is rendered, we use the

following graphics hardware components, which are standard on commodity graphics chipsets such as the nVidia GeForce4, SGI Infinite Reality 3, and ATI Radeon:

- **frame buffer –** u x v array of color values of the first-visible surface of the visual hull. Each element in the frame buffer has four values: red, green, blue, and alpha.
- **depth buffer –** u x v array of depth values from the eye viewpoint to the first-visible surface of the visual hull.
- **stencil buffer** – u x v array of integer values. The stencil buffer is used to store auxiliary values and has basic arithmetic operations such as increment, decrement and clear. The stencil buffer is used to count object pixel projection intersections during volume-querying.
- **texture map –** x x y array of color values that holds camera images and object pixel maps.
- **projected textures** – generates texture coordinates for a graphics primitive, such as a triangle, by multiplying the vertex position by the texture matrix.
- **alpha testing** – determines whether to render a textured pixel based on a comparison against a reference alpha value.
- **stencil testing** – determines whether to render a pixel based on a comparison of the pixel's stencil value against a reference stencil value.

Using the results of image segmentation, each camera's image, with the corresponding object-pixel map in the alpha channel, is loaded into a texture. The camera image color values are not used in generating object shape. Section 3.3 discusses how the image color values are used for deriving object appearance.

Volume-querying a point. First we discuss using the graphics hardware to implement volume-querying for a single point, and then we extend the explanation to larger primitives. For any given novel view V (with perspective matrix M_V) and n cameras, we want to determine if a 3-D point P is in the visual hull. For notation, P (upper case) to represent the 3-D point that projects onto 2-D pixel p (lower case) in the desired novel view image plane. Equation 2 states that for P to be within the visual hull, it must project onto an object pixel in each camera, i.e., $O_{i,j} = 1$ for $C_{m,\,i} * P$. Conversely, when rendering P with projected

camera textures, *P* must be textured with an object pixel from each camera, or it is not within the visual hull.

Rendering a textured point *P* involves

- Transforming the 3-D point *P* into 2-D screen space *p*,
- Indexing into the 2-D texture map for the texel that projects onto the *P*,
- Writing the texel color to the frame buffer, if the alpha value for $p = 1$.

To perform this operation, *P* is rendered *n* times. When rendering *P* for the *i*th time, camera *i*'s texture is used, and the texture matrix is set to the camera *i*'s projection matrix ($C_{m,i}$). This generates texture coordinates for *P* that are a perspective projection of image coordinates from the camera's location. Because the object pixel map is loaded into the alpha channel (see above), a texel is only applied if it corresponds to an object pixel.

The stencil buffer value for *p* is used to count the number of cameras whose object pixels texture *P* as a result of the above. The stencil buffer value is initialized to 0. Since only texels with an alpha of 1 can texture a point, if *P* is textured by camera *i*, it means *P* projected onto an object pixel in camera *i* ($P = C_{m,i}^{-1} O_{i,p}$), and *p*'s stencil buffer is incremented by 1.

Once all *n* textures are projected, *p*'s stencil buffer will contain values in the range [0, *n*]. We want to keep *p* as part of the virtual representation, i.e., within the visual hull, only if its stencil value is equal to *n*. To do this we change the stencil test to clear *p*'s stencil buffer and frame buffer values if *p*'s stencil value < *n*.

Since *P* is rendered from the novel view, *p*'s depth buffer (*z-buffer*) value holds the distance of *P* from the novel viewpoint. The frame buffer holds the color value on the nearest volume-queried point on the visual hull, which is an automatic result of the foregoing operation. We discuss different approaches to coloring later.

Volume-Querying a 2-D Primitive. We now extend the idea of volume-querying to 2-D primitives, such as planes. To render the visual hull from a novel viewpoint, we want to volume query all points within the volume of the view frustum. As this volume is continuous, we sample the volume with a set of planes perpendicular to the view direction, and completely filling the reconstruction viewport. Instead of volume-querying one point at a time, the volume-querying is done on the entire plane primitive. The set of planes are volume-queried from front to back. This is similar to other plane sweep techniques [Seitz97].

To perform volume-querying on a plane using graphics hardware, the plane is rendered n+1 times, once with each camera's object-pixel map projected onto it, and once to keep only pixels with a stencil buffer value = n. Pixels with a stencil value of n correspond to points on the plane that are within the visual hull. The frame buffer, depth buffer, and stencil buffer are not cleared between planes. The resulting frame buffer and depth buffer describe the volume-sampled first visible surface of the visual hull from the novel viewpoint. Equation 3 represents how the algorithm generates the virtual representation's shape.

Equation 3 – Plane sweeping

$$p \ni \exists k \sum_{k=near;k=k+PS}^{far} f(U,S,k)$$

Where

PS – Spacing between planes for plane sweep volume-querying [meters]

U – Estimated pose of the user (Tracker report for position and orientation, field of view, near plane, far plane)

S – Novel view screen resolution (u x v) [pixels]

f(U,S,k) – generates a plane rectangle that fully fills the viewport a distance k from the estimated user's pose

The spacing (PS) and number (far - $near$ / PS) of the planes are user-defined. In practice, given the resolution and location of our input cameras, we sample the volume with PS = 1.5 centimeter spacing between planes throughout the participant's view frustum.

In practice, the camera images contain non-linear distortions that the linear projected-texture hardware cannot process. Not taking into account these intrinsic camera parameters, such as radial distortion, focal length, and principal point, will result in an object pixel's projection not sweeping out the same volume in virtual space as in real space. Instead of using the projected texture hardware, the system computes undistorted texture coordinates. Each plane is subdivided into a regular grid, and the texture coordinates at the grid points are undistorted through pushing the image coordinates through the intrinsic camera model as discussed in [Bouguet98]. Our full camera model and corresponding scene point transformation is described by Equation 4 as follows.

Equation 4 – Camera Model and Scene Point Transformation

$$P_{camera\_space} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = C_r \bullet P + C_t$$

$$p_{normalized\_pinhole\_projection} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \end{bmatrix}$$

$$p_{distorted} = \begin{bmatrix} x_d \\ y_d \end{bmatrix} = (1 + C_r{}^2) * \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$

$$p_{final} = \begin{bmatrix} C_f(1) & C_f(1) & C_{pp}(1) \\ 0 & C_f(2) & C_{pp}(2) \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_d * C_f(1) + C_{pp}(1) \\ y_d * C_f(2) + C_{pp}(2) \\ 1 \end{bmatrix}$$

Where

P – a 3-D point (3 x 1 vector) [meters]

p – 2-D projection of P (2 x 1 vector)

and each camera i is defined as described earlier in Equation 2.

Although the texture is still linearly interpolated between grid points, we have observed that dividing the plane into a 5 x 5 grid and undistorting the texture coordinates at the grid points reduces error in visual hull shape, and since the overall algorithm performance is not transformation-load bound, reconstruction performance is not hampered by making this distortion correction.

OpenGL Psuedocode.

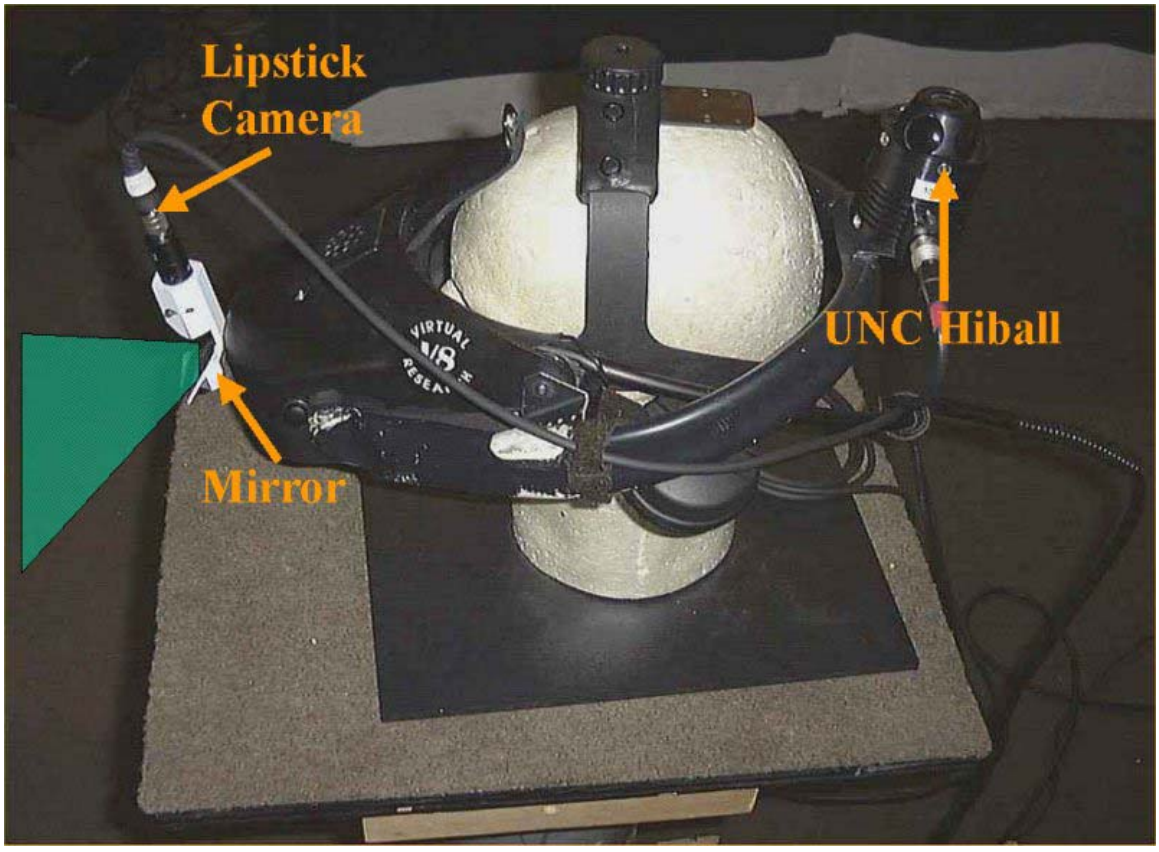
The following is psuedocode for the plane sweeping and volume-querying stages, given the image segmented camera images already loaded as textures.

```
//Enable the alpha test so we only texture object pixels
glEnable( GL_ALPHA_TEST );
glAlphaFunc( GL_GREATER, 0.0 );

//Turn on the stencil test
glEnable( GL_STENCIL_TEST );

//Since the stencil buffer keeps relevant pixels front-to-back, it
performs implicit z-testing
glDepthFunc( GL_ALWAYS );

//Enable texturing
glEnable( GL_TEXTURE_2-D );

//Sweep planes from near to far
for ( fPlane = fNear; fPlane < fFar; fPlane += fStep )
{
      //Stencil operations are set to increment if the pixel is
      //textured
      glStencilOp( GL_KEEP, GL_KEEP, GL_INCR );

      //For all cameras we draw a projected texture plane
      for each camera i
      {
            //The test function is updated to draw only if a stencil
            //value equals the number of cameras already drawn

            glStencilFunc( GL_EQUAL, i, ~0 );

            //Bind the camera i's current texture
            glBindTexture( GL_TEXTURE_2-D, camera i's texture );

            //Draw the plane
            DrawPlane();
      }
      //We want to keep only pixels with a stencil value equal
      //to iNumCameras
      glStencilFunc( GL_GREATER, iNumCameras, ~0 );

      //Zero everything else
      glStencilOp( GL_KEEP, GL_ZERO, GL_ZERO );

      glBindTexture( GL_TEXTURE_2-D, WHITE );
      DrawPlane();
}
```

3.3 Capturing Real Object Appearance

Appearance for a Head-Tracked Point of View. Volume-querying only captures the real object's shape. Since we were generating views of the real objects from the participant's perspective, we wanted to capture the real object's appearance from the participant's dynamic (head-tracked) point of view. This is the special case that the novel view = participant's view. A lipstick camera with a mirror attachment was mounted onto the head-mounted display (HMD), as seen in Figure 5. Because of the geometry of the fixture, this camera had a virtual viewpoint and view direction that is quite close to the participant's viewpoint and view direction. We used the image from this camera for texturing the visual hull.

Figure 5 – Virtual Research V8 head-mounted display (HMD) with UNC HiBall optical tracker and lipstick camera (with an overlaid field of view) attached to the front mounted mirror.



This particular camera choice finesses a set of difficult problems of computing the correct pixel color for the visual hull, which involves accounting for visibility and lighting. Choosing the correct pixel color given only the input camera images is difficult, because each visual hull surface point could project onto

pixels with differing color in the camera images. This is especially true if other real objects were obscuring the vector from a visual hull point to a camera, such as if the participant were to hold his hand a few inches above a block. Given proper camera placement, viewing the scene from the side would reconstruct the top of the block, but the block will project onto different colors in the different camera images. Visibility testing is the resolution of this problem by using obscuration information.

Appearance for a General Point of View. For the more general case where the novel view is not the same as the participant's view, then data from the camera images are used to color the visual hull. Since our algorithm does not build a traditional model, computing color and visibility per pixel is expensive and not easily supported.

We implemented two approaches for coloring the first visible surface of the visual hull for the general view case. The first approach blended the camera textures during plane sweeping. During plane rendering, each texture was given a blend weighting, based on the angle between each camera's view direction and the normal of the plane. The results have some distinct texturing artifacts, such as incorrect coloring, textures being replicated on several planes, and noticeable texture borders. These artifacts were due to our not computing visibility, to visual hull sampling, and to the differences in shape between the real object and the visual hull.

The second approach generated a coarse mesh of the reconstruction depth buffer. We assume the camera that most likely contributed to a point's color is that with a view direction closest to the mesh's normal. For each mesh point, its normal is compared to the view directions of the cameras. Each vertex gets its color from the camera whose viewing direction most closely matches its normal. The process was slow and the result still contained artifacts.

Neither of our two approaches returns a satisfactory general viewpoint coloring solution. The Image Based Visual Hulls algorithm by Matusik computes both the model and visibility and is a better suited for reconstruction from novel viewpoints other than the participant's [Matusik00, 01].

3.4 Combining with Virtual Object Rendering

During the plane-sweeping step, the planes are rendered and volume-queried in the same coordinate system as the one used to render the virtual environment. Therefore the resulting depth buffer values are correct for the novel viewpoint. In the absence of error, rendering the virtual objects into the same frame buffer and depth buffer correctly resolves occlusions between real objects and virtual objects based on depth from the eye. The real-object avatars are visually composited with the virtual environment by the rendering process.

Combining the real-object avatars with the virtual environment must include the interplay of lighting and shading. For real-object avatars to be lit by virtual lights, a polygon mesh of the reconstruction depth buffer values is generated. The mesh is then rendered with the OpenGL lighting. The lit vertices are then modulated with the HMD camera texture through using OpenGL blending. We can also use standard shadowing algorithms to allow virtual objects to cast shadows on the real-object avatars [Foley96].

Shadows of a real-objects avatar on virtual objects can be calculated by reconstructing the real object from the light source's viewpoint. The resulting depth buffer is converted into a texture to shadow VE geometry.

3.5 Performance Analysis

The visual hull algorithm's overall work is the sum of the work of the image segmentation and volume-querying stages. This analysis does not take into account the time and bandwidth costs of capturing new images, transferring the image data between processors, and the rendering of the virtual environment.

The process of rendering a primitive has two main computational requirements, a transformation stage that converts the primitive's 3-D vertices into screen space 2-D points, and a rasterization stage that converts these 2-D points to pixels and writes the data to the frame buffer. The number of triangles that are transformed from world space to screen space is the *transformation load*, and the number of frame buffer pixels that need to be written to is the *fill rate load*.

The image segmentation work is composed of determining object pixels. Each new camera image pixel is subtracted from a background pixel and the result compared against a segmentation threshold value at every frame. Given n cameras with u x v resolution, $u*v*n$ subtract and compares are required.

The volume-querying work has both a transformation load and a fill rate load. For n cameras, rendering l planes with u x v resolution and divided into an i x j camera-distortion correction grid, the geometry transformation work is $(2(n*i*j)+2)*l$ triangles per frame. Volume-querying each plane computes $u * v$ point volume-queries in parallel. Since every pixel is rendered $n+1$ times per plane, the *fill rate load* = $(n+1)*l*u*v$ per frame.

Figure 6 – Geometry transformations per frame as a function of number of cameras planes (X) and grid size (Y). The SGI Reality Monster can transform about 1 million triangle per second. The nVidia GeForce4 can transform about 75 million triangles per second.

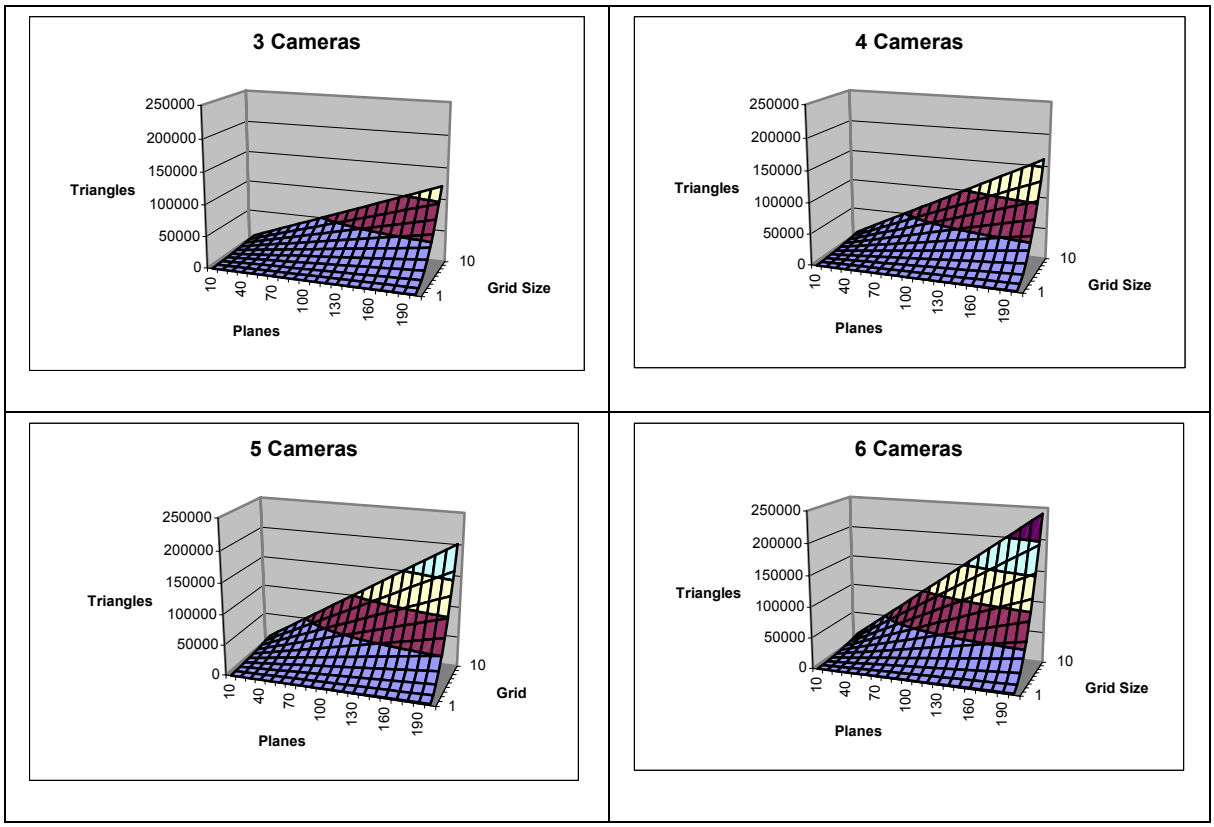
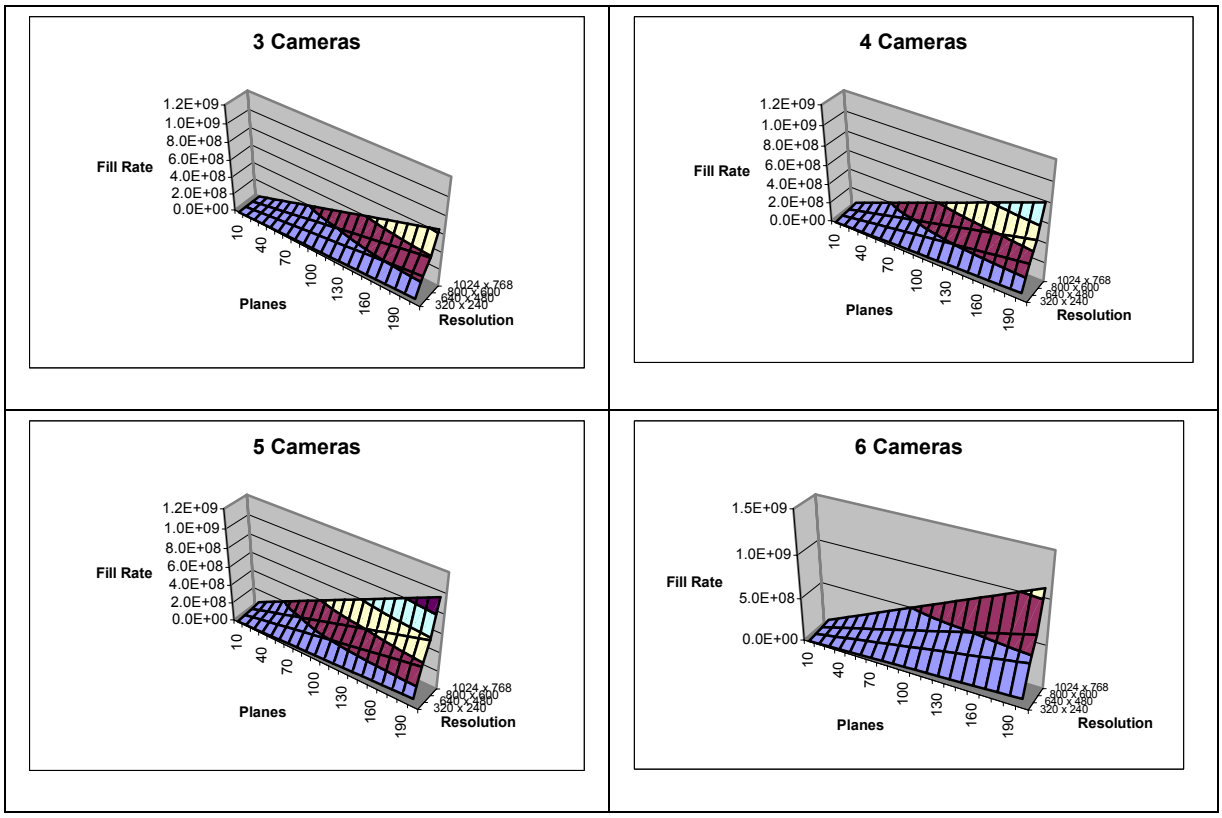


Figure 7 – Fill rate as a function of number of cameras, planes (X) and resolution (Y). The SGI Reality Monster has a fill rate of about 600 million pixels per second. The nVidia GeForce4 has a fill rate of about 1.2 billion pixels per second.



For example, with:

$n = 3$

x x y = 720 x 243 (single field NTSC cameras) – 30 frames per second

$far - near$ = 1 m

PS = 1 cm

u x v = 320 x 240 window

reconstruction rate of fifteen frames per second

the image segmentation does $15.7 * 10^6$ subtracts and segmentation threshold tests per second, $0.23 * 10^6$ triangles per second are perspective-transformed, and the fill rate must be $0.46 * 10^9$ per second.

The SGI Reality Monster can transform about $1.0 * 10^6$ triangles per second and has a fill rate of about $0.6*10^9$ pixels per second. The nVidia GeForce4 can transform about $75.0 * 10^6$ million triangles

per second and has a fill rate of about 1.2 * 10^9 pixels per second [Pabst02]. The fill rate requirements limits the number of planes with which we can sample the volume, which in turn limits the reconstruction accuracy. At 320 x 240 resolution with 3 cameras and reconstructing at 15 frames per second, on the SGI, we estimate one can use 130 planes, and on a GeForce4, 261 planes.

3.6 Accuracy Analysis

How closely the final rendered image of the virtual representation of a real object matches the actual real object has three separate components: how closely the shape matches, how closely the location matches, and how closely the appearance matches.

Sources of Error for Capturing Real Object Shape. In general, the two primary sources of error in shape between a real object and its corresponding real-object avatar are the end-to-end system latency, discussed in detail below, and the visual hull approximation of the real object's shape. Fundamental to using the visual hull approach, errors in real object shape approximation impose a lower bound on overall error, regardless of other sources of error. The difference in shape between the visual hull and the real object are covered in [Niem97]. For example, a 10 cm diameter sphere, viewed by three cameras located 2 meters away on the three primary axes, would have some points approximately 1.26 cm outside the sphere be within the sphere's visual hull. For objects with convexity or poorer camera placement, the error would be greater.

The final image, I_{final}, is determined from a sample point set in 3-space, located on a set of planes. Equation 5 shows that the final image is a combination of three primary components, the image segmentation (Equation 1), volume-querying (Equation 2), and visual hull sampling (Equation 3).

Equation 5 - Novel view rendering of the visual hull

$$I_{final} = \forall p, p \ni \exists k \sum_{k=near;k=k+PS}^{far} f(U,S,k), \text{such that } p = C_{m,i} \bullet P, \forall i \ni j\left(p = \left|L_{i,j} - B_{i,j}\right| > T_{i,j}\right)$$

Where:

I_{final} – Novel view of the visual hull of the real object

P – a 3-D point (3 x 1 vector) [meters]

p – 2-D projection of P (2 x 1 vector)

PS – Spacing between planes for plane sweep volume-querying [meters]

U – Estimated pose of the user (Tracker report for position and orientation, field of view, near plane, far plane), U_m is the projection matrix defined by the user's pose.

S – Novel view screen resolution (u x v) [pixels]

$L_{i,j}$ – Pixel j of the source image for camera i (x x y resolution) [pixels]

$B_{i,j}$ – Pixel j of the background image for camera i (x x y resolution) [pixels]

$T_{i,j}$ – Pixel j of the segmentation threshold map for camera i (x x y resolution) [pixels]

and camera i is defined as described earlier with Equation 2. There are three kinds of error for I_{final}, errors in shape, location, and appearance.

Image Segmentation. Here is the equation for image segmentation again (Equation 1). For pixel j, camera i

$$\forall i,j \quad O_{i,j} = \begin{bmatrix} \left|L_{i,j} - B_{i,j}\right| > T_{i,j} & 1 \\ \left|L_{i,j} - B_{i,j}\right| \leq T_{i,j} & 0 \end{bmatrix}$$

The errors in the image segmentation for a pixel come from three sources:

1) The difference between foreground object color and the background color is smaller than the segmentation threshold value.

2) The segmentation threshold value is too large, and object pixels are missed – commonly due to high spatial frequency areas of the background.

3) Light reflections and shadowing from foreground objects cause background pixels to differ from the background image, by greater than the segmentation threshold value.

The incorrect segmentation of pixels results in the following errors of visual hull size:

1) Labeling background pixels as object pixels incorrectly increases the size of the visual hull

2) Labeling object pixels as background pixels incorrectly reduces the size of the visual hull or yields holes in the visual hull.

Errors in image segmentation do not contribute to errors in visual hull location.

Our Experience: We reduced the magnitude of the segmentation threshold values by draping dark cloth on most surfaces to reduce high spatial frequency areas. We reduced shadows and other image segmentation errors by keeping lighting constant and diffuse, and using with foreground objects that were significantly different in color from the background. We used Sony DFW-500 cameras, and they had approximately a 2 percent color variation for the cloth-draped static scene. During implementation we also found that defining a minimum and maximum segmentation threshold per camera (generated by empirical testing) helped lower image segmentation errors.

Volume-querying. The next source of error is how closely the virtual volume that an object pixel sweeps out matches the physical space volume. This depends on the inverse of the camera matrix (C_m^{-1}) that projects pixels from each camera's image plane into rays in the world. Recall that the camera matrix is defined by the camera's extrinsic parameters {C_t translation (3 x 1 vector) and C_r rotation (3 x 3 matrix)}, intrinsic parameters {C_d radial distortion (scalar), C_{pp} principal point (2 x 1 vector), C_f focal lengths (2 x 1 vector)}, and resolution C_s (x x y).

We assume that the camera pixels are rectangular, and subject to only radial (and not higher-order) distortions. Here are the equations for the camera model (Equation 4) and volume-querying (Equation 2) again.

$$P_{camera\_space} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = C_r \bullet P + C_t$$

$$p_{normalized\_pinhole\_projection} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \end{bmatrix}$$

$$p_{distorted} = \begin{bmatrix} x_d \\ y_d \end{bmatrix} = (1 + C_r^{\,2}) * \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$

$$p_{final} = \begin{bmatrix} C_f(1) & C_f(1) & C_{pp}(1) \\ 0 & C_f(2) & C_{pp}(2) \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_d * C_f(1) + C_{pp}(1) \\ y_d * C_f(2) + C_{pp}(2) \\ 1 \end{bmatrix}$$

$P \ni \sim \mathrm{VH}_{\mathrm{object}}$ iff $\forall\ i, \ni j$ such that $O_{i,j} = C_{m,i}^{-1} * P,\ O_{i,j} = 1$

Given a camera configuration with a 1 cubic meter reconstruction volume, the primary factors that affect volume-querying accuracy are: camera rotation (C_r) and camera resolution (C_s). The projection (p) of the 3-D point (P) onto the 2-D camera image plane is sensitive to rotation error. For example, 0.1 degree of rotational error in camera position would in result in 0.58 cm error in the reconstruction volume.

$$P_{projected} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = C_r \bullet P + C_t$$

The camera resolution determines the minimum size of a foreground object to be visualized. The undistorted 2-D projection of a 3-D point is eventually rounded into two integers that reference the camera's object-pixel map. This rounding introduces error into volume-querying. Our cameras are located such that the largest distance from any camera to the farthest point in the reconstruction volume is 3.3 m. Given that we use one field of the NTSC-resolution cameras (720 x 243) with 24-degree FOV lenses, a pixel sweeps out a pyramidal volume with at most a base 0.58 cm by 0.25 cm.

In summary, errors in camera calibration affect visual hull shape. The error in the shape of the visual hull depends primarily on the error in camera rotation. The projection of this error into the volume gives us a upper bound on the uncertainty of a volume-queried point.

The effect on visual hull location is a bit more difficult to quantify. An error in camera calibration would cause object pixels to sweep out a volume not registered with the physical space sweeping from the camera's image plane element through the lens and into the volume.

An error in a camera's calibration will shift the projection of an object pixel, but this does not necessarily change the *location* of the visual hull. The erroneous portion of the volume being swept out will be unlikely to intersect the object pixel projections from the other cameras, and thus the visual hull would only decrease (or similarly increase) in size, but not move.

For example, suppose three cameras image a 7.5 cm cube foreground object. Assume that a camera, looking straight down on the cube from 2 meters away, had a 0.1-degree rotation error about some axis. The visual hull would decrease in size by about 0.4 cm in some world-space dimension. The error in one camera's projection of the object pixels that represent the cube probably will not intersect all the other camera's projection of the object pixels that represent the cube. In summary, calibration error would not be likely to change the visual hull location, as *all* the cameras would need to have a calibration error in such a way as to shift the object pixel projections in the same world space direction.

Observations: We placed the cameras as close to the working volume as possible. To determine each camera's extrinsic parameters, we attached the UNC HiBall to a stylus and used it to digitize the camera's location and points in the camera's scene. From these points, we calculated the camera's extrinsic parameters. The HiBall is sub-millimeter-accurate for position and 0.1-degree-accurate for rotation. Our method of using the camera image to locate points to determine extrinsic parameters introduces about 1-pixel of error for the rotation parameters. One of the parameters we are unsure of the accuracy is camera position. This point corresponds to the center of projection for a camera, and is typically physically

somewhere within the camera enclosure. We only estimate this point. Assuming that there is at most 2 cm of error, which is probable given our cameras' physical dimensions, then that would introduce 2 cm of translation error and 0.05 degrees of rotation error.

To estimate the camera's internal parameters, we captured an image of a regular checkerboard pattern in the center of the reconstruction volume that took up the entire camera's field of view. Then we used the stylus again to capture specific points on the checkerboard. The digitized points were overlaid on the captured image of the checkerboard and the intrinsic parameters were hand-modified to undistort the image data to match the digitized points. The undistorted points have about 0.5 cm of error for checkerboard points (reprojecting the pixel error into 3-space) within the reconstruction volume.

This 0.5 cm estimated error for the center of the reconstruction volume is another component of the error of the results for volume-querying a point. This means there is an estimated 0.5 cm error for the edges of the visual hull shape, and an upper bound of 0.5 cm error for visual hull location, depending on the positions of cameras, and other foreground objects.

Plane Sweeping. Plane sweeping is sampling the participant's view frustum for the visual hull to generate a view of the visual hull from the participant's perspective. The UNC HiBall is attached to the HMD and returns the user's viewpoint and view direction (*U*). The tracker noise is sub-millimeter in position, and 0.1-degree in rotation. Projecting this into the space at arms-length, results in the translation contributing 0.1 cm of error, and rotation contributing 0.14 cm of error, both of which are well below the errors introduced by other factors. The screen resolution (*S*) defines the number of points the plane will volume-query (u x $\underline{v}$). At arms length distance, the 34 degree vertical FOV of the HMD has a sampling resolution of 0.2 cm. The primary factor that affects the sampling of the visual hull is the spacing between the planes (*PS*), and its value is our estimate for error from this step. Here is the equation for plane sweeping again (Equation 3).

$$p \ni \exists k \sum_{k=near;k=k+PS}^{far} f(U,S,k)$$

Where

PS – Spacing between planes for plane sweep volume-querying [meters]

U – Estimated pose of the user (Tracker report for position and orientation, field of view, near plane, far plane)

S – Novel view screen resolution (*u* x *v*) [pixels]

f(*U*, *S*, *k*) – generates a plane that fills the entire viewport, a distance *k* from the user's viewpoint

Observations: With our implementation, our plane spacing was *PS* = 1.5 cm through the reconstruction volume. This spacing was the largest trade-off we made with respects to visual hull accuracy. More planes generated a better sampling of the volume, but reduced performance.

Sources of Error for Capture of Real-Object Location. The location of the rendered VH is going to be some error distance (*E*) from the actual real object. The error can be classified as follows:

- Static Error
    - The difference between the estimated and actual locations of the participant's eyes within the HMD
- Dynamic error
    - Lack of camera synchronization
    - End-to-end system latency
    - Tracker Error for determining user pose

The error is given by Equation 6 as follows.

Equation 6 - Error between the rendered virtual representation and real object

$$E = I_{final} - \left(\widetilde{U}_m \bullet R_{object}\right)$$

Where R_{object} is the geometry of the real object. The transform from the HiBall to the participant's eyes and look-at direction varies substantially between participants. To estimate this transform, we created a test platform that included a digitized real object, and adjusted parameters until the participant felt that the virtual rendering of the real object was registered with the real object. From running several participants,

we generated a transform matrix from the reported HiBall position to the participant's eye position. We observed that for real objects at arms length, location varied in screen space by about ten pixels.

The cameras are not synchronized, and this causes reconstruction errors for highly dynamic real objects, as data is captured at times that may differ by at most one frame time. At 30 camera frames per second, this is 33 milliseconds. Because of this, the reconstruction is actually being performed on data with varying latency. Real objects that move significantly between the times the cameras' images were captured will have avatar size, shape, and location errors because each camera's object pixels would sweep out a part of the volume that the object occupied at different times. In our experience the lack of camera synchronization was not noticeable, or at least it was much smaller in magnitude than other reconstruction errors.

The end-to-end system latency was estimated to be 0.3 seconds. The virtual representation that is rendered to the participant is the reconstruction of the object's shape and appearance from the participant's perspective 0.3 seconds earlier. For objects of typical dynamics on a tabletop application, such as moving a block (~30 cm/sec), this results in the rendering of the visual hull to have up to 9 cm in translation error between the real object and the real-object avatar. The magnitude of the latency is such that our participants recognized the latency and its effects on their ability to interact with both virtual and real objects. They compensated by waiting until the real-object avatars were registered with the real object after each interaction.

The user's perspective, in addition to the real object, can also have moved since the camera source images were captured. Thus the error in the user's viewpoint and view direction (U) is coupled with the latency issue for dynamic error. The latency error contribution to errors in visual hull location is the sum of the change in the user's viewpoint and the change in real object's location over the 0.3 seconds since image capture.

Sources of Error for Capturing Real Object Appearance. We texture mapped the reconstructed shape with a camera mounted on the HMD. The front-mounted camera's image was hand-tuned with interactive sliders in the application user interface to keep the textured image registered to the real objects. We did not calibrate this front camera. Since voids around the reconstructed image are much less bothersome than losing part of the front camera texture, we set the sliders so the visible image fell slightly within the reconstructed visual hull. We do not have an estimate for the appearance discrepancy between the real object and the textured visual hull.

Error Summary. The visual hull shape error is affected by image segmentation, volume-querying, and visual hull sampling. Each pixel incorrectly labeled in the image segmentation stage results in at most a 0.5 cm error in the reconstruction volume. Camera calibration errors are typically manifested as reducing the size of the visual hull. Our estimates of using the HiBall and checkerboard pattern for calibration totals 0.5 cm of error. Finally, visual hull sampling at the 1.5 cm resolution plane spacing introduced up to 1.5 cm of error to the visual hull shape. For example, a real finger could be aligned between sampling planes and not show up at all as a reconstructed object.

The visual hull location error is affected by the camera calibration, latency, and error in the head-tracker reports. Camera calibration errors would change the visual hull location only if the errors would cause the projection of object pixels from one camera that corresponded to one foreground object to intersect with the projection of object pixels from all other cameras of other foreground objects. A much stronger affect on location error is the effect of latency, both on tracker pose and the position of dynamic objects. The visual hull presented to the user represents the location of the object from his perspective 0.3 seconds ago.

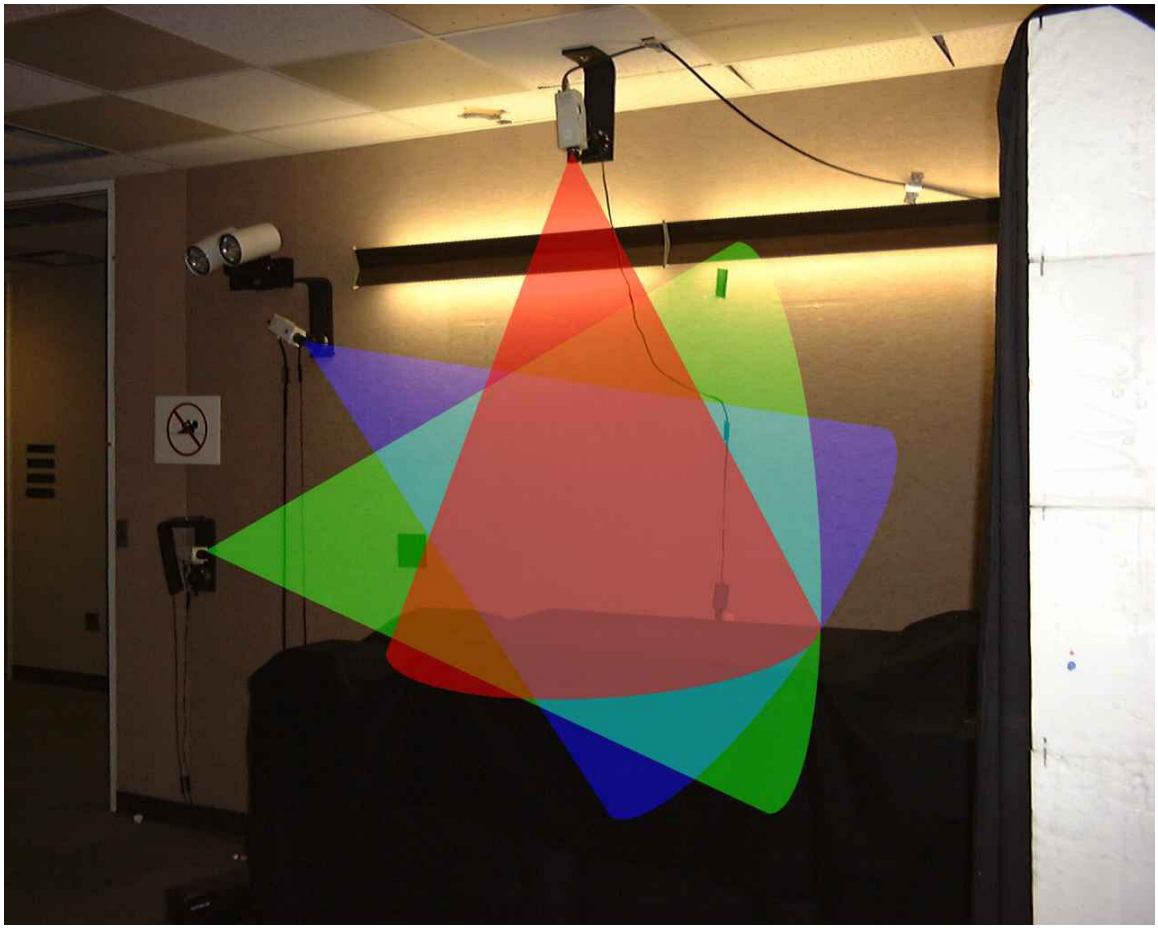
As a rough practical test, I moved my extended finger (about 1 cm in diameter) around through the entire reconstruction volume. I then examined the time-varying reconstruction width of the finger to evaluate actual error. The apparent width varied somewhat, and occasionally at the edges of the reconstruction volume, the finger would disappear. Two fingers together then gave an apparent width of about one

centimeter. Apparent finger width variation was better than our worst-case bounds, and would lead to an estimate of +/- 0.5 cm for average visual hull shape error.

3.7 Implementation

Hardware. We have implemented the reconstruction algorithm in a system that reconstructs objects within a 1.6 m x 1.3 m x 1 m volume above a tabletop, as diagramed approximately in Figure 8.

Figure 8 – The overlaid cones represent each camera's (really rectangular) field of view. The reconstruction volume is the intersection of the camera view frusta.



The system uses three fixed NTSC cameras (720 x 486 resolution) and one camera mounted on a Virtual Research V8 HMD (640 x 480 resolution) as described above. One fixed camera was mounted directly overhead, one to the left side of the table, and one at a diagonal about three feet above the table. The

placing of the fixed cameras was not optimal; the angles between the camera view directions are not as far apart as possible. Lab space and maintainability constrained this.

When started, the system captures and averages a series of five images for each camera to derive the background images and the segmentation thresholds. Since NTSC divides each frame into two fields, we initially tried having one image for each camera, updating whichever field was received from the cameras. For dynamic real objects, this caused the visual hull to have bands of shifted volumes due to reconstructing with interlaced textures. Our second approach captured one background image per field for each camera, and did reconstruction per field. Unfortunately, this caused the virtual representations of stationary objects to move. Although the object was stationary, the visual hulls defined by the alternating fields were not identical, and the object appeared to jitter. We found the simple approach of always working with the same field – we chose field zero – was a compromise. While this increased the reconstruction error, latency was reduced and dynamic real objects exhibited less shearing.

The participant is tracked with the UNC HiBall, a scalable wide-area optical tracker mounted on the HMD as shown in Figure 5 [Welch01g]. The image also shows the HMD mounted camera and mirror fixture used for coloring the reconstruction.

The four cameras are connected to Digital In – Video Out (DIVO) boards on an SGI Reality Monster system. Whereas PC graphics cards could manage the transformation and pixel fill load of the algorithm, the SGI's video input capability, multiple processors, and its high memory-to-texture bandwidth made it a better solution when development first began.

The SGI has multiple graphics pipelines, and we use five pipes: a *parent pipe* to render the VE and assemble the reconstruction results, a *video pipe* to capture video, two *reconstruction pipes* for volume-querying, and a *simulation pipe* to run simulation and collision detection as discussed in Chapter 4. First, the video pipe obtains and broadcasts the camera images. Then the reconstruction pipes asynchronously grab the latest camera images, perform image segmentation, perform volume intersection, and transfer their

results to the parent pipe. The number of reconstruction pipes is a trade-off between reconstruction latency and reconstruction frame rate, both of which increase with more pipes. The simulation pipe runs virtual simulations (such as rigid-body or cloth) and performs the collision detection and response tests. All the results are passed to the parent pipe, which renders the VE with the reconstructions. Some functions, such as image segmentation, are calculated with multiple processors.

The reconstruction is done into a 320 x 240 window to reduce the fill rate requirements. The results are scaled to 640 x 480, which is the resolution of VE rendering. The Virtual Research V8 HMD has a maximum resolution of 640 x 480 at 60 Hz.

Figure 9 – Screenshot from our reconstruction system. The reconstructed model of the participant is visually incorporated with the virtual objects. Notice the correct occlusion between the participant's hand (real) and the teapot handle (virtual).

Performance. The reconstruction system runs at 15-18 frames per second for 1.5 centimeter spaced planes about 0.7 meters deep (about 50 planes) in the novel view volume. The image segmentation takes about

one-half of frame computation time. The reconstruction portion runs at 22-24 frames per second. The geometric transformation rate is 16,000 triangles per second, and the fill rate is $1.22 * 10^9$ pixels per second. The latency is estimated at about 0.3 of a second.

The reconstruction result is equivalent to the first visible surface of the visual hull of the real objects, within the sampling resolution (Figure 9).

Advantages. The hardware-accelerated reconstruction algorithm benefits from both past and future improvements in graphics hardware. It also permits using graphics hardware for detecting intersections between virtual models and the real-objects avatars. We discuss this in Chapter 4.

The participant is free to bring in other real objects and naturally interact with the virtual system. We implemented a hybrid environment with a virtual faucet and virtual particle-system water. The participant's avatar casts shadows onto virtual objects and interacts with a water particle system from the faucet. We observed participants cup their hands to catch the water, hold objects under the stream to watch particles flow down the sides, and comically try to drink the synthetic water. Unencumbered by additional trackers and intuitively interacting with the virtual environment, participants exhibit uninhibited exploration, often doing things we did not expect.

Disadvantages. Sampling the volume with planes gives this problem $O(n^3)$ complexity, where n is the sampling resolution. Substantially large working volumes would force a tradeoff between sampling resolution and performance. We have found for 1.5-centimeter plane spacing for novel view volumes 1 meter deep, reconstruction speed is real-time and reconstruction quality is sufficient for tabletop applications.

4. Collision Detection and Other Real-Virtual Interactions

4.1 Overview

Collision detection and collision response algorithms, along with lighting and shadow rendering algorithms, improve the incorporation of real objects into the hybrid environment. For example, real-object avatars can be used as inputs to simulations to provide a more natural and realistic interface with the VE. That is, one would interact with virtual objects the same way as if the entire environment were real.

Besides including real objects in our hybrid environments visually, as was covered in Chapter 3, we want the real-object avatars to physically affect the virtual elements of the environment. For instance, as shown in Figure 10, a participant's avatar parts a virtual curtain to look out a virtual window. The interaction between the real hand and virtual cloth has two steps. First, our algorithm detects the collision between virtual objects (cloth) and real objects (hands). Then the collision resolution algorithm computes information used by the cloth simulation to compute the appropriate response at each time-step.

Figure 10 – A participant parts virtual curtains to look out a window in a VE. The results of detecting collisions between the virtual curtain and the real-object avatars of the participant's hands are used as inputs to the cloth simulation.

This chapter discusses algorithms for detecting collisions and determining plausible responses to collisions between real-object avatars and virtual objects. We define interactions as objects affecting one another. Given hybrid environments that contain both real and virtual objects there are four types of interactions we need to consider:

- Real-real. Collisions between real objects are resolved by the laws of physics; forces created by energy transfers in the collision can cause the objects to move, deform, and change direction.
- Virtual-virtual. Collisions between virtual objects are managed with standard collision detection packages, and physics simulations determine response.

- Real-virtual. For the case of real objects colliding and affecting virtual objects, we present a new image-space algorithm to detect the intersection of virtual objects with the visual hulls of real objects. The algorithm also returns data that the simulation can use to undo any unnatural interpenetration of the two objects. Our algorithm builds on the volume-querying technique presented in Chapter 3.
- Virtual-real. We do not support the case of virtual objects affecting real objects due to collisions. Whenever real-object avatars and virtual objects collide, the application modifies only the virtual objects. In making this choice, our reasoning was as follows:
  - Real-object avatars should always remain registered with the real objects.
  - Our system does not include any mechanism for applying forces to the real object, so virtual objects cannot physically affect the real objects themselves.
  - Therefore, virtual objects are not allowed to affect the real-object avatar.

4.2 Visual Hull – Virtual Model Collision Detection

Overview. Standard collision detection algorithms detect collisions among objects defined as geometric models [Lin98]. Because our system does not explicitly create a geometric model of the visual hull in the reconstruction process, we needed to create new algorithms that use camera images of real objects as input, and detect collisions between real-object avatars and virtual objects. Similar to the novel view reconstruction in Chapter 3, the collision detection algorithm uses volume querying to test the virtual object's primitives for collisions with the real-object avatars.

The inputs to our real-virtual collision detection algorithm are a set of n live video camera images and some number of virtual objects defined traditionally by geometric boundary representation primitives, triangles. Triangles are the most common representation for virtual objects, and graphics hardware is specifically designed to accelerate transformation and rendering operations on triangles. The algorithm is extendable to other representations.

The outputs of the real-virtual collision detection algorithm are:

- A set of 3-D points on the boundary representation of the virtual object in collision with a real-object avatar (CP_i).

The outputs of the collision response algorithm are estimates within some tolerance for:

- The 3-D point of first contact on the virtual object (CP_{obj}).
- The 3-D point of first contact on the visual hull (CP_{hull}).
- A 3-D recovery vector (V_{rec}) along which to translate the virtual object to move it out of collision with the real-object avatar.
- The scalar distance to move the virtual object (D_{rec}) along the recovery vector to remove CP_{obj} from collision.
- The 3-D surface normal vector at the point of first contact on the visual hull (N_{hull}).

Assumptions. A set of simplifying assumptions makes interactive-time real-virtual collision detection a tractable problem.

A**ssumption 1: Only virtual objects move or deform as a consequence of collision.** This follows from our restriction that virtual objects do not affect real objects. The behavior of virtual objects is totally under the control of the application program, so they can be moved as part of a response to a collision. We do not attempt to move real objects or the real-object avatars.

Assumption 2: Both real objects and virtual objects are stationary at the time of collision. Collision detection is dependent only upon position data available at a single instant in time. Real-object avatars are computed anew each frame. We do not keep history; consequently, no information about the motion of the real objects, or of their hulls over time, is available to the real-virtual collision detection algorithm.

A consequence of Assumption 2 is that the algorithm is unable to determine *how* the real objects and virtual objects came into collision. Therefore the algorithm cannot specify the exact vector along which to move the virtual object to return it to the position it occupied at the instant of collision. Our algorithm simply suggests a way to move it out of collision.

Assumption 3: There is at most one collision between a virtual object and a real-object avatar at a time. If the real-object avatar and virtual object intersect at disjoint locations, we apply a heuristic to estimate the point of first contact. This is due to our inability to backtrack the real object to calculate the true point of first contact. For example, virtual fork tines penetrating the visual hull of a real sphere would return only one estimated point of first contact. We move the virtual object out of collision based on our estimate for the deepest point of collision.

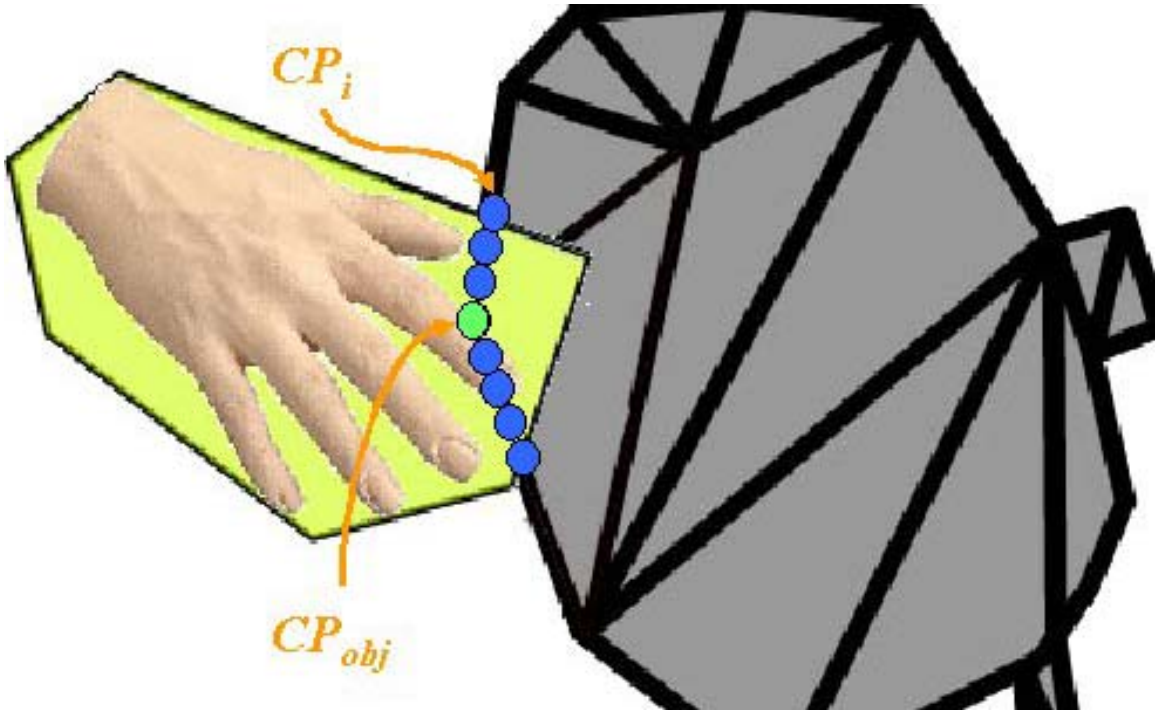
Assumption 4: The real objects that contribute to the visual hull can be treated as a single object. Although the real-object avatar may appear visually as multiple disjoint volumes, e.g., two hands, computationally there is only a single visual hull representing all real objects in the scene. The system does not distinguish between the multiple real objects during collision detection. In the example, the real oil filter and the user's hand form one visual hull. This is fine for that example – we only need to know if the mechanic can maneuver through the engine – but distinguishing real objects may be necessary for other applications.

Assumption 5: **Collisions are detected shortly after the virtual object intersects the visual hull.** This assumes the frame rate is fast compared to the motion of virtual objects and real objects. The consequence is that moving the virtual object along a vector defined in our algorithm will approximate backing the virtual object out of collision. This assumption might be violated, for example, by a virtual bullet shot into a thin sheet of real plywood. The rate of motion of the real or virtual object might be so great that the discrete time rendering might cause the collision detection to miss a collision, or the collision resolution to erroneously move the virtual object out the far side of the real object.

Approach. There are two steps for managing the interaction of virtual objects with a real-object avatar. The first and most fundamental operation is determining whether a virtual object, defined by a set of triangles representing its surface, is in collision with a real object, computationally represented by its visual hull volume.

Since we do not track the real objects, we cannot backtrack the real object to determine the exact points of first collision on the virtual object and the real object. So we only estimate the position and point of first contact of each object. The application can use additional data, such as the normal at the point of contact, or application-supplied data, such as virtual object velocity, to compute more physically accurate collision responses.

Figure 11 – Finding points of collision between real objects (hand) and virtual objects (teapot). Each triangle primitive on the teapot is volume queried to determine points at the surface of the virtual object within the visual hull (blue points).



Algorithm Overview. The algorithm first determines the *collision points*, CP_i, as shown as blue dots in Figure 11. From the set of collision points, we identify one collision point that is the maximum distance from a reference point, RP_{obj} (typically the center of the virtual object). This point is labeled the *virtual object collision point*, CP_{obj}, shown as a green dot in Figure 11.

We want to find a vector and a distance to move CP_{obj} out of collision. This is the *recovery vector*, V_{rec}, which is from CP_{obj} towards the RP_{obj}. V_{rec} intersects the visual hull at the *hull collision point,* CP_{hull}. The distance, D_{rec}, to move CP_{obj} along V_{rec} is the distance between CP_{obj} and CP_{hull}.

The final piece of data computed by our algorithm is the normal to the visual hull at CP_{hull}, if it is needed. The following sections describe how we compute each of these values. In our discussion of the algorithm, we examine the collision detection and response of a single virtual object colliding with a single real object's avatar.

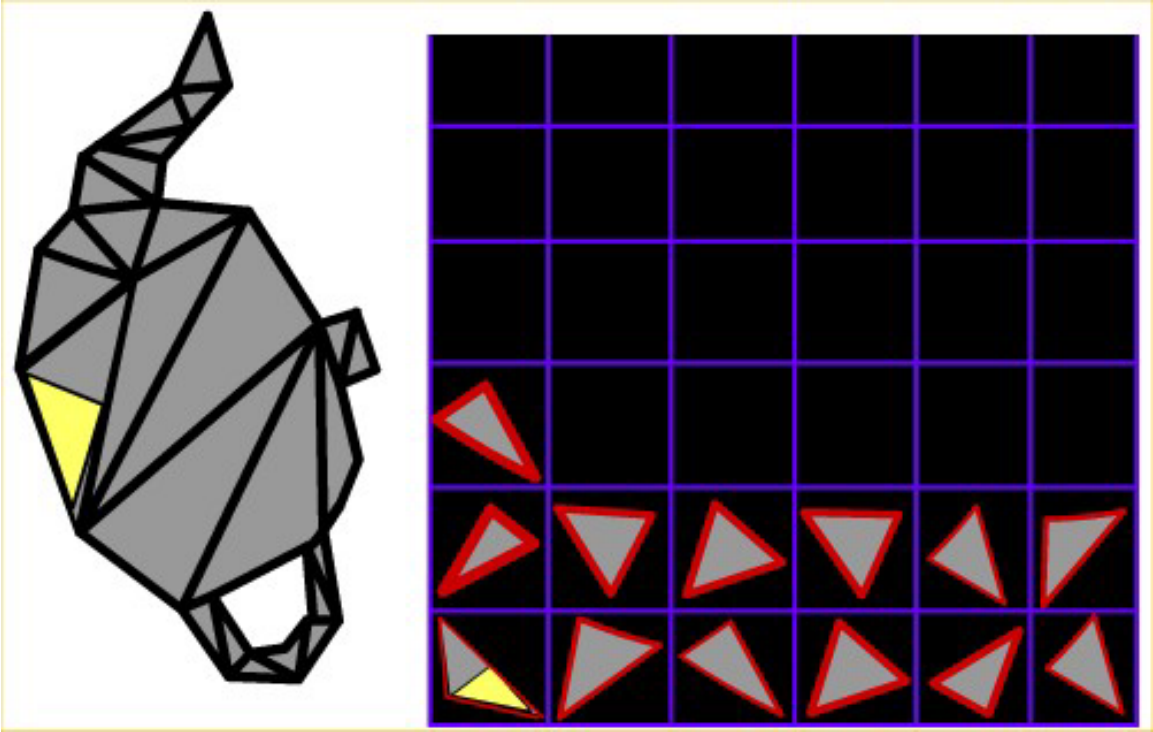
Finding Collision Points. Collision points, CP_i, are points on the surface of the virtual object that are within the visual hull of the real object. As the virtual surfaces are continuous, the set of discrete collision points is a sampling of the virtual object surface.

The real-virtual collision detection algorithm uses the fundamental ideas of volume-querying via the graphics hardware as described in Chapter 3. The visual hull is sampled with the triangles defining the surface of the virtual object, to determine if any points on the primitive are inside the visual hull. If so, the virtual object is intersecting the real-object avatar, and a collision has occurred. Note that the novel viewpoint reconstruction surface is not used in collision detection, and the real-virtual collision detection algorithm is view-independent.

As in the novel viewpoint reconstruction, the algorithm first sets up n projected textures, one corresponding to each of the n cameras and using that camera's image, object-pixel map, and projection matrix.

Volume-querying each triangle for collision detection involves rendering the triangle n times, once with each camera's object-pixel map projected as a texture (Figure 12). Pixels with a stencil value of n correspond to points on the triangle that are in collision with the visual hull.

Figure 12 – Each primitive is volume queried in its own viewport.



During each rendering pass a pixel's stencil buffer is incremented if the pixel is part of the triangle being scan converted and if that pixel is textured by a camera's object pixel. After the triangle has been rendered with all *n* projected textures, the stencil buffer will have values in the range of [0...*n*]. If, and only if, all *n* textures are projected onto a point, is that point in collision with the visual hull (Figure 4 diagrams the visual hull of an object).

The stencil buffer is read back and pixels with a stencil value of *n* represent points of collision between the visual hull and the triangle. We can find the coordinates of the 3-D point by unprojecting the pixel from screen space coordinates (u, v, depth) to world space coordinates (x, y, z). These 3-D points form a set of collision points, CP_i, for that virtual object. This set of points is returned to the virtual object simulation.

The real-virtual collision detection algorithm returns whether a collision exists and if so, a set of collision points for each triangle. How a simulation utilizes this information is application- and even object-dependent. This division of labor is similar to current collision detection algorithms. Also, like current

collision detection algorithms, e.g. [Lin98], we do not provide a suite of tools for collision response. We describe *one* option below.

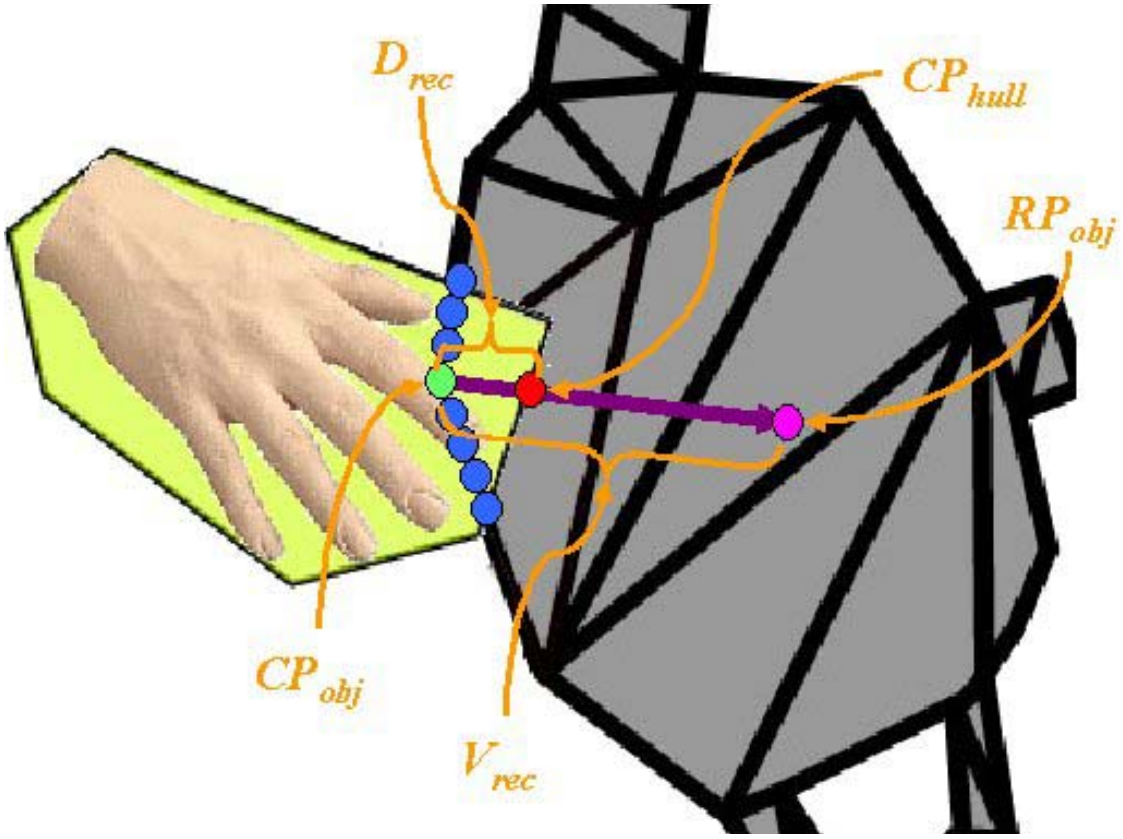
Note that if the triangle is projected 'on edge' during volume-querying, the sampling of the triangle surface during scan-conversion (getting the triangle to image space) will be sparse and collision points could be missed. For example, rendering a sphere for volume-querying from any viewpoint will lead to some of the triangles being projected on edge, which could lead to missed collisions. The size of the triangle also affects collision detection, as the volume-querying sampling would be denser for smaller triangles than larger triangles. No one viewpoint and view-direction will be optimal for all triangles in a virtual object. Thus, each triangle is volume queried in its own viewport, with its own viewpoint and view-direction.

To maximize collision detection accuracy, we wanted each triangle to fill its viewport as completely as possible. To do this, each triangle is rendered from a viewpoint along the triangle's normal, and a view direction that is the inverse of the triangle's normal. The rendered triangle is normal to the view direction, and the viewpoint is set to maximize the size of the triangle's projection in the viewport.

Recovery From Interpenetration. We present one approach for using the collision information to compute a plausible collision response for a physical simulation. The first step is to move the virtual object out of collision.

We estimate the point of first contact on the virtual object CP_{obj} to be the point of deepest penetration on the virtual object into the visual hull. We approximate this point with the collision point that is farthest from a reference point, RP_{obj}, of the virtual object. The default RP_{obj} is the center of the virtual object. As each collision point CP_i is detected, its distance to RP_{obj} is computed by unprojecting CP_i from screen space to world space (Figure 13). The current farthest point is conditionally updated. Due to our inability to backtrack real objects (Assumption 2), this point is not guaranteed to be the point of first collision of the virtual object, nor is it guaranteed to be unique. If multiple points are the same distance, we arbitrarily choose one of the points from the CP_i set for subsequent computations.

Figure 13 – Diagram showing how we determine the visual hull collision point (CP_{hull} - red point), virtual object collision point (CP_{obj} - green point), recovery vector (V_{rec} - purple vector), and recovery distance (D_{rec} - red arrow).



Recovery Vector. Given that the virtual object is in collision at our estimated point of deepest penetration CP_{obj}, we want to move the virtual object out of collision by the shortest distance possible. The vector along whose direction we want to move the virtual object is labeled the recovery vector, V_{rec} (Figure 13). Since we used the distance to RP_{obj} to estimate CP_{obj}, we define the recovery vector as the vector from CP_{obj} to RP_{obj}, as shown in Equation 7:

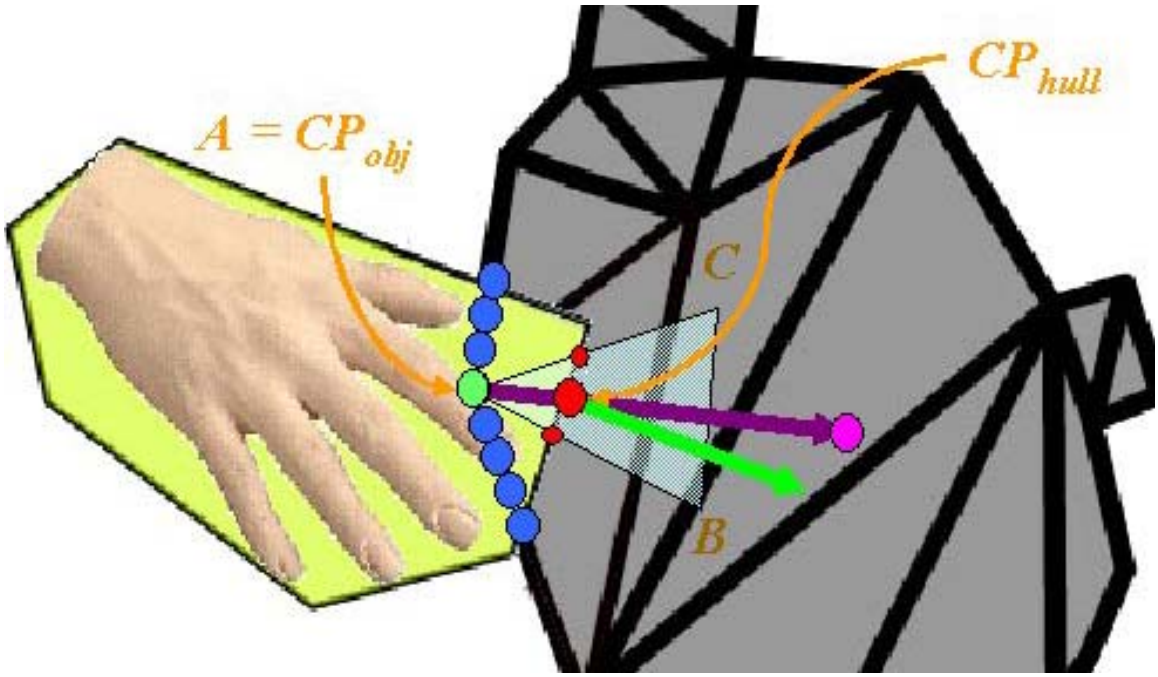
Equation 7 - Determining recovery vector

$$\boldsymbol{V_{rec} = RP_{obj} - CP_{obj}}$$

This vector represents the best estimate of the shortest direction to move the virtual object so as to move it out of collision. This vector works well for most objects, though the simulation can provide an alternate V_{rec} for certain virtual objects with constrained motion, such as a hinged door, to provide better object-specific results. We discuss using a different V_{rec} in a cloth simulation in the final section of this chapter.

Point of First Contact on Visual Hull. The recovery vector, V_{rec}, crosses the visual hull boundary at the *hull collision point*, CP_{hull}. CP_{hull} is an estimate of the point on the visual hull where the objects first came into contact. We want to back out CP_{obj} such that it coincides with CP_{hull}. Figure 14 illustrates how we find this point.

Figure 14 – By constructing triangle ABC (where A = CP_{obj}), we can determine the visual hull collision point, CP_{hull} (red point). Constructing a second triangle DAE that is similar to ABC but rotated about the recovery vector (red vector) allows us to estimate the visual hull normal (green vector) at that point.



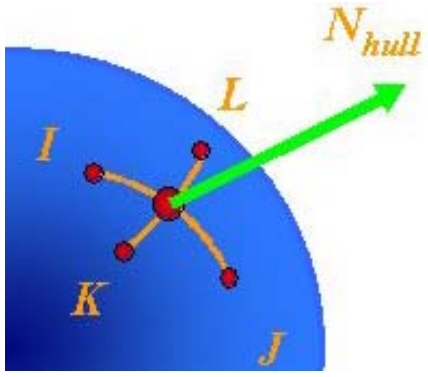
One wants to search along V_{rec} from RP_{obj} until one finds CP_{hull}. Given standard graphics hardware, which renders lines as thin triangles, this entire search can be done by volume-querying one triangle. First, we construct an isosceles triangle *ABC* such that $A = CP_{obj}$ and the base, *BC*, is bisected by V_{rec}. Angle *BAC* (at CP_{obj}) is constructed to be small (10 degrees) so that the sides of the triangle intersect the visual hull very near CP_{hull}. The height of the triangle is made relatively large in world space (5 cm) so that the base of the triangle is almost guaranteed to be outside the visual hull for our setup. Then, we volume-query the visual hull with the triangle *ABC*, rendering it from a viewpoint along the triangle's normal and such that V_{rec} lies along a scan line. The hull collision point is found by stepping along the scan line corresponding to V_{rec}, starting at the base of the triangle, searching for the first pixel (the red point in Figure 14) within the

visual hull (stencil buffer value of the pixel = n). Unprojecting that pixel from screen to world space yields the CP_{hull}. If the first pixel tested along the scan line is in the visual hull, this means the entire triangle is inside the visual hull, and we double the height of the triangle and iterate.

The recovery distance, D_{rec}, is the distance between the CP_{obj} and CP_{hull}. This is not guaranteed to be the minimum separation distance as is found in some collision detection packages [Hoff01], rather it is the distance along vector V_{rec} required to move CP_{obj} outside the visual hull and approximates the minimum separation distance.

Normal at Point of Visual Hull Collision. Some application programs require the surface normal, N_{hull}, at the hull collision point to calculate collision response (Figure 14). Our algorithm calculates this when it is requested by the application. We first locate 4 points on the visual hull surface near CP_{hull} and use them to define two vectors whose cross product gives us a normal to the visual hull at CP_{hull}. We locate the first two points, I and J, by stepping along the BA and CA of triangle ABC, finding the pixels where these lines intersect the visual hull boundary, and unprojecting the pixels to get the world space coordinates of the two points. We then construct a second triangle, DAE, identical to ABC except in a plane roughly perpendicular to ABC. We locate points K and L by stepping along DA and EA, then cross the vectors IJ and KL, to produce the normal, N_{hull} as shown in Figure 15.

Figure 15 – From volume querying *ABC* we get points *I* and *J*. From volume querying *DAE*, we find *K* and *L*. Crossing the vectors *JI* and *LK* gives us the visual hull normal N_{hull}.

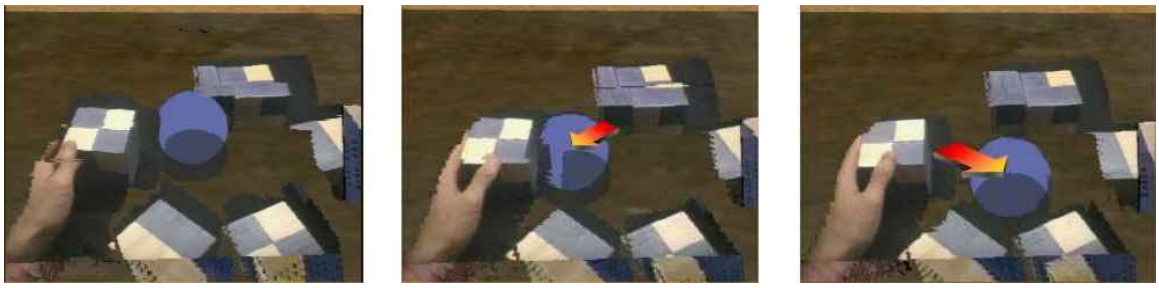


Implementation. The collision detection and response routines run on a separate *simulation pipe* on the SGI. At each real-virtual collision time-step, the simulation pipe performs the image segmentation stage to obtain the object-pixel maps.

To speed computation, collision detection is done between the visual hull and axis-aligned bounding boxes for each of the virtual objects. For any virtual object whose bounding box was reported as in collision with the visual hull, a per-triangle test is done. If collisions exist, the simulation is notified and passed the set of collision points. The simulation managing the behavior of the virtual objects decides how it will respond to the collision, including if it should constrain the response vector. Our response algorithm then computes the recovery vector and distance. The surface normal is computed if the simulation so requests.

Figure 16 includes frames taken from a dynamic sequence in which a virtual ball is bouncing around between a set of real blocks. The visual hull normal is used in the computation of the ball's direction after collision.

Figure 16 – Sequence of images from a virtual ball bouncing off of real objects. The overlaid arrows shows the balls motion between images.



4.3 Performance Analysis

Given n cameras and virtual objects with m triangles and testing each triangle in a u x v resolution viewport in a x x y resolution window, the geometry transformation cost is ($n * m$) per frame. The fill rate cost is ($n*m*u* v$)/2. There is also a computational cost of ($x*y$) pixel readbacks and compares to find pixels in collision. For example, our curtain hybrid environment, as shown in Figure 17, had 3 cameras with 720 triangles that made up the curtains. We used 10 x 10 viewports in a 400 x 400 window for collision

detection. The collision detection ran at 6 frames per second. The geometry transformation load was (3 * 720) * 6 Hz = 12,960 triangles per second. The fill rate load is (3*720*10*10)/2 * 6 Hz = 648,000 pixels per second. There are also 160,000 pixel readbacks and compares.

For collision response, the transformation cost is 2 triangles per virtual object in collision. The fill rate load is (x * y * n) = (400 * 400 * 3) = 480,000 pixels per collision.

Even though our implementation is not heavily optimized, we can achieve roughly 13,000 triangles per second for collision detection and response. This was a first implementation of the algorithm, and there are many optimizations with regards to minimizing OpenGL state changes that should improve the performance of the algorithm. The current realized performance is substantially lower than the theoretical performance possible on the SGI.

4.4 Accuracy Analysis

We assume the objects in collision are static when discussing error and sampling resolution issues. The error in visual hull location for dynamic objects, discussed in the Section 3.6, dominates any of the best-case static errors.

The collision detection accuracy depends on the accuracy of the visual hull and the resolution of the viewport into which the primitives are rendered. The error analysis for the visual hull is described in the previous chapter. We analyze the effect of viewport resolution on the collision detection accuracy. The higher the viewport resolution for collision detection, the more closely spaced the points on the triangle that are volume-queried. We assume a square viewport (having $u = v$ makes it easier on the layout of viewports in the framebuffer). For a u x v resolution viewport and a triangle bounding box of x x y size, the collision detection resolution is x / u by y / u (meters). This is the spacing between the volume queried points on the triangle surface. That is since we project each triangle such that it maximally fills the viewport (exactly half the pixels are part of the triangle), the collision detection resolution will be the longest dimension of the triangle divided by the viewport horizontal resolution.

Triangles can be rendered at higher resolution in larger viewports, producing a higher number of more closely spaced collision points and less collision detection error. Also, a higher resolution viewport means that fewer number of triangles can be volume queried in the collision detection window. If all the triangles cannot be allocated their own viewports in a single frame buffer, then multiple volume-query and read-back cycles are needed.

Hence, there is a speed-accuracy tradeoff in establishing the appropriate level of parallelism: the more viewports there are, the faster the algorithm executes but the lower the pixel resolution available for the collision calculation for each primitive. Smaller viewports have higher parallelism, but may result in missed collisions.

The size in world space of the virtual object triangles will vary substantially, but for tabletop size objects, the individual triangles would average around 2 cm per bounding box side, which would have 0.2 cm x 0.2 cm collision point detection error. For example in our sphere example (Figure 16), the virtual sphere had 252 triangles and a radius of 10 cm. The average size of a bounding box for each triangle was 1.3 cm by 1.3 cm. This would result in collision detection at a 0.13 cm/pixel resolution, which is less than the errors in visual hull location and visual hull shape. The cloth system (Figure 17) had nodes 7.5 cm x 3 cm apart. The collision detection resolution was 0.75 cm/pixel x 0.3 cm/pixel. These values determine the spatial frequency for volume-querying and bound the maximum error in finding a collision point.

For collision response, we examine the computation of the CP_{hull} point, as this impacts the distance along the recovery vector, D_{rec}, to back out the virtual object, and the uncertainty of the N_{hull} vector. We assume a square window, as we typically use the collision detection window for collision resolution. For x x y resolution collision response window and length l for the major axis of triangle ABC, the resolution accuracy for detecting CP_{hull} is l/x. Due to Assumption 5 (our frame rate is comparable to the motion of the objects), we initially set l to be 5 cm. That is we assume that there is no more than 5 cm of interpenetration. With the 400 x 400 window, this results in .0125 cm resolution for detecting CP_{hull}. If

there is more than 5 cm of penetration, we double l (doubling the size of triangle ABC) and volume query again. Again, the magnitude of errors due to sampling is substantially smaller than the error in the visual hull location (dynamic error in particular) and visual hull shape.

Thus we can estimate the following errors for the collision detection and response values, independent of any visual hull shape and visual hull location errors. We assume 2 cm virtual triangle size, 10 x 10 viewports, and 400x400 window,

Collision points (CP_i)– 0.75 cm error

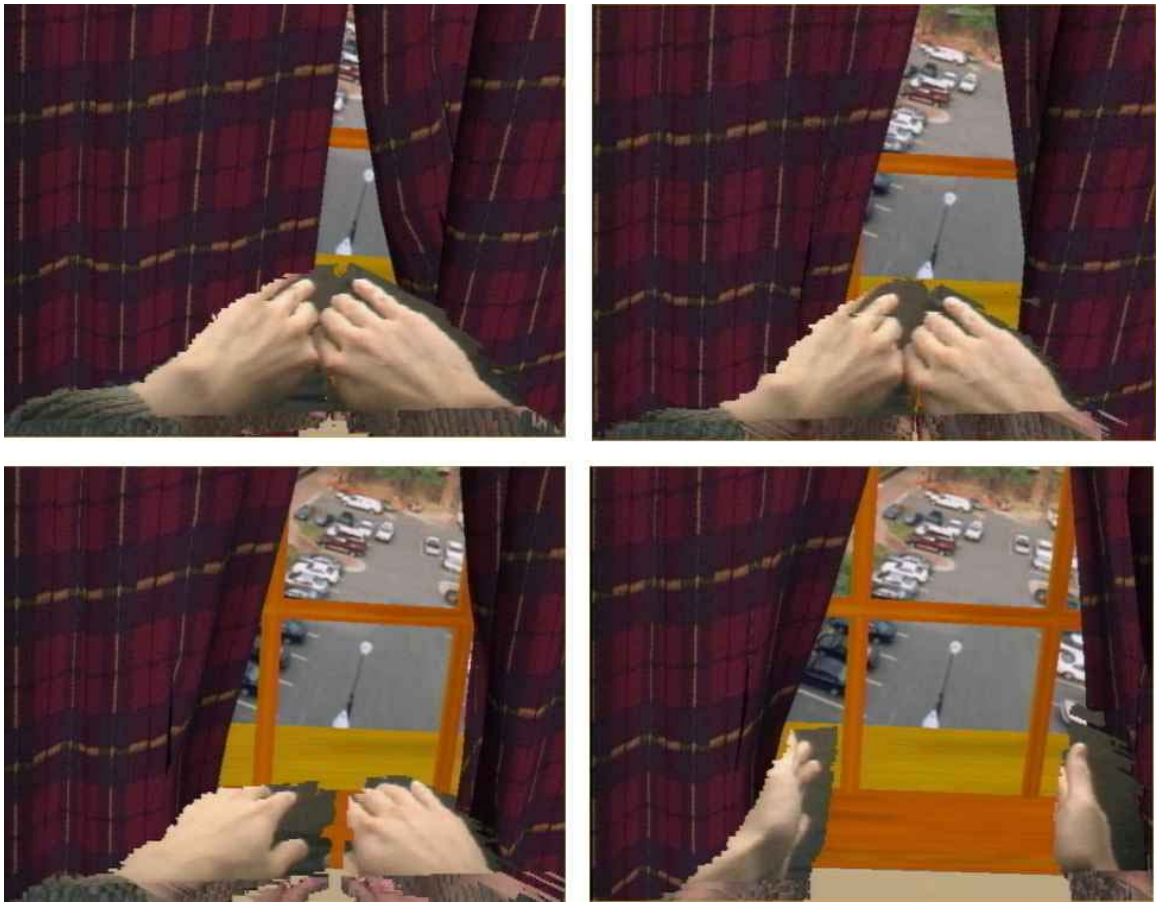
Point of first contact on the virtual object (CP_{obj})– 0.75 cm error

Point of first contact on the visual hull given the collision points (CP_{hull}) – 0.0125 cm error

Distance along recovery vector to move virtual object along – 0.0125 cm error

4.5 Algorithm Extensions

Figure 17 – Sequence of images taken from a VE where the user can interact with the curtains to look out the window



Collision Detection Extensions. Figure 17 is a sequence of frames of a user pushing aside a curtain with his hands. The collision response in this example shows the use of our algorithm with a deformable virtual object, the cloth. It further shows how the algorithm considers constraints in determining the direction of the V_{rec}. The cloth is simulated by a system of nodes in a mesh. To apply our algorithm to a deformable object, we consider each triangle independently, and individually detect collisions with real objects. For each triangle in collision, the calculated recovery vector and distance is passed to the cloth simulation as displacement vectors for the cloth simulation nodes. In the case of the curtains, we would like to constrain their motion to translation in the horizontal direction. So instead of computing a V_{rec} using the center of the object, we define a direction of motion and pass it to the algorithm. Now, when the objects move to get out of collision, the motion is primarily in the direction defined by the constraint vector.

Volume-querying can be done with primitives other than those of a surface boundary, such as distance fields. This proximity information can be visualized as thermal radiation of real objects onto virtual objects, magnetic fields of real objects, or barriers in a motion planning simulation.

The depth buffer from novel-viewpoint reconstruction can be converted into a polygonal mesh. We have incorporated these surfaces as collision objects in a particle system. As each reconstruction was completed, an updated surface was passed as a buffer to the particle system. Figure 18 shows a water particle system interacting with the user carrying a real plate. The user and the plate were reconstructed from a viewpoint above the table, and the resulting depth buffer was passed to the water particle system.

Figure 18 – The real-object avatars of the plate and user are passed to the particle system as a collision surface. The hand and plate cast shadows in the VE and can interact with the water particles.

5. User Study

5.1 Purpose

Motivation. The purpose of this study was to identify the effects of interaction methodologies and avatar visual fidelity on task performance and sense-of-presence during a cognitive manual task. We performed the study for two reasons. First, we are interested in what factors make virtual environments effective. Second, we wished to evaluate a new system that enables natural interactions and visually faithful avatars.

The real-time object reconstruction system allows us to evaluate the effects of interacting with real objects and having visually faithful avatars on task performance and presence. Previously, these topics would have been difficult to study due to complexity of traditional modeling and tracking techniques.

First, our system lets us investigate how performance on cognitive tasks, i.e. time to complete, is affected by interacting with real versus virtual objects. The results will be useful for training and assembly verification applications, as those applications often require the user to solve problems while interacting with tools and parts.

Second, our system lets us investigate whether having a visually faithful avatar, as opposed to a generic avatar, increases sense-of-presence. The results will provide insight into the need to invest the additional effort to render a high fidelity visual avatar. This will be useful for designers of immersive virtual environments, such as phobia treatment and entertainment VEs, that aim for high levels of participant sense-of-presence.

Background. The Effective Virtual Environments (EVE) research group at the University of North Carolina at Chapel Hill conducts basic research on what makes a virtual environment (VE) effective. This

work is a part of a larger effort to identify components crucial to effective virtual environments and builds upon the results of the study of the effect of passive haptics on presence and learning in virtual environments [Insko01]. Previous work by the EVE group includes evaluating physiological measures for sense-of-presence, the effects of static haptics, locomotion, and display field of view on presence, learning, and task performance in virtual environments [Meehan01, Usoh99, Razzaque01, Arthur00]. Task performance, sense-of-presence, learning, behavioral measures, and physiological measures are common metrics used to evaluate the effectiveness of VEs.

The Virtual Environments and Computer Graphics research group at the University College London, led by Mel Slater, has conducted numerous user studies. Their results show that the presence of avatars increases self-reported user sense-of-presence [Slater93]. They further hypothesize that having visually faithful avatars rather than generic avatars would increase presence. In their experiences, Heeter and Welch comment that having an avatar improved their immersion in the VE. They then hypothesize that a visually faithful avatar would provide an improvement [Heeter92, Welch96r].

We are interested in determining whether performance and sense-of-presence in VEs with cognitive tasks would significantly benefit from the participants' interacting with real objects rather than virtual objects.

VEs can provide a useful training, simulation, and experimentation tool for expensive or dangerous tasks. For example, in design evaluation tasks, users can examine, modify, and evaluate multiple virtual designs with less cost and time in VEs than by building real mock-ups. For these tasks, VEs contain virtual objects that approximate real objects. Would the ability to interact with real objects have a sufficiently large training effectiveness-to-cost ratio to justify deployment?

5.2 Task

Design Decisions. In devising the task, we sought to abstract tasks common to VE design applications, so to make our conclusions applicable to a wide range of VEs. Through surveying production VEs [Brooks99], we noted that a substantial number involve participants doing spatial cognitive manual tasks.

We use the following definition for spatial tasks:

"The three major dimensions of spatial ability that are commonly addressed are spatial orientation – mentally move or transform stimuli, spatial visualization – manipulation of an object using oneself as reference, and spatial relations – manipulating relationships within an object" [Satalich95]. Training and design review tasks executed in VEs typically have spatial components that involve solving problems in three dimensions.

"Cognition is a term used to describe the psychological processes involved in the acquisition, organisation and use of knowledge – emphasising the rational rather than the emotional characteristics" [Hollnagel02]. The VE applications we aim to study typically contain a significant cognitive component. For example, layout applications have users evaluating different configurations and designs. Tasks that involve spatial and cognitive skills more than motor skills or emotional decisions may be found in some commonly used intelligence tests.

We specifically wanted to use a task that involves cognition and manipulation, while avoiding tasks that primarily focus on participant dexterity or reaction speed for the following reasons:

- Participant dexterity variability would have been difficult to pre-screen or control. There was also the potential for dexterity, instead of interaction, to dominate the measures. The selected task should involve a simple and easily understood physical motion to achieve a cognitive result.
- Assembly design and training tasks done in VEs do not have a significant dexterity or reaction-speed component. Indeed, the large majority of immersive virtual environments avoid such perceptual motor-based tasks.
- VE technical limitations on interactions would limit many reaction-speed-based tasks. For example, a juggling simulator would be difficult to develop, test, and interact with, using current technology.
- Factors such as tracking error, display resolution and variance in human dexterity, could dominate results. Identifying all the significant interaction and confounding factors would be difficult.

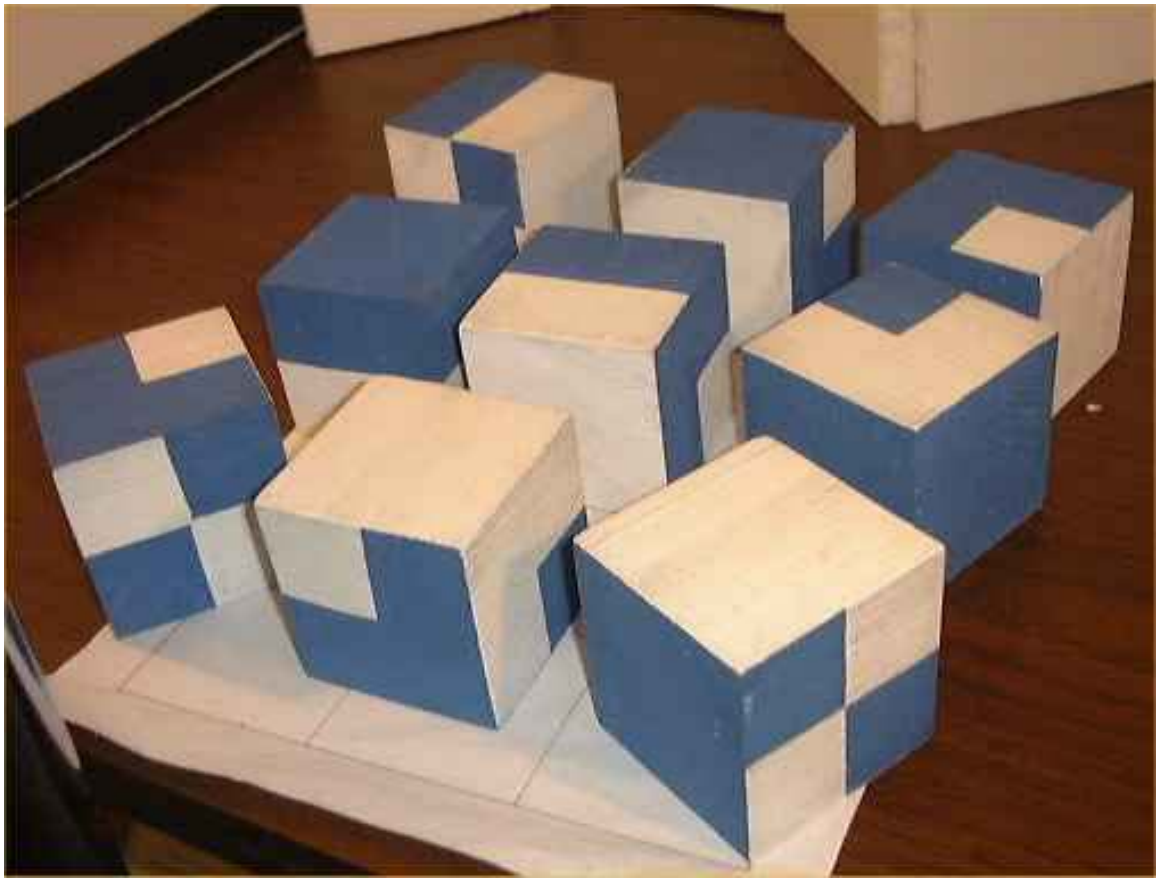
The task we designed is similar to, and based on, the block design portion of the Wechsler Adult Intelligence Scale (WAIS). Developed in 1939, the Wechsler Adult Intelligence Scale is a test widely used to measure intellectual quotient, IQ [Wechsler39]. The WAIS is composed of two major components, verbal and performance, each with subsections such as comprehension, arithmetic, and picture arrangement. The block-design component measures reasoning, problem solving, and spatial visualization, and is a part of the performance subsection.

In the standard WAIS block design task, participants manipulate small one-inch plastic or wooden cubes to match target patterns. Each cube has two faces with all white, two all red, and two half-white half-red divided diagonally. The target patterns are four or nine block patterns. Borders may or may not be drawn around the target pattern. The presence of borders affects the difficulty level of the patterns. The WAIS test measures whether a participant correctly replicates the pattern, and awards bonus points for speed. There are different time limits for the different target patterns based on difficulty and size.

There were two reasons we could not directly use the block design subtest of the WAIS. First, the WAIS test and patterns are copyrighted. Unlike the WAIS test, we administered a random ordering of patterns of relatively equal difficulty (as determined by pilot testing), rather than a series of patterns with a gradually increasing level of difficulty.

Second, the small one-inch cubes of the WAIS would be difficult to manipulate with purely virtual approaches. The conditions that used the reconstruction system would be hampered by the small block size because of reconstruction error. We therefore increased the block size to a 3” cube.

Figure 19 – Image of the wooden blocks manipulated by the participant to match a target pattern.



Task Description. Participants manipulated a number of 3"wooden blocks to make the top face of the blocks match a target pattern. Each cube had its faces painted with the six patterns shown in Figure 19. The faces represented the possible quadrant-divided white-blue patterns. The nine wooden cubes were identical.

There were two sizes of target patterns, *small* four-block patterns in a two-by-two arrangement, and *large* nine-block patterns in a three-by-three arrangement. Appendix A.9 shows the patterns used in the experiment.

We had two dependent variables. For task performance we measured the time (in seconds) for a participant to arrange the blocks to match the target pattern exactly. The dependent variable was the difference in a participant's task performance between a baseline condition (real world) and a VE condition. For sense-of-

presence, the dependent variable was the sense-of-presence scores from a presence questionnaire administered after the experience.

Design. The user study was a between-subjects design. Each participant performed the task in a real space environment (RSE), and then in one of three virtual environment conditions. The independent variables were the interaction modality (real or virtual blocks) and the avatar fidelity (generic or visually faithful). The three virtual environments had:

- Virtual objects with a generic avatar (purely virtual environment - PVE)
- Real objects with a generic avatar (hybrid environment - HE)
- Real objects with a visually faithful avatar (visually-faithful hybrid environment – VFHE)

The task was accessible to all participants, and the target patterns were intentionally made to be of a medium difficulty. Our goal was to use target patterns that were not so cognitively easy as to be manual dexterity tests, nor so difficult that participant spatial visualization ability dominated the interaction.

Pilot Study. In April 2001, Carolyn Kanagy and I conducted a pilot test as part of the UNC-Chapel Hill PSYC130 Experiment Design course. The purpose of the pilot study was to assess the experiment design and experiment conditions for testing the effect of interaction modality and avatar fidelity on task performance and presence. The subjects were twenty PSYC10 students, fourteen males and six females. The participants ranged from 18 - 21 years old and represented a wide variety of college majors.

Each participant took a test on spatial ability and did the block manipulation task on four test patterns (two small, and two large patterns) in a real environment (RSE) and then again in either a purely virtual (PVE) or visually faithful hybrid environment (VFHE). These experimental conditions is described more fully in Section 5.3. We present here the pilot test results.

For each participant, we examined the *difference* in task performance between the real and purely virtual or between the real and visually faithful hybrid environments. Thus we were looking at the impedance the

virtual environment imposed on task performance. Table 1 shows the average time difference between the VE performance (purely virtual or visually-faithful hybrid) and the real space performance.

Table 1 – (Pilot Study) Difference in Time between VE performance and Real Space performance

	Average Time Difference Small Patterns	Average Time Difference Large Pattern
Purely Virtual Environment – Real Space Environment	23.63 seconds	100.05 seconds
Visually-Faithful Hybrid Environment – Real Space Environment	8.95 seconds	40.08 seconds

We performed a two-tailed t-test and found a significant difference in the impedance of virtual task performance, compared to the real space task performance, between the two conditions [$\underline{t} = 2.19$, $df = 18$, $\underline{p}<0.05$ (small patterns), $\underline{t} = 3.68$, $df = 18$, $\underline{p}<0.05$ (large patterns)]. Therefore we concluded that manipulating purely virtual objects within a VE created a substantially greater degradation in performance than did manipulating real objects within a VE.

We administered questionnaires for sense-of-presence during the VE experience. They were not found to be significantly different between the two conditions. We were surprised that the visually faithful avatars did not result in an increased sense-of-presence, as the visual fidelity and kinematic fidelity of the avatars were substantially higher in the VFHE condition than in the PVE condition.

We also administered questionnaires on simulator sickness. There was not a significant difference between the two conditions' effects on simulator sickness.

Although the task performance in the visually faithful hybrid environment was significantly closer to the real space environment than was that in the purely virtual environment, we did not know whether the performance improvement was because of realistic avatars, interacting with real objects, or a combination of both. This led us to develop a third condition for the final user study that had the user manipulate real objects but with a generic avatar.

In the pilot study, the difference in task performance mean appears to be very strong, but because the variance in task performance among subjects was high the significance level of the t test was just marginal. We therefore wished to design the final study so as to reduce this unwanted variability. Fortunately, the spatial ability test suggested a way to do this.

Participants in the pilot study also took the Guilford-Zimmerman Aptitude Survey, Part 5: Spatial Orientation. Spatial ability was the strongest covariate ($r = -0.41$) with task times. This result suggested that if we controlled for spatial ability variability in participants, we would get stronger, more useful results. Further supporting this was the fact that those with poor spatial aptitude scores were more likely to ***be unable to complete*** the target patterns at all. We were unable to use the data from those participants who could not complete all the test patterns.

Our pilot study experiences led us to modify the experiment design in the following ways:

- We included an additional condition to separate avatar fidelity effects from interaction modality effects.
- We modified the real world task to operate in an enclosed space so as to match more closely training and design tasks that require the user to manipulate objects not in a direct line of sight.
- Upon consulting with Professor Mel Slater of University College London and reviewing literature [Slater99], we changed the presence questionnaire from the Witmer-Singer Presence Questionnaire to the Steed-Usoh-Slater Presence Questionnaire [Usoh00].
- We controlled for people with high spatial aptitude by requiring participants to have taken a calculus course. Cognitive psychology professor Edward Johnson advised us that spatial aptitude and enrollment in higher-level mathematics courses correlate strongly.
- We learned that participants were wary that their performance was being compared to others' performances. Prior to the final study, we explained to participants that we were comparing each participant's VR performance to his real space performance and not to other participants' performances.

And specifically for variance reduction:

- We randomized the assignment of the target patterns to conditions to help prevent the inter-pattern difficulty variability from skewing our results. In the pilot study, the sets of target patterns assigned to the real and to the virtual condition were identical for each participant. This had the undesired biasing effect of having a particularly difficult or easy pattern always affecting the same condition. Although we performed pilot testing to try to select equally difficult patterns, the relative difficulty was not known.
- We lengthened the task training. We noted that the task had a definite learning effect that we were not interested in measuring. A longer practice session in the real space helped reduce the performance variances for all conditions below those seen in the pilot study.

The pilot study provided us invaluable experience in understanding the factors that contributed to task performance. This resulted in much stronger results in the final user study.

The pilot study also made us aware of the biasing effect of unevenly balanced patterns. Thus before the main study, we performed testing to select patterns of relatively equal difficulty. Five volunteers were timed on a suite of large and small patterns. Any patterns that were consistently different than a person's average time were discarded. We had on hand extra patterns of equal difficulty to the test and practice suite of patterns in case of some anomaly.

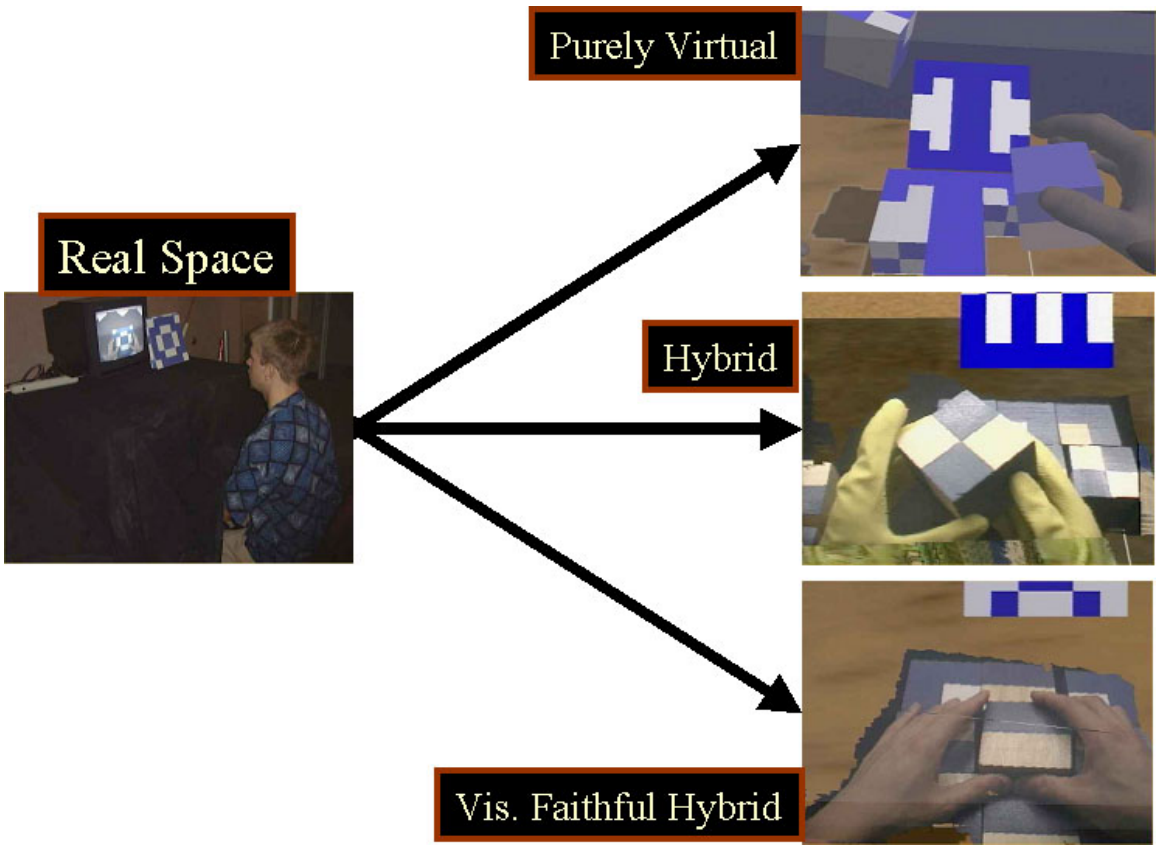
From the set of patterns, a random ordering was generated for each participant. Each pattern would appear as either a practice or test pattern in either the real space or VE condition. All participants saw the same twenty patterns (ten practice, ten test), just in different orders.

5.3 Final Study Experiment Conditions

Each participant performed the block pattern-matching task in an enclosed real space environment (RSE) without any VE equipment. Next, they performed the same task in one of three virtual environments:

purely virtual (PVE), hybrid (HE) , or visually-faithful hybrid (VFHE). Figure 20 shows the different environments where participants performed the block-manipulation task.

Figure 20 – Each participant performed the task in the RSE and then in one of the three VEs.



The participants were randomly assigned to one of the three groups, 1) RSE then PVE, 2) RSE then HE, or 3) RSE then VFHE.

Real Space Environment (RSE). In the real space environment (RSE), the participant sat at a desk as shown in Figure 21. On the desk were the nine wooden blocks inside a rectangular 36" x 25" x 18" enclosure. The side facing the participant was open and the whole enclosure was draped with a dark cloth. Two small lights lit the inside of the enclosure.

Figure 21 – Real Space Environment (RSE) setup. The user watches a small TV and manipulates wooden blocks to match the target pattern.

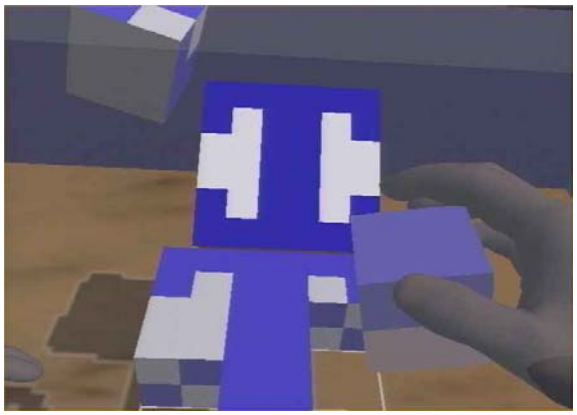
A 13" television placed atop the enclosure displayed the video feed from a "lipstick camera" mounted inside the enclosure. The camera had a similar line of sight as the participant. The participant performed the task while watching the TV. The target pattern was placed next to the TV.

RSE Equipment: Real blocks, TV, lipstick camera, cloth-draped enclosure, small lights.

Purely Virtual Environment (PVE). In the purely virtual environment (PVE), participants stood at a four-foot high table. The table surface was centered in the reconstruction volume. As shown in Figure 22, participants wore Fakespace Pinchgloves, each tracked with Polhemus Fastrak magnetic trackers, and a Virtual Research V8 head-mounted display (HMD).

Figure 22 – Purely Virtual Environment (PVE) setup. The user wore tracked pinchgloves and manipulated virtual objects.

Figure 23 – PVE participant's view of the block manipulation task.



The gloves acted like a switch. When the participant pinched two fingers together (such as the thumb and forefinger), the gloves signaled the system to pick up a virtual object, and a grasping hand avatar was displayed. When the participant released the pinch, the gloves signaled the system to drop the virtual object in hand, and the open hand avatar was displayed.

Participants manipulated virtual blocks with a generic avatar. The block closest to an avatar's hand was highlighted. This informed the participant that if he or she were to pinch, the highlighted block would be grasped.

If the participant pinched while a virtual block was highlighted, the virtual block snapped into the virtual avatar's hand so that the hand appeared to be holding the block. To rotate the block, the participant rotated his hand while maintaining the pinching gesture.

If the participant released the block within six inches of the workspace surface, the block snapped (with both translation and rotation) into an unoccupied position in a three by three grid. This snapping removed the need for fine-grained interaction that might have artificially inflated the time to complete the task. If the block was released a few inches away from the grid, it simply dropped onto the table. If the block was released more than six inches above the table, the block floated in mid-air. This floating facilitated rotating

virtual blocks faster. There was no inter-block collision detection, and block interpenetration was not automatically resolved. Participants typically resolved any block interpenetration that occurred.

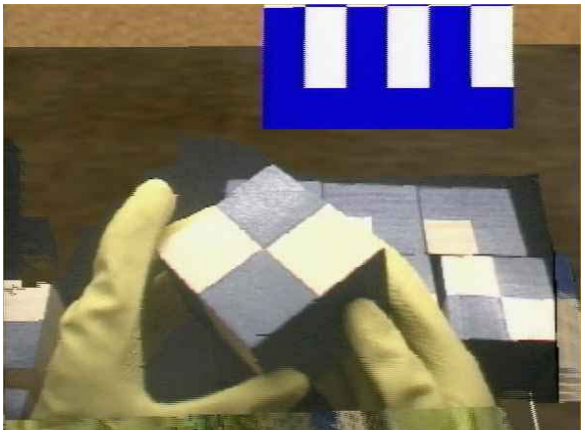
The target pattern was displayed as a card within the environment. The PVE program ran at a minimum of twenty frames per second. Figure 23 shows a screenshot of the images the participant saw.

PVE Equipment: Fakespace Pinchgloves, Polhemus Fastrak trackers, Virtual Research V8 HMD. All the virtual environments conditions were rendered with the SGI Reality Monster graphics supercomputer housed in Sitterson Hall at the University of North Carolina at Chapel Hill. The PVE ran on one rendering pipe with four raster managers.

Hybrid Environment (HE). In the hybrid environment (HE), the participants stood at the same table as in the PVE. They wore the same V8 HMD, and yellow dishwashing gloves as shown in Figure 24. The participant did not wear any hand trackers, as the reconstruction system generated real-time virtual representations of the user and the blocks. The physical and virtual setups were similar to the PVE.

Figure 24 – Hybrid Environment (HE) setup. Participant manipulated real objects while wearing dishwashing gloves to provide a generic avatar.

Figure 25 – HE participant's view of the block manipulation task.



Real blocks, identical to those in the RSE, were manipulated as both the participant and the blocks were incorporated into the VE by using the real-object reconstruction system. The HMD displayed a reconstruction of the participant within the VE, texture mapped with images from a HMD mounted camera.

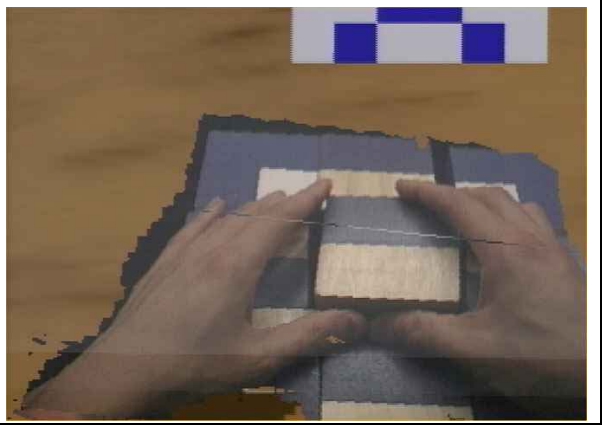
All participants saw an avatar with generic appearance and accurate shape because they were wearing identical dishwashing gloves. The HE and VFHE ran at a minimum of twenty frames per second, and the reconstruction algorithm ran at a minimum of twelve frames per second. Figure 25 shows a screenshot of the images the participant saw.

HE Equipment: Real blocks, HMD with mounted lipstick camera (with a similar line-of-sight as the user), three wall-mounted cameras for reconstruction system, dishwashing gloves.

Visually-Faithful Hybrid Environment (VFHE). The visually-faithful-hybrid environment (VFHE), as shown in Figure 26, was similar to the HE except the participants did not wear gloves.

Figure 26 – Visually Faithful Hybrid Environment (VFHE) setup. Participants manipulated real objects and were presented with a visually faithful avatar.

Figure 27 – VFHE participant's view of the block manipulation task.



The avatar representation was visually faithful to the person, as the shape reconstruction was texture-mapped with images from a HMD mounted camera. The participant saw an image of his own hands, warts and all. Figure 27 shows a screenshot of the images the participant saw.

VFHE Equipment: Real blocks, HMD with mounted lipstick camera, three wall-mounted cameras for reconstruction system.

Virtual Environment. In all three of the virtual conditions (PVE, HE, VFHE), the VE was composed of a table inside a room with a corresponding real Styrofoam table that is registered to the same space. The room was a relatively generic room with a radiosity-as-textures global illumination solution as shown in Figure 28. The room was populated with several virtual objects, including a lamp, a plant, and a painting of the Mona Lisa. The enclosure in the RSE was also rendered in the VE, but instead of being obscured by cloth, it was rendered with transparency to allow the participants to see into it.

Figure 28 – Virtual environment for all three (PVE, HE, VFHE) conditions.

Rationale for Conditions. We expected that a participant's performance in a real environment, with real blocks and no VE equipment, would produce the best results, as the interaction and visually fidelity were both optimal. Thus we compared how close the task performance in the three virtual environments was to the RSE task performance. We compared the reported sense-of-presence in the three virtual environments to each other.

We used the RSE for task training to reduce variability in individual task performance and as a baseline. This is because the block design task had a definite learning curve. The more patterns participants did, the better at doing them they became. We found that most participants required two to three practice patterns of each size to get performance to a stable level. Doing the task in the RSE allowed participants to practice there, as opposed to spending additional time in the VE. We wanted to limit time in the VE to less than fifteen minutes, since many pilot subjects began complaining of fatigue after that amount of time in the VE.

People who were good or poor at the task in the RSE likely carried that aptitude into the VE condition. To reduce variability in the task performance measures, we examined the relative difference between a participant's performance in the RSE and his performance in the VE condition, rather than comparing absolute performances between the VE conditions.

The PVE represented the way things are usually done with current technology. All the objects were virtual, and interactions were accomplished with specialized equipment and gestures. The difference in task performance between the RSE and the PVE corresponded to the additional impedance of interacting with virtual objects.

The HE was used to separate the effects on task performance afforded by interacting with real objects and by being provided a visually-faithful visual representation. Participants interacted with real objects but saw a generic avatar through wearing visually-similar dish-washing gloves.

The VFHE allowed us to evaluate the cumulative effect of both natural interactions and high visual fidelity on performance and presence. We were interested in seeing how close participants' performance in our reconstruction system would be to their ideal RSE performance.

5.4 Measures

Audio, video, and written notes were recorded for all participants. Anonymous IDs were used throughout the experiment and data analysis.

Task Performance Measures. For task performance, we measured the time each participant took to replicate *correctly* the target pattern. We also recorded if the participant incorrectly concluded that they had correctly replicated the target pattern. In such cases, the experimenter informed the subject of the error and the subject continued to work on the problem. Each participant eventually completed every pattern correctly.

Sense-of-presence Measures. For sense-of-presence, participants answered the Steed-Usoh-Slater Presence Questionnaire (SUS) [Usoh00] after completing the task in the VE condition.

Other Factors. To observe the correlation of spatial ability with task performance, we administered the Guilford-Zimmerman Aptitude Survey, Part 5: Spatial Orientation before the experience. To assess the level of discomfort or simulator sickness, the Kennedy – Lane Simulator Sickness Questionnaire was given before and after the experience.

Participant Reactions. At the end of the session, we interviewed each participant on their impressions of their experience. Finally, we recorded self-reported and experimenter-reported behaviors.

Subject Information. Forty participants completed the study, thirteen each in the purely virtual environment (PVE) and hybrid environment (HE) groups, and fourteen in the visually-faithful hybrid environment (VFHE) group. There were two participants who could not complete the experiment due to equipment problems, one due to errors in data recording, and one due to nausea in the PVE. We did not use their data in computing results.

The participants were primarily male (thirty-three males and eight females) undergraduate students enrolled at UNC (thirty-one undergraduates, three masters students, three PhD students, one staff member, and two spouses of graduate students). Participants were recruited through short presentations in UNC undergraduate Computer Science courses and word of mouth.

The participants had little (fewer than two prior sessions) immersive virtual reality experience. They reported their prior VR experience as Mean = 1.37 (standard deviation = 0.66, min = 1, max = 4) on a scale from 1 (Never before) to 7 (A great deal).

Most participants reported they use a computer a great deal. They reported their computer usage as M = 6.39 (s.d. = 1.14, min = 3, max = 7) on a scale from 1 (Not at all) to 7 (Very much so).

Participants were asked, during the past three years, what was the most they played computer and video games. Most reported between one to five hours a week, M=2.85 (s.d. = 1.26, min = 1, max = 5) on the following scale (1. Never, 2. Less than 1 hour per week, 3. Between 1 and 5 hours per week, 4. Between 5 and 10 hours per week, 5. More than 10 hours per week).

There were no significant differences between the groups in previous VR experience, computer usage, or video game play.

During the recruiting process, we listed the following restricting factors:

- Participants must be able to walk without assistance and have use of both hands.
- Participants must have 20/20 vision in both eyes or as corrected.
- Participants cannot have a history of epilepsy, seizures, or strong susceptibility to motion sickness.
- Participants must be able to communicate comfortably in spoken and written English.
- Participants cannot have significant previous experience (more than two sessions) with virtual reality systems.
- Participants must have taken or be currently enrolled in a higher-level mathematics course (Math31, Calculus of Functions of One Variable, or equivalent).

A second set of criteria was verified at the beginning of the session:

- Participants must be in their usual states of good fitness at the time of the experiment (i.e., participants are excluded if they have used sedatives, tranquilizers, decongestants, anti-histamines, alcohol, or other significant medication within 24 hours of the session).
- Participants must be comfortable with the HMD display, and must easily fit the HMD on their heads.

5.5 Experiment Procedure

The study was conducted over three days. For convenience with respect to setup and equipment, each day's participants were assigned to a specific study condition. The participants did not have prior knowledge about the experiment of its condition, and thus the overall effect was a random assignment of participants to conditions.

Each participant went through a one-hour session that involved three stages:

- **Pre-experience** – forms and questionnaires were filled out
- **The experience** –
    - First a block design task in the RSE
    - Then in one of PVE, HE, or VFHE
- **Post-experience** – debriefing, and more questionnaires were filled out

Upon arriving, all participants read and signed a consent form (Appendix A.1). All participants then went through a final screening where they completed a questionnaire designed to gauge their physical and mental condition (Appendix A.2). This was to establish the participants were not physically compromised in a way that might affect their task performance. Three participants in the HE condition reported having had more than three alcoholic drinks in the past 24 hours. We examined their task performance measures and noted that not considering their data in our statistical analysis did not change the statistical significance of the overall results. Thus, we did not throw out their data.

Next, the Kennedy-Lane Simulator Sickness Questionnaire (Appendix A.3) was given. The same questionnaire was administered after the experience to assess the effects of the VE system on the participants' physical state.

Finally, the participants were given a spatial ability test, The Guilford-Zimmerman Aptitude Survey Part 5: Spatial Orientation (Appendix A.4). The participant read the instructions and did practice problems for a five-minute period, then answered as many multiple choice questions as possible in a ten-minute period. This test enabled us to correlate spatial ability with the task performance measures.

Real Space. After completing the pre-experience stage, the participant entered the room with the real space environment (RSE) setup. The participant was presented with the set of nine painted wooden blocks. The participant was told that the blocks are all identical. The video camera was shown, the cloth lowered, and the TV turned on. The participant was told that they would be manipulating the blocks while viewing the blocks and their hands by watching the TV. The participant was instructed to manipulate the blocks until the pattern showing on the top face of the blocks duplicated a series of target patterns. The participant was also told that we would record the time it took them to correctly complete each pattern. The participant was instructed to examine the blocks and become comfortable with moving them.

When the participant understood the procedure, they were given a series of six practice patterns, three small (2 x 2) and then three large (3 x 3) patterns. The participant was told how many blocks are involved in the pattern and to notify the experimenter when he thought he had correctly reproduced the pattern. When the practice patterns were completed, the first test pattern was presented. Recall that the order of the patterns that each participant sees is unique, though all participants see the same twenty patterns (six real space practice, six real space timed, four VE practice, four VE timed).

We recorded the time required to complete each test pattern correctly. If the participant misjudged the completion of the pattern, we noted this as an error and told the participant to attempt to fix the errors, without stopping the clock. The final time was used as the task performance measure for that pattern.

Between patterns, the participant was asked to randomize the blocks' positions and orientations. The task continued until the participant had completed all six timed test patterns, three small and three large.

Virtual Space. The experimenter helped the participant put on the HMD and any additional equipment particular to the particular VE condition (PVE – tracked pinch gloves, HE – dishwashing gloves). Following a period of adaptation to the VE, the participant practiced performing the task on two small and two large patterns. The participant then was timed on two small and two large test patterns. Participants were told they could ask questions and take breaks between patterns if they desired. Only one person (a PVE participant) asked for a break.

Post Experience. After completing the task, the participants were interviewed about their impressions of and reactions to the session. The debriefing session was a semi-structured interview; the specific questions asked (attached as Appendix A.6) were only starting points, and the interviewer could delve more deeply into responses for further clarification or to explore unexpected conversation paths. In the analysis of the post-experience interviews, we used axial coding to identify trends and correlate responses to shed light on the participants' subjective evaluation of their experiences. When reviewing the trends, note that not every participant had a response to every question that could be categorized. In fact, most participants spent much of the interview explaining to us how they felt the environment could be improved, regardless of the question.

Next, participants filled out the simulator sickness questionnaire again. By comparing their pre- and post-experience scores, we could assess if their level of simulator sickness had changed while performing the task (Appendix A.7). Finally, a modified Slater – Usoh – Steed Virtual Presence Questionnaire (Appendix A.8) was given to measure the participants' level of presence in the VE.

Managing Anomalies. If the head tracker lost tracking or crashed, we quickly restarted the system (estimated to be about 5 seconds). In almost all the cases, the participants were so engrossed with the task they never noticed any problems and continued working on the task. We noted long tracking failures, and

participants who were tall (which gave our aging HiBall tracker problems) were allowed to sit to perform the task. None of the tracking failures appeared to affect the task performance time significantly.

On hand was a set of additional patterns for replacement of voided trials, such as if a participant dropped a block onto the floor. This was used twice, and the substitutions were noted.

Statistical Analysis. The independent variables are the different VE conditions (purely virtual, hybrid, and visually-faithful hybrid). The dependent variable for task performance was the difference in the time to correctly replicate the target pattern in the VE condition compared to the RSE. The dependent variable for sense-of-presence was the sense-of-presence score on the Steed-Usoh-Slater Presence Questionnaire.

We use a two-tailed t-test to determine if the disparity in the observed values between groups is due to chance or to an actual difference between the conditions. The T-test and the related *p*-value describe this likelihood. It is common to accept results as a significant difference in the observed factor if the observed *p*-values are less than a 0.05 level. This level, called the α value, represents the chance that we are making a Type 1 error (labeling a result as significant even though the true state is to the contrary). We use an α=0.05 level for significance unless otherwise stated. At this level there is a 95% probability that the observed difference between the means was due to an actual difference of the factor in the conditions rather than to chance.

5.6 Hypotheses

Task Performance: Participants who manipulate real objects will complete a spatial cognitive manual task in less time than will participants who manipulate corresponding virtual objects.

Sense-of-Presence: Participants represented in the VE by a visually faithful self-avatar will report a higher sense-of-presence than will participants represented by a generic self-avatar.

Associating Conditions with Hypotheses.

- Our first hypothesis was that the difference in task performance between both the hybrid environment (HE) and visually-faithful hybrid environment (VFHE), and the real space environment (RSE), would be smaller than the difference in performance between the purely virtual environment (PVE) and RSE, i.e. interacting with real objects improves task performance.
- Our second hypothesis was that self-reported sense-of-presence in the VFHE would be higher than in either the PVE or HE, i.e. avatar visual fidelity increases sense-of-presence.
- Further, we expected no significant difference in task performance for participants in the VFHE and HE conditions, i.e. interacting with real objects improves task performance regardless of avatar visual fidelity.
- Finally, we expected no significant difference in sense-of-presence for participants in the HE and PVE conditions, i.e. generic hand avatars would have similar effects on presence regardless of the presence of real objects.

5.7 Results

Task Performance. The complete task performance results are in Appendix B.1. Tables 2-10 and Figures 28-30 summarize them.

Table 2 – Task Performance Results

	Small Pattern Time (seconds)				Large Pattern Time (seconds)			
	Mean	S.D.	Min	Max	Mean	S.D.	Min	Max
Real Space (n=40)	16.81	6.34	8.77	47.37	37.24	8.99	23.90	57.20
Purely Virtual (n=13)	47.24	10.43	33.85	73.55	116.99	32.25	70.20	192.20
Hybrid (n=13)	31.68	5.65	20.20	39.25	86.83	26.80	56.65	153.85
Vis Faith Hybrid (n=14)	28.88	7.64	20.20	46.00	72.31	16.41	51.60	104.50

Table 3 – Difference in Task Performance between VE condition and RSE

	Small Pattern Time (seconds)		Large Pattern Time (seconds)	
	Mean	S.D.	Mean	S.D.
Purely Virtual - Real Space	28.28	13.71	78.06	28.39
Hybrid – Real Space	15.99	6.37	52.23	24.80
Visually Faithful Hybrid – Real Space	13.14	8.09	35.20	18.03

Figure 29 – Difference between VE and RSE performance for Small Patterns. The lines represent the mean difference in time for each VE condition.

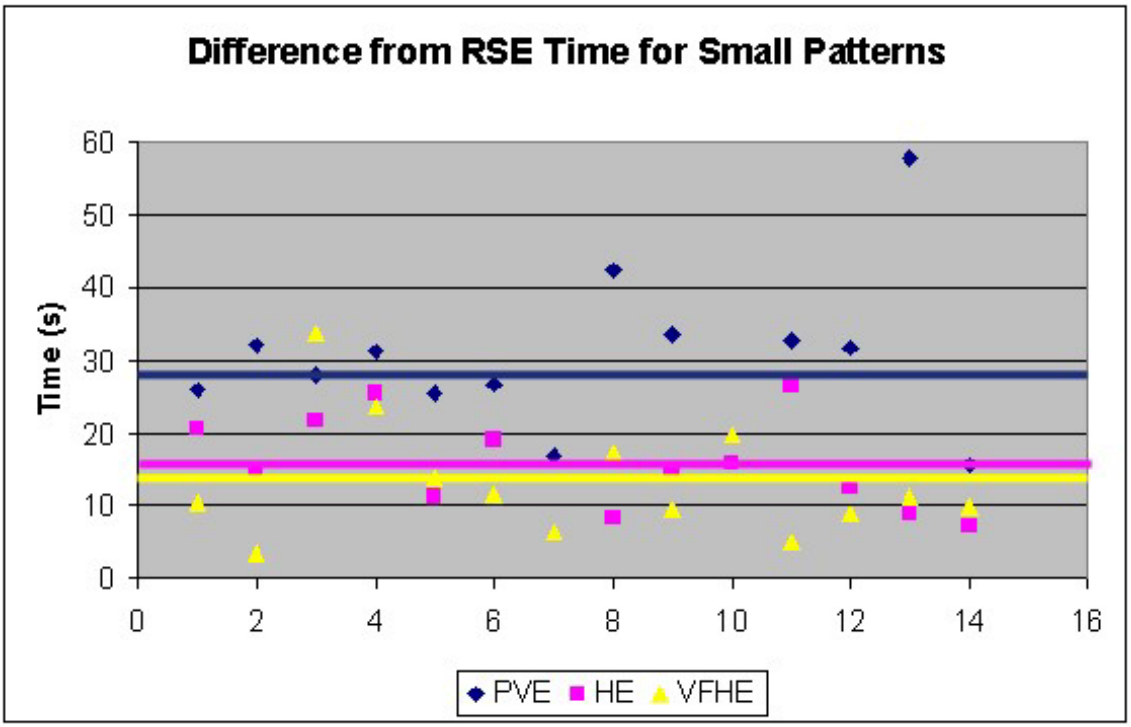


Figure 30 – Difference between VE and RSE performance for Large Patterns. The lines represent the mean difference in time for each VE condition.

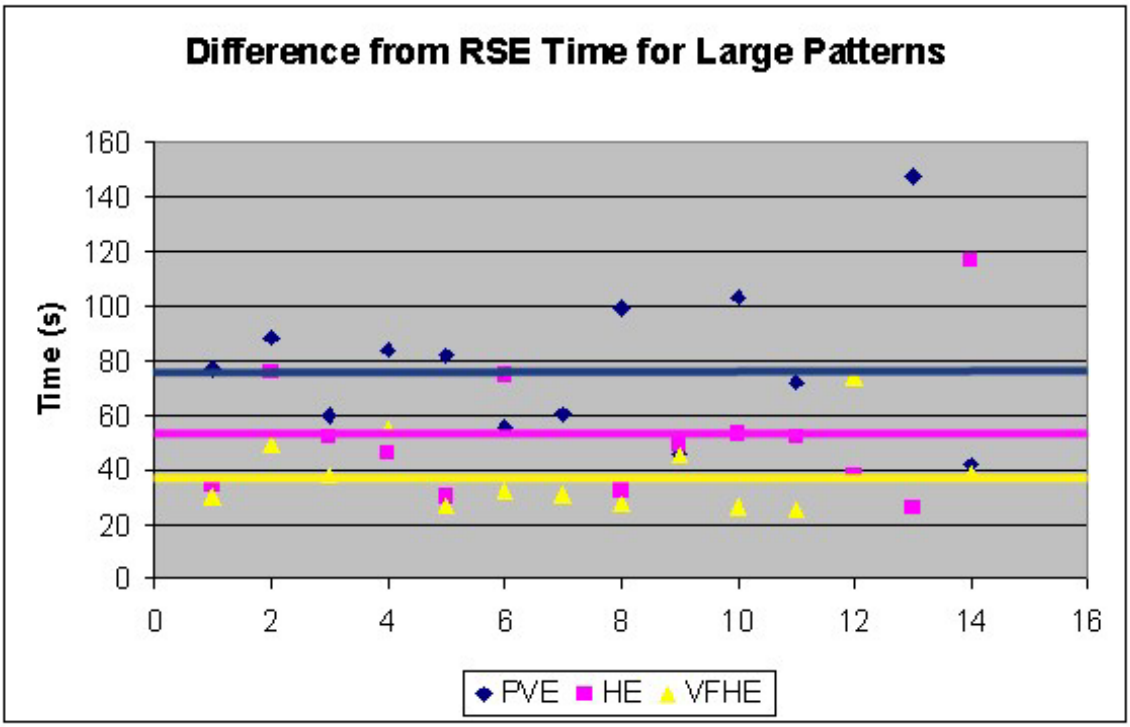


Table 4 – Between Groups Task Performance Comparison

	Small Pattern		Large Pattern	
	t – test with unequal variance	*p – value*	t – test with unequal variance	*p - value*
PVE – RSE vs. VFHE – RSE	3.32	0.0026**	4.39	0.00016***
PVE – RSE vs. HE – RSE	2.81	0.0094**	2.45	0.021*
VFHE – RSE vs. HE – RSE	1.02	0.32	2.01	0.055^{+}

* - significant at the α=0.05 level
** - significant at the α=0.01 level
*** - significant at the α=0.001 level
$^{+}$ - requires further investigation

For small patterns, both VFHE and HE task performances were significantly better than PVE task performance. For large patterns, both VFHE and HE task performances were significantly better than PVE task performance (Table 2). The difference in task performance between the HE and VFHE was not significant at the α=0.05 level (Table 4).

Performing the block-pattern task took longer in any virtual environment than it did in real space: The purely virtual environment participants took 2.84 (small patterns) and 3.23 (large patterns) **times** as long as they did in the real space (Table 3).

The task performance difference between real space performance and virtual environment performance was less for hybrid environment and visually-faithful hybrid environment participants: HE participants took 2.16 and 2.55 times as long, and the VFHE took only 1.92 and 2.04 times, as long as shown in Table 5.

Table 5 – Relative Task Performance Between VE and RSE

	Small Pattern				Large Pattern			
	Mean	S.D.	Min	Max	Mean	S.D.	Min	Max
Purely VE / RSE	2.84	0.96	0.99	4.66	3.23	0.92	2.05	5.03
Hybrid VE / RSE	2.16	0.60	1.24	3.07	2.55	0.75	1.63	4.13
Visually Faithful Hybrid VE / RSE	1.92	0.65	1.16	3.71	2.04	0.59	0.90	3.42

In the SUS Presence Questionnaire, the final question asked how well the participants thought they achieved the task, on a scale from 1 (not very well) to 7 (very well). The VFHE (5.43) and PVE (4.57) groups were significantly different ($\underline{t}_{27} = 2.23$, $p=0.0345$) at the $\alpha=0.05$ level.

Table 6 – Participants' Response to How Well They Thought They Achieved the Task

How well do you think you achieved the task? (1..7)		
	Mean	S.D.
Purely Virtual Environment	4.57	0.94
Hybrid Environment	5.00	1.47
Visually Faithful Hybrid Environment	5.43	1.09

Sense-of-presence. The complete sense-of-presence results are in Appendix B.B.

We augmented the standard Steed-Usoh-Slater Presence Questionnaire with two questions that focused on the participants' perception of their avatars. The entire questionnaire is included as Appendix A.6.

- How much did you associate with the visual representation of yourself (your avatar)? During the experience, I associated with my avatar (1. not very much, 7. very much)
- How realistic (visually, kinesthetically, interactivity) was the visual representation of yourself (your avatar)? During the experience, I thought the avatar was (1. not very realistic, 7. very realistic)

Table 7 – Steed-Usoh-Slater Sense-of-presence Scores for VEs

	Total Sense-of-presence Score Scale from 0..6			
	Mean	S.D	Min	Max
Purely VE	3.21	2.19	0	6
Hybrid VE	1.86	2.17	0	6
Visually Faithful Hybrid VE	2.36	1.94	0	6

Figure 31 – Raw Steed-Usoh-Slater Sense-of-presence Scores. The horizontal lines indicate means for the VE conditions. Note the large spread of responses.

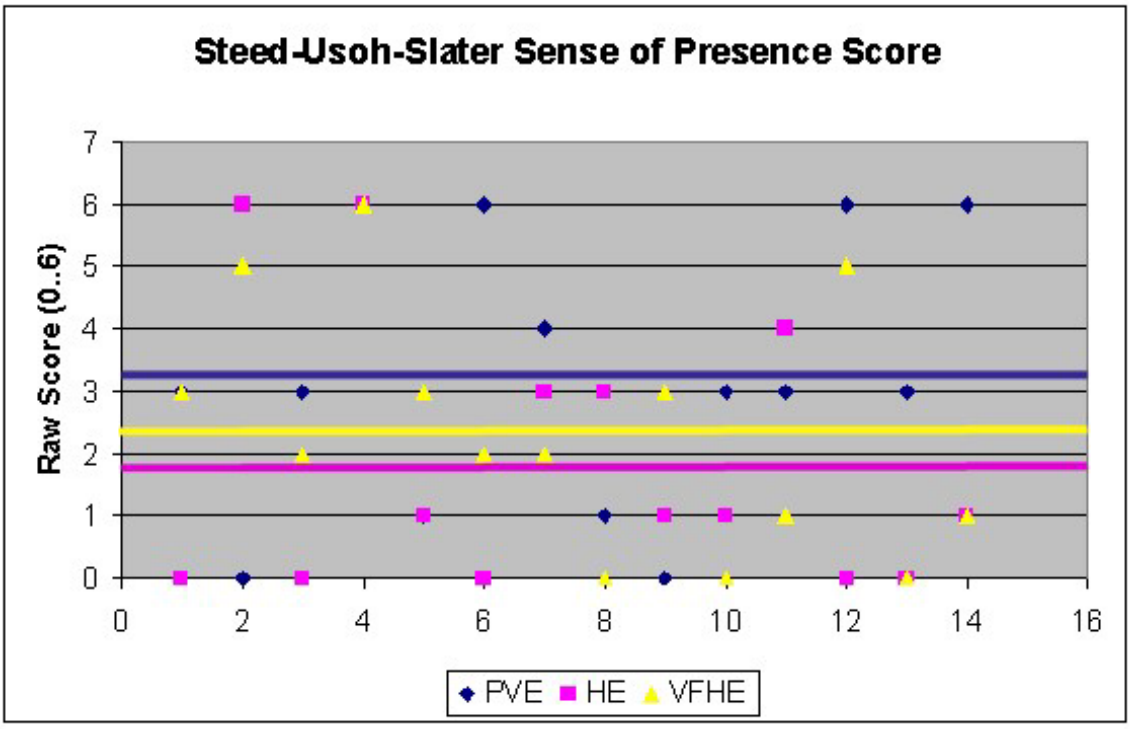


Table 8 – Steed-Usoh-Slater Avatar Questions Scores

	Association with avatar 1. Not very much… 7. Very much				Avatar realism 1. Not very realistic… 7. Very realistic			
	Mean	S.D.	Min	Max	Mean	S.D.	Min	Max
Purely VE	4.43	1.60	1	6	3.64	1.55	1	7
Hybrid VE	4.79	1.37	2	6	4.57	1.78	2	7
Visually Faithful Hybrid VE	4.64	1.65	2	7	4.50	1.74	3	7

Table 9 – Comparing Total Sense-of-presence Between Conditions

	Between Groups Total Sense-of-presence	
	t – test with unequal variance	*p – value*
PVE – VFHE	1.10	0.28
PVE – HE	1.64	0.11
VFHE – HE	0.64	0.53

Other Factors. Simulator sickness was not significantly different between the groups at the $\alpha = 0.05$ level. The complete results are included as Appendix B.5.

Spatial ability was not significantly different between groups, as shown in Table 10. The complete spatial ability test results are included as Appendix B.6. This shows that the groups were not biased by the base spatial ability skills of participants. Spatial ability was moderately correlated ($r = -0.31$ for small patterns, and $r = -0.38$ for large patterns) with performance.

Table 10 – Simulator Sickness and Spatial Ability Between Groups

	Between Groups Simulator Sickness		Between Groups Spatial Ability	
	t – test with unequal variance	*p – value*	t – test with unequal variance	*p – value*
PVE vs. VFHE	1.16	0.26	-1.58	0.13
PVE vs. HE	0.49	0.63	-1.41	0.17
VFHE vs. HE	-0.57	0.58	0.24	0.82

5.8 Discussion

Task Performance.

Task Performance Hypothesis: Participants who manipulate real objects will complete a cognitive manual task in less time than will participants who manipulate corresponding virtual objects.

For the case we investigated, **interacting with real objects provided a quite substantial performance improvement over interacting with virtual objects for cognitive manual tasks**. Although task performance in all the VE conditions was substantially worse than in the real space environment, the task performance of hybrid and visually-faithful hybrid participants was significantly better than for purely virtual environment participants.

There is a slight difference between HE and VFHE performance (Table 4, p=0.055), and we do not have a hypothesis as to the cause of this result. This is a candidate for further investigation.

These results showing significantly poorer task performance when interacting with purely virtual objects leads us to believe that the same hindrances would affect practice, training, and learning the task.

Handling real objects makes task performance and interaction in the VE more like the actual task.

Sense of Presence.

Sense-of-presence Hypothesis: Participants represented in the VE by a visually faithful self-avatar will report a higher sense-of-presence than will participants represented by a generic self-avatar.

Although interviews showed visually faithful avatars (VFHE condition) were preferred, there was no statistically significant difference in reported sense-of-presence compared to those presented a generic avatar (HE and PVE).

There were no statistically significant differences at the α=0.05 level between any of the conditions for all eight sense-of-presence questions. There were no differences when examining the individual questions or the sum total sense-of-presence score.

Based on a study, Slater cautions against the use of the SUS Questionnaire to compare presence across virtual environment conditions, but also points out that no current questionnaire seems to support such comparisons [Slater00]. Just because we did not see a presence effect with the SUS Questionnaire does not mean that there was none.

Participant Interviews. An observation from the post-experience interviews showed that many participants in the purely virtual condition note that the "avatar moved when I did" and gave a high mark to the avatar questions. Some in the visually faithful avatar condition said, "Yeah, I saw myself" and gave an equally high mark to the avatar questions. This resulted in similar scores to the questions on avatar realism.

In hindsight, the different components of the self-avatar (appearance, movement, and interactivity) should perhaps have been divided into separate questions. Regardless of condition, the participant response had a movement first, appearance second trend. From this, we hypothesize *kinematic fidelity of the avatar is more important than visual fidelity for sense-of-presence*. Developing techniques to determine the effect of visual fidelity, separate from dynamic fidelity, on sense-of-presence, could be an area of future research, but we believe this might prove to not be very fruitful as we believe the additional impact of visual fidelity is not very strong.

Debriefing Trends. We list here the major trends and discuss all trends in more detail later.

- When asked about the virtual representation of their bodies, PVE and HE participants commented on the fidelity of motion, whereas VFHE participants commented on the fidelity of appearance. This leads us to hypothesize that appearance fidelity seems to include motion fidelity.
- Participants in all groups responded that they were almost completely immersed when performing the task.

- Participants in all groups responded that they felt the virtual objects in the room (such as the painting, plant, and lamp) improved their sense-of-presence, even though they had no direct interaction with these objects.
- Participants in all groups responded that seeing an avatar improved their sense-of-presence.
- 7 out of 27 VFHE and HE participants mentioned that the tactile feedback of working with real objects improved their sense-of-presence.

The following interview trends consistent with results of previous research or our experiences with VEs:

- Being involved in a task heightened sense-of-presence.
- Interacting with real objects heightened sense-of-presence [Insko01].
- System latency decreased sense-of-presence [Meehan01].

Debriefing Results – Major Trends. A better picture of the effect of the visually faithful avatars and interacting with real objects can be drawn from the debriefing responses of the participants.

Participants presented with generic avatars, the PVE and HE conditions, remarked that the motion fidelity of the avatars contributed to their sense-of-presence. In fact, all comments on avatar realism from PVE and HE conditions related to motion accuracy.

- “Once I got used to where the hands were positioned… it felt like they were my hands.”
- “It was pretty normal, it moved the way my hand moved. Everything I did with my hands, it followed.”
- “They followed my motions exactly, I thought”
- “I thought they behaved pretty well. I didn't feel like I was looking at them, though. I felt I was using them more like a pointer, than the way I would look at my own hands.”
- "The only thing that really gave me a sense of really being in the virtual room was the fact that the hands moved when mine moved, and if I moved my hand, the room changed to represent that movement."

- "Being able to see my hands moving around helped with the sense of 'being there'."

On the other hand, many, but not all, of the VFHE participants explicitly commented on the visual fidelity of the avatars as an aid to presence. In fact, all comments on avatar realism from VFHE related to visual accuracy.

- "Nice to have skin tones, yes (I did identify with them)"
- "Yeah, those were my hands, and that was cool... I was impressed that I could see my own hands"
- "My hands looked very realistic… Yeah, they looked very real."
- "Appearance looked normal, looked like my own hands, as far as size and focus looked absolutely normal… I could see my own hands, my fingers, the hair on my hands"

From the interviews, participants who saw a visually faithful avatar assumed that the movement would also be accurate. From this we hypothesize that for VE users, visual fidelity encompasses kinetic fidelity.

Many participants reported that while engaged in the task, they believed completely they were in the presented virtual environment. In all the environments, head tracking and seeing other objects populating the virtual environment were the most commonly reported as factors that added to the presence.

Perhaps two quotes from the participants sum up the reconstructed avatars best:

- "I thought that was really good, I didn't even realize so much that I was virtual. I didn't focus on it quite as much as the blocks. "
- "I forget… just the same as in reality. Yeah, I didn't even notice my hands."

Debriefing Results – Minor Trends.

- Among the HE and VFHE participants, 75% noticed the reconstruction errors and 25% noticed the reconstruction lag. Most in the HE and VFHE complained of the limited field of view of the working environment. Interestingly, the RSE had a similar limited working volume and field of view, but no participant mentioned it.

- 65% of the VFHE and 30% of the HE participants noted their avatar looked real.
- 93% of the PVE and 13% of the HE and VFHE participants complained that the interaction with the blocks was unnatural.
- 25% of the HE and VFHE participants felt the interaction was natural.

When asked what **increased their sense-of-presence** in the VE:

- 26% of the HE and VFHE participants said that having the real objects and tactile feedback increased their sense-of-presence.

When asked what **decreased their sense-of-presence** in the VE:

- 43% of PVE participants commented that the blocks not being there or behaving as expected reduced their sense-of-presence.
- 11% of HE and VFHE participants also mentioned that manipulating real objects decreased their sense-of-presence because "they reminded them of the real world."

Finally, participants were asked how many patterns they needed to practice on before they felt comfortable interacting with the virtual environment. Based on their responses, VFHE participants felt comfortable significantly more quickly than PVE participants ($T_{26} = 2.83$, $p=0.0044$) at the $\alpha=0.01$ level. Participants were comfortable with the workings of the VE almost an entire practice pattern earlier (1.50 to 2.36 patterns).

Observations.

- Two-handed interaction greatly improved performance over one-handed interaction.
- All participants quickly developed a partitioning algorithm to assist them in solving the patterns. Participants would mentally grid the target pattern into either 4 or 9 squares. Then for each target pattern subsection, participants would grab a block and try to locate the matching face.
- The typical methodology for manipulation was to pick up a block, rotate it to a different orientation, and check if the new face is the desired pattern. If not, rotate again. If it is, place the

block and get the next block. The interactions to rotate the block dominated the difference in times between VE conditions.

- The next most significant component of task performance was the selection and placement of the blocks. Both these factors were improved through the natural interaction, motion constraints, and tactile feedback of real blocks.

Interesting Results. Using the pinch gloves had some unexpected fitting and hygiene consequences in the fourteen-participant PVE group.

- The pinch gloves are a one-size-fits-all, and two members had large hands and had difficulty fitting into the gloves.
- Two of the participants had small hands and had difficulty registering pinching actions because the gloves' pinch sensors were not positioned appropriately.
- One participant became nauseated and had to quit the experiment before finishing all the patterns. He mentioned fatigue and a reported a high level of simulator sickness. The pinch gloves quickly became moist with his sweat. This was a hygiene issue for subsequent participants.

Some PVE participants physically mimicked their virtual avatar's motions. Initially, they easily pinched to pick up virtual blocks. Recall that visually, the participants would see their virtual hand half-close to grab a virtual block as shown in Figure 33, when physically their hand would be making a pinching motion as shown in Figure 32. Some participants began to make the same hand motion that they saw, instead of the pinching motion required to select a block as shown in Figure 34. This caused no selection to be made and confused the participant. We noticed this phenomenon in the pilot study. In the main study, when the experimenter observed this behavior, he reminded the participant to make pinching motions to grasp a block.

Figure 32 – The PVE pinching motion needed to select a block.

Figure 33 – Images that participant saw when grabbing a block.

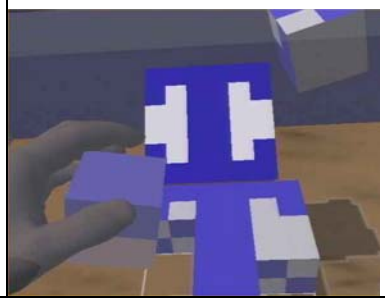


Figure 34 – Some participants started grasping midway, trying to mimic what they saw.

The PVE embodied several interaction tricks to provide an easy shortcut for some common tasks. For example, blocks would float in midair if the participant released the block more than six inches above the table. This eased the rotation of the block and allowed a select, rotate, release mechanism similar to a ratchet wrench. Participants, in an effort to maximize efficiency, would grab blocks and place them all in midair before the beginning of a test pattern. This allowed easy and quick access to blocks. The inclusion of the shortcuts was carefully considered to assist in interaction, yet led to adaptation and learned behavior that might be detrimental for training tasks.

In the RSE, participants worked on matching the mentally subdivided target pattern one subsection at a time. Each block was picked up and rotated until the desired face was brought into view. Some participants noted that this rotation could be done so quickly that they could just randomly spin each block to find a desired pattern.

In contrast, two PVE and one HE participant remarked that the slower interaction of grabbing and rotating a block in the VE influenced them to memorize the relative orientation of the block faces to improve performance.

Manipulating real objects also benefited from natural motion constraints. Tasks such as placing the center block into position in a nine-block pattern and closing gaps between blocks were easily done with real objects. In the PVE condition (all virtual objects), these interaction tasks would have been difficult and

time-consuming. We removed collision detection and provided snapping upon release of a block to alleviate these handicaps.

When we were developing the different conditions, we started with a basic VE renderer that allowed the user to walk around a scene. We wanted to augment the scene with the block manipulation task, which included simple models of the enclosure, target pattern, and in the PVE case virtual blocks and user avatar. The PVE condition required over three weeks of development for coding specialized tracking, interaction, shadowing, and rendering code. In contrast, the HE and VFHE conditions were developed within hours. The incorporation of real blocks and avatars did not require writing any code. The reconstruction system obviates prior modeling and incorporating additional trackers. Further, the flexibility of using real objects and the reconstruction system enabled minor changes to be made to the HE and VFHE conditions without requiring much code rework.

5.9 Conclusions

We conducted a study to evaluate the effects of interacting with real objects and of visually faithful avatars on task performance and presence in a spatial cognitive task VE. From our results, we conclude:

Interacting with real objects significantly improves task performance over interacting with virtual objects in spatial cognitive tasks, and more importantly, it brings performance measures closer to that of doing the task in real space. In addition, the way the participant performs the task in the VE using real objects is more similar to how they would do it in a real environment. Even in our simple task, we saw evidence that manipulating virtual objects sometimes caused mistraining of manipulation actions and participants to develop VE-specific approaches to the task.

Training and simulation VEs are specifically trying to recreate real experiences, and would benefit substantially from having the participant manipulate as many real objects as possible. The motion constraints and tactile feedback of the real objects provide additional stimuli that create an experience much closer to the actual task than one with purely virtual objects. Even if a real-object reconstruction system is

not employed, we believe that instrumenting, modeling and tracking the real objects that the participant will handle would significantly enhance spatial cognitive tasks.

Motion fidelity is more important than visual fidelity for self-avatar believability. We hypothesize that motion fidelity is the primary component of self-avatar believability. We believe that a visually faithful avatar is better than a generic avatar, but from a sense-of-presence standpoint, the advantages do not seem very strong.

Designers should focus their efforts first to focus on tracking then on rendering the user avatar model for immersive VEs. If an real-object reconstruction system is not employed, we believe that texture mapping the self-avatar model with captured images of the user would provide high quality motion and visual fidelity and result in a substantial immersion benefit.

6. NASA Case Study

Motivation. In order to evaluate the potential utility of this technology in a real-world task, we applied our reconstruction system to an assembly verification task. Given virtual models of complex multipart devices such as satellites and engines, designers want to determine if assembling the device is physically possible. Answering this question involves managing parts, various tools, and people with a large variance in shape. Experimenting with different designs, tools, and parts using purely virtual objects requires generating or acquiring virtual models for all the objects in the environment, and tracking those that are moved. We believe that this additional work impedes the use of VEs for evaluating multiple designs quickly and interactively. Further our user study results suggest that the lack of haptic feedback lowers the overall VE effectiveness for such hands-on tasks as those found in assembly planning and verification.

Using a hybrid VE, one that combines real and virtual objects, allows the participant to interact with the virtual model using real tools and critical parts with his own hands. We believe this would benefit assembly verification tasks.

6.1 NASA Collaboration

We have begun a collaboration with the NASA Langley Research Center (NASA LaRC) to see how using our system could assist in evaluating payload designs and assembly layouts. Space planning errors can have a significant impact in terms of money, scheduling, and personnel. We have worked with NASA experts in a variety of engineering, science, and technical disciplines to identify tasks critical to their work that would potentially benefit from hybrid VEs. Data concerning NASA LaRC motivations, comments, and suggestions are taken directly from oral or written responses to surveys, interviews, and informal remarks during experiments and discussions.

Driving Problems. NASA LaRC payload designers are interested in examining models of payloads and payload subsystems for two major tasks, assembly verification and assembly training.

NASA LaRC payload designers want to discern possible assembly, integration, and testing problems early in the project development cycle. Currently, different subsystems are separately subcontracted out. The integration of the many different subsystems always generates compatibility and layout issues, even with the greatest care in the specification of subsystem design.

Currently, it is difficult to evaluate the interaction of the different subpayloads, as the complexity and nuances of each component are understood well only by the group that developed that subsection. For example, attaching external cables is a common final integration task. With each payload being developed separately, the NASA LaRC designers described several occasions when they encountered spacing problems during the final cable attachment step. The payloads had conformed to specifications, but the reality of attaching the cables showed inadequate space for hands, tools, or parts. These layout issues resulted in schedule delays, equipment redesign, or makeshift engineering fixes.

Currently, simplified physical mock-ups are manufactured for design verification and layout, and the assembly procedure is documented in a step-by-step instruction list. The NASA LaRC payload designers recounted several occasions when the limited fidelity of mock-ups and assembly documents caused significant problems to slip through to later stages.

Given payload models, NASA LaRC payload designers want to train technicians in assembly and maintenance procedures. Much of the equipment is specific to a given payload, and training on virtual models would provide repetition and enable *more* people to become proficient in critical assembly stages. Also, beginning training before physical mock-ups or the actual devices are available would increase the amount of time to train. This would be useful, because certain tasks, such as releasing a delicate paraffin latch properly, require highly specific skills.

LaRC designers currently receive payload subsection CAD models from their subcontractors early in the design stage, before anything gets built. They would like to use these models to investigate assembly, layout, and integration. Changes in the early project stages are substantially cheaper in money, time, and personnel than fixes in later stages. With the critical time constraints for payload development, testing multiple design alternatives quickly would be valuable. A virtual environment potentially offers such an ability.

We believe that a hybrid VE system would enable designers to test configurations using the final assembly personnel, real tools, and parts. We hypothesize that such a hybrid VE would be a more effective system for evaluating hardware designs and planning assembly than a purely virtual one.

6.2 Case Study: Payload Spacing Experiment

Overview. To evaluate the applicability of hybrid VEs to NASA LaRC assembly tasks, we designed an abstracted payload layout and assembly task for four LaRC payload designers. We presented the designers with task information in approximately the same manner as they receive it in actual design evaluation. They discussed approaches to the task and then executed the assembly procedure in the hybrid VE. We interviewed the designers on how useful the hybrid system would be for tasks they currently have in payload assembly, testing, and integration.

Our first step in understanding the issues involved in payload assembly was to visit the NASA LaRC facilities and meet with engineers and technicians. They showed us the different stages of developing a payload, and outlined the issues they regularly face. Specifically, we were shown a weather imaging satellite, the CALIPSO project, and a light imager unit on the satellite called the photon multiplier tube (PMT). Figure 35 shows an engineering mock-up of the real PMT, without the imager tube that fits in the center cylindrical channel.

We received CAD models of the PMT (Figure 36), and abstracted a task that was similar to many of the common assembly steps, such as attaching components and fastening cable connectors.

Figure 35 – Engineering mock up of PMT box without center imaging tube.

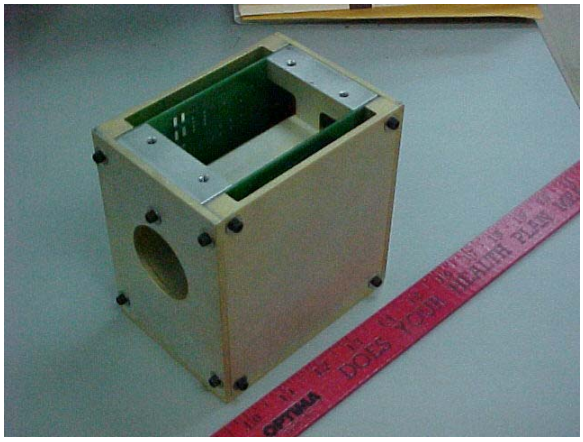
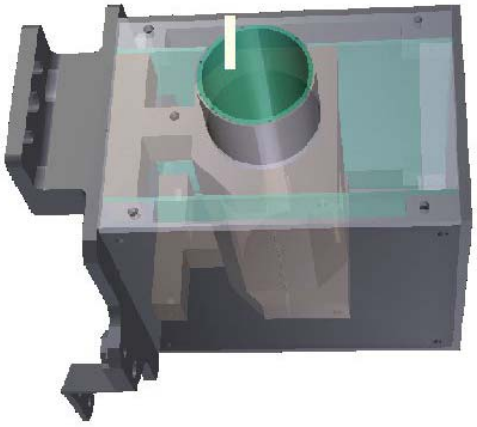


Figure 36 - CAD model of PMT box.



Assembly Task Description. The PMT model, along with two other payloads (payload A and payload B), was rendered in the VE. The system performed collision detection among the virtual payloads and the real-object avatars. The system indicated collisions by rendering in red the virtual object in collision as shown in Figure 37.

Figure 37 – Collisions between real objects (pipe and hand) and virtual objects (payload models) cause the virtual objects to flash red.

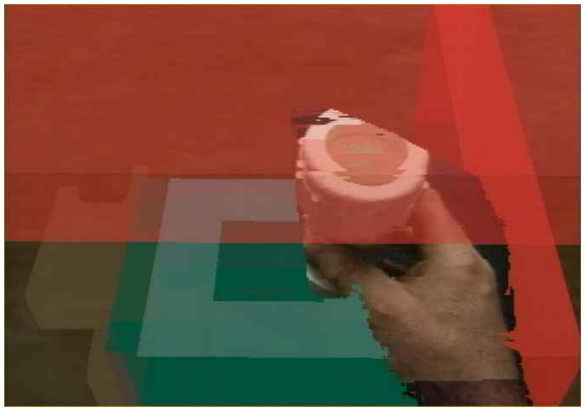


Figure 38 – Parts used in the shield fitting experiment. PVC pipe prop, power cord, tongs (tool), and the outlet and pipe connector that was registered with the virtual model.

Using real objects (Figure 38) and interacting with the PMT model, the participant was to screw a cylindrical shield (mocked-up as a PVC pipe) (Figure 41, Figure 42) into a pipe receptacle and then plug a power connector into an outlet inside the shield (Figure 43, Figure 44). If the participant required

additional assistance, we provided tools to aid in the task (Figure 45, Figure 46). The designers were to determine how much space was required between the *top* of the PMT box and the *bottom* of payload A as shown in Figure 39. A diagram of the task is shown in Figure 40.

Figure 39 – The objective of the task was to determine how much space between the PMT and the payload above it (red arrow) is required to perform the shield and cable fitting task.

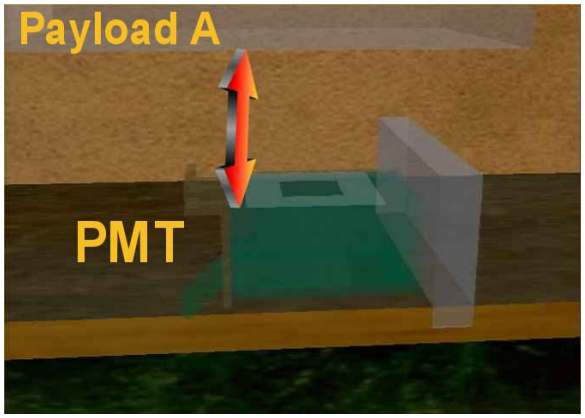


Figure 40 – Cross-section diagram of task. The pipe (red) and power cable (blue) need to be plugged into the corresponding connector down the center shaft of the virtual PMT box.

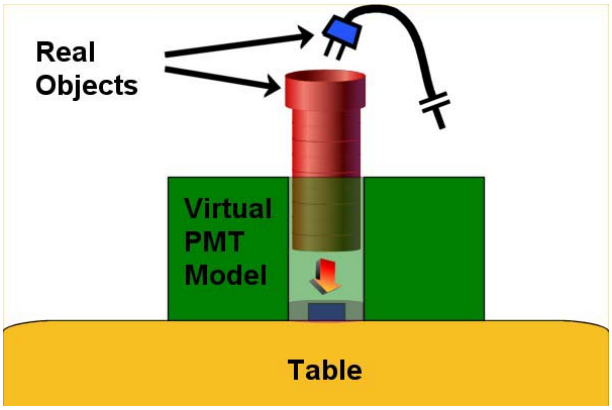


Figure 41 – The first step was to slide the pipe between the payloads and then screw it into the fixture.

Figure 42 – 3rd person view of this step.

Figure 43 – After the pipe was in place, the next step was to fish the power cable down the pipe and plug it into the outlet on the table.

Figure 44 – 3rd person view of this step. Notice how the participants holds his hand very horizontally to avoid colliding with the virtual PMT box.

Figure 45 – The insertion of the cable into the outlet was difficult without a tool. Tongs were provided to assist in the plugging in the cable.

Figure 46 – 3rd person view of this step.

Experimental Procedure. On March 12, 2002, four NASA LaRC payload designers and engineers performed the task experiment using our reconstruction system. We asked the participants to fill out a survey (Appendix C.1) before attempting the task. We provided basic information about the size and orientation of different components and connectors, and specifically the pipe (14 cm long, and 4 cm in diameter) they needed to attach into the receptor at the bottom of the PMT box cylinder. The survey asked:

- How much space between the PMT and payload A is necessary to perform the pipe insertion and power cable attachment procedures?
- How much space between the PMT and payload A would you actually allocate (given typical payload layout space constraints) for the pipe insertion and power cable attachment procedures?

After completing the survey, each participant performed the pipe insertion and power cable attachment procedure in the reconstruction system.

- First, participants donned the HMD and walked around the VE to get used to it.
- Next, they tested the collision detection system by moving their hands into intersection with the PMT box and then with payload A to see the visual results (rendered in red) of collisions with virtual objects.
- Then, they picked up the pipe and eased it into the center cylindrical assembly while trying to avoid colliding with either payload A or the PMT box.
- After the pipe was lowered into the cylindrical shaft of the PMT, they snaked the power cord down the tube and inserted it into the outlet.

The experimenter could dynamically adjust the space between the PMT and payload A. As the participant asked for more or less space, the experimenter adjusted the height of payload A (moving it up and down). With this interaction, different spatial configurations of the two payload subassemblies could be quickly evaluated.

The post-experience survey focused on the participant's reaction to the actual space required between the PMT and payload A, as interactively determined while they were in the VE. This survey is summarized in Table 11 and attached as Appendix C.2. The responses of all participants are attached as Appendix C.3.

Results. Given that the pipe was 14 cm long and 4 cm in diameter:

Table 11 – LaRC participant responses and task results

	Participant #			
	#1	#2	#3	#4
(Pre-experience) How much space is necessary between Payload A and the PMT?	14 cm	14.2 cm	15 - 16 cm	15 cm
(Pre-experience) How much space would you actually allocate?	21 cm	16 cm	20 cm	15 cm
Actual space required (determined in VE)	15 cm	22.5 cm	22.3 cm	23 cm
(Post-experience) How much space would you actually allocate after your VE experience?	18 cm	16 cm (modify tool)	25 cm	23 cm

Space is scarce and the engineers were stingy with it. This was especially true for participants #2 and 4 who had experience with actual payload assembly. Participant #3 was a flight software engineer and had less experience with actual installing payload hardware.

Each participant was able to complete the task. Participant #1 was able to complete the task without using a special tool, since the power cable was stiff enough to force into the outlet. Since an aim was to impress upon the participants the possibility of requiring unforeseen tools in assembly or repair, we used a more flexible cable for the remaining participants. While trying to insert the power cable, participants #2, 3, and 4 noted they could not complete the task. When asked what they required, they all remarked they wanted a tool to assist in plugging in the cable. They were handed a tool (a set of tongs) and were then able to complete the power cable insertion task as shown in Figure 46. Using the tool required increasing the spacing between the PMT and Payload A so as to avoid collisions. Interactively changing the spacing allowed testing new spacing design layouts while in the VE. The use of the tool increased the required spacing between the PMT box and payload A from 14 cm to an average 24 cm.

The more flexible power cable could not be snaked down the pipe and inserted into the outlet without some device to help push the connector when it was inside the pipe. This was because the pipe was too narrow for the participant's hands. Connecting the power cable before attaching the pipe still has the same spacing issues. The virtual PMT box still hinders the attachment of the power cable, regardless of whether or not the pipe has been inserted (Figure 40).

Whereas in retrospect it was obvious that the task would not be easily completed without an additional tool, not one of the designers anticipated this requirement. We believe the media by which the assembly information was provided (diagrams, task descriptions, and assembly drawings), made it difficult for designers, even though each had substantial payload development experience, to catch subtle assembly integration issues. On average, the participants allocated 5.6 cm too little space between the payloads on their pre-experience surveys.

The hybrid VE system provided readily identifiable benefits over purely virtual approaches for conducting the assembly verification task quickly and effectively:

- The participants saw themselves, tools, and critical parts within the environment.
- The interaction with the VE was very natural, and participants needed no instruction. After having it explained that virtual parts would change color when in collision, participants began carrying out the task almost immediately after putting on the HMD.
- Participants quickly adapted hand positions to avoid collisions with the virtual payload models (See Figure 44).
- The system could accommodate incorporating various tools extemporaneously, without either prior modeling or any additional development. Different layouts, task approaches, and tools could be evaluated quickly.
- The motion constraints of the pipe threads and the power cable socket aided interaction with these objects. Purely virtual approaches would be hard-pressed to provide comparable interactions.

Physical mock-ups are more costly to build than virtual models, require substantial time to create, and have varying degrees of fidelity to the final payload. These characteristics reduce their use early in the design evaluation stage. The NASA LaRC personnel recounted a scenario in which even the high-quality replications used in later development stages had simplifications that hurt the final integration. "Connector savers", cable connectors replications used to reduce the wear on the actual connectors, did not contain a small bolt that was in the designs, and on the actual connector. When the final cable was to be attached, the bolt did not allow a proper fit of the cable into the connector. The NASA engineers recounted that they had to force the cable down, turn the cable so at least a few threads were holding it connected, and hope that the launch vibration would not unseat the cable and ruin years of work.

Compared to physical mock-ups, hybrid VEs can provide a cheaper and quicker alternative system for evaluating designs and layouts, especially in the early phases. Further, as illustrated in the above example, there are occasions when using full CAD models provides a distinct advantage for evaluation tasks over physical mock-ups, which will contain simplifications.

Debriefing. The three participants who required the tool to complete the task were *extremely* surprised that a tool was needed and that so much additional space was required to accommodate the tool. When they discovered the required spacing was much more than the amount they allocated, they immediately commented on the potential time and schedule savings of evaluating designs at the model stage.

The post-experience survey asked the participants to quantify the time and financial savings that early identification of the spacing error would provide.

Table 12 - LaRC participant responses to the financial and scheduling implications of early identification of integration spacing errors

	Participant #			
	#1	#2	#3	#4
Time cost of spacing error	days to months	30 days	days to months	months
Financial cost of spacing error	$100,000s - $1,000,000+	largest cost is huge hit in schedule	$100,000s - $1,000,000+	$100,000s

All participants responded that the financial implications could be anywhere from moderate (hundreds of thousands of dollars), such as keeping personnel waiting till a design fix was implemented, to extreme (millions of dollars), such as causing launch delays. In every payload design, time is the most precious commodity, and critical-path delays could even result in missing a launch date. Every participant mentioned that identifying problems such as the spacing error would provide the largest benefit in reducing schedule delays.

Participants exhibited interesting interaction methods with the virtual model. The virtual model was not very detailed, and the visual contrast between real and virtual objects was rather obvious. Yet, participants made concerted efforts to avoid touching the virtual model. Upon being told about his effort to avoid touching the purely virtual PMT box, a participant said, "that was flight hardware… you don't touch flight hardware." The familiarity and relevancy of the task made the experience vivid for the participants. Participants commented that their actions in the VE were very similar to how they would actually approach the task.

After completing the task, participants remarked that VEs and object reconstruction VEs would be useful for the following payload development tasks:

- Assembly training.
- Hardware layout (including cable routing and clearance testing).
- Evaluating designs for equipment integration and fitting.
- Evaluating designs for environmental testing – e.g., how to arrange a payload inside a thermal-vacuum chamber.

Lessons Learned. The NASA LaRC payload designers and engineers were very optimistic about applying traditional VEs, object reconstruction VEs, and simulations to aid in payload development. They are interested in looking at virtual models to evaluate current payload integration tasks and upcoming payload designs.

There are substantial gains to be realized by using virtual models in almost every stage of payload development. But, using virtual models has the most significant benefit in the design stage. Further, early identification of assembly, integration, or design issues would result in considerable savings in terms of time, money, and man-hours. Many of their tasks involve technicians interacting with a payload with tools and parts. These tasks are well suited to be simulated within an object reconstruction VE.

7. Summary and Future Work

We have developed a system for incorporating dynamic real objects into a hybrid environment. This involved developing algorithms for generating virtual representations of real objects in real time and algorithms for collision detection and response between these virtual representations and other virtual objects.

7.1 Review of results

Real-Time Object Reconstruction Algorithms. We have presented an algorithm that exploits graphics hardware to generate a real-time view-dependent sampling of real objects' visual hull from multiple camera views. The resulting virtual representations are used to render visually faithful participant self-avatars and as objects in simulations. The system does not require additional trackers or require *a priori* object information, and enables natural interaction between the real objects and the rest of the virtual environment.

Real – Virtual Object Interaction Algorithms. We extended the real-time image-based object reconstruction system to detect collisions between real and virtual objects and to respond plausibly. This required new algorithms for colliding polygonally-defined virtual objects with the dynamic real-object avatars.

User Study on Real Objects in VEs. We then conducted studies to evaluate the advantages that manipulating real objects could provide over manipulating purely virtual objects in a cognitive manual task.

The results suggest that manipulating and interacting with real objects in a VE provide a significant task performance improvement over interacting with virtual objects. We believe this is because the objects' interaction affordances are complete and proper and because the participant has haptic feedback.

The user study results did not show a significant difference in participant-reported sense-of-presence for those represented by a visually faithful personalized self-avatar over those represented by generic self-avatars. Those represented by the visually faithful self-avatars did, however, show a preference for the personalization. We have concluded that the principal attribute of avatars for presence is kinetic fidelity. Visual fidelity is important, yet apparently less so. We hypothesize that for participants, visual fidelity encompasses kinetic fidelity. If they see a visually faithful avatar, they *expect* it to move realistically as well. We feel with further studies, with a more developed presence questionnaire, could identify the effects of visually faithful avatars.

Table 13 - Comparing Participant sense of presence scores, avatar questions, and debriefing responses

	SUS Sense-of-Presence Score	**Association with avatar 1. Not very much… 7. Very much**	**Avatar realism 1. Not very realistic… 7. Very realistic**	**Any comments on your virtual body?**
PVE	3.21	4.43	3.64	Movement Accuracy
HE	1.86	4.79	4.57	Movement Accuracy
VFHE	2.36	4.64	4.50	Visual Accuracy

Applying the system. We believe that performance of many assembly verification tasks could be improved by allowing the participant to interact real objects in a VE. Our work with NASA LaRC has shown that the system could provide a substantial benefit in hardware layout and assembly verification tasks. The reconstruction system enables complex interactions with virtual models to be performed with real tools and parts. If in standard practice, this would allow more time and opportunities to train personnel to perform delicate and complex operations. Further, designers could evaluate payload development issues dealing with assembly verification, layout, and integration early in the design cycle.

7.2 Future Work

The current implementation of the reconstruction system is a prototype. The future work focuses on increasing algorithm performance, improving algorithm results, and examining other VE interaction research directions.

Reconstruction Algorithm. The most urgent system improvement is to port it to a networked cluster of PCs. Currently the system runs on a SGI Reality Monster graphics supercomputer. The high cost, limited number in service, and infrequent hardware upgrades makes it a poor platform if the system is to become widely used. Cost is the primary factor. The mass-market forces that drive development of commodity hardware have led to low prices and rapid progress. The large advances in consumer-grade computers, networking, and graphics cards have made a networked PC-based system an attractive, continually evolving solution. Other current image-based scene reconstruction algorithms are already making use of networked PCs. Examples are the Virtualized Reality [Baba00], 3-D Tele-Immersion [Raskar98] and Image-Based Visual Hull projects [Matusik00].

A possible hardware configuration for image-based scene reconstruction would be as follows. A dedicated network of PCs, each connected via high bandwidth (e.g. Firewire) to high-quality high-resolution cameras, capture the scene. Each PC has enough computation power to perform the image segmentation required by the algorithm, plus filtering not now performed. Next, the resulting images are sent, possibly with compression, to a central PC that does the reconstruction.

For our algorithms, the newer high bandwidth PC buses from the system memory to texture memory (such as AGP 4x and AGP 8x) provide the necessary throughput for uploading the camera images into graphics card texture memory in real time. The graphics requirements of the algorithm are not very high, and current graphics cards, (e.g. ATI Radeon or nVidia GeForce4) can provide interactive performance.

Porting the algorithm to PCs would allow us to benefit from the performance and feature improvements embodied in the constantly advancing graphics hardware. With new generations of faster graphics cards, the reconstruction algorithm would be able to provide results more rapidly and/or with more resolution.

To further improve the speed of the reconstruction algorithm, we also look at methods for reducing the fill rate of the algorithm, the current bottleneck. One optimization would be to compute bounding boxes for object pixel clusters in the object-pixel maps. Then for a plane in the plane-sweeping stage of the

reconstruction, the bounding boxes can be projected onto the plane and tested for intersections to reduce the size of, or eliminate completely, the plane being volume-queried. This reduces the fill rate requirement of the algorithm.

A big improvement in system accuracy would come from using higher-resolution cameras (which would demand higher performance). Further work also needs to be done on the quality of the input into the reconstruction algorithms. The results of image segmentation are sensitive to shadows and to high-spatial-frequency background image. The image segmentation errors result in increased visual hull size caused by incorrectly labeled object pixels, and holes in the visual hull caused by incorrectly labeled background pixels. Applying filtering and noise-reduction image-processing algorithms on the input camera images and using more rigorous and precise camera calibration would improve the visual hull shape and size accuracy. Computer control of workspace illumination would enable instantaneous selective blinking so that shadows can be distinguished from real objects.

Collision detection accuracy could be improved through better use of framebuffer resolution. The resolution of collision detection is dependent on the size of the viewport the primitive is rendered into during volume-querying. Thus using viewports whose size depends on the primitive being rendered would achieve a uniform spatial resolution for detecting collisions. Any improvements in visual hull shape accuracy would also improve collision detection accuracy.

Collision response could be improved by finding better algorithms for determining penetration depth, collision points, and surface normals. The results of the current algorithm provide only estimates. Volume querying with different primitives, such as distance fields, during collision detection could result in more accurate results for collision response.

The inability to backtrack the motions of real objects limits our ability to recover realistically from interpenetration of real and virtual objects. Caching previous camera images, and tracking real objects within these camera images, would enable backtracking. Examining the shape and motion of a tracked

object across several frames, would enable information, such as object velocity and rotation, to be derived. This additional information would allow simulations to compute more realistic collision responses.

User Study. Possibly the most interesting area of future research, is the study of avatars and interaction in immersive VEs in ways enabled by the reconstruction technique.

Does avatar visual fidelity affect presence in virtual environments? We believe it does. If so, how strong is the effect? Even though our user study does not show a significant difference in presence due to avatar appearance, the user interviews leads us to believe there is some effect.

The block design matching task we used might have been too cognitively engrossing for post-experience questionnaires to focus on sense-of-presence. Some participants reported a low level of immersion while "just looking around", but a high level of immersion while performing the task. Future work would involve identifying tasks, behavioral measures, and associated metrics that can isolate and separate the effect of avatar visual fidelity and avatar motion fidelity on sense-of-presence in VEs.

8. Bibliography

[Abdel-Aziz71] Y. Abdel-Aziz and H. Karara. "Direct Linear Transformation from Comparator Coordinates Into Object Space Coordinates in Close-Range Photogrammetry", In *Proceedings of the Symposium on Close-Range Photogrammetry*. Falls Church, VA: American Society of Photogrammetry, pp. 1-18.

[Arthur00] K. Arthur, "Effects of Field of View on Performance with Head-Mounted Displays", Department of Computer Science, UNC-Chapel Hill, 2001, Unpublished dissertation.

[Baba00] S. Baba, H. Saito, S. Vedula, K.M. Cheung, and T. Kanade. "Apperance-Based Virtual-View Generation for Fly Through in a Real Dynamic Scene", In *Proceedings of VisSym '00* (Joint Eurographics – IEEE TCVG Symposium on Visualization), May, 2000.

[Baciu99] G. Baciu, W. Wong and H. Sun. "RECODE: An Image-based Collision Detection Algorithm", In *Proceedings* of the *Journal of Visualization and Computer Animation*, Vol. 10, No. 4, 1999, pp. 181-192.

[Badler99] N. Badler, R. Bindiganavale, J. Bourne, J. Allbeck, J. Shi, and M. Palmer. "Real Time Virtual Humans", In *Proceedings of International Conference on Digital Media Futures*, British Computer Society, Bradford, UK, April, 1999.

[Banerjee99] A. Banerjee, P. Banerjee, N. Ye, and F. Dech. "Assembly Planning Effectiveness Using Virtual Reality", Presence, Vol. 8, No. 7, pp. 204-217, 1999.

[Bouguet98] J. Bouguet. "Camera Calibration from Points and Lines in Dual-Space Geometry", Technical Report, Department of Computer Science, California Institute of Technology, 1998.

[Bowman97] D. Bowman and L. Hodges. "An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments", In 1997 ACM Symposium on Interactive 3-D

Graphics, pp. 35-38 (April 1997). ACM SIGGRAPH. Edited by Michael Cohen and David Zeltzer. ISBN 0-89791-884-3.

[Boyles00]. M. Boyles and S. Fang. "Slicing-Based Volumetric Collision Detection", *ACM Journal of Graphics Tools*, Vol. 4, No. 4, pp. 23-32, 2000.

[Breen95] D. Breen, E. Rose, R. Whitaker. "Interactive Occlusion and Collision of Real and Virtual Objects in Augmented Reality", Munich, Germany, European Computer Industry Research Center, 1995.

[Brooks99] F. Brooks Jr. "What's Real About Virtual Reality?" *IEEE Computer Graphics and Applications,* Vol 19, No. 6, pp. 16-27.

[Bush99] T. Bush. "Gender Differences in Cognitive Functioning: A Literature Review", *The Cyber-Journal of Sport Marketing*, Vol. 1.

[Butcher00] J. Butcher, C. Bass, and L. Danisch. "Evaluation of Fiber-Optic Sensing Band For the Characterization of Deformation Contours", *Southern Biomedical Engineering Conference* 2000.

[Carr98] J. Carr, W. Fright, A. Gee, R. Prager and K. Dalton. "3-D Shape Reconstruction using Volume Intersection Techniques", In *IEEE International. Conference on Computer Vision Proceedings*, 1095-1110, January 1998.

[Castleman96] K. Castleman. Digital Image Processing, Prentice Hall, Upper Saddle River, New Jersey, 1996, pp. 452-460.

[Chien86] C. Chien and J. Aggarwal. "Volume/Surface Octrees for the Representation of Three-Dimensional Objects", *Computer Vision, Graphics, and Image Processing,* Vol. 36, No. 1, pp. 100-113, October 1986.

[Daniilidis00] K. Daniilidis, J. Mulligan, R. McKendall, G. Kamberova, D. Schmid, R. Bajcsy. "Real-Time 3-D Tele-immersion", In *The Confluence of Vision and Graphics*, A Leonardis *et al*. (Eds.), Kluwer Academic Publishers, 2000, pp. 253-266.

[Edelsbrunner94] H. Edelsbrunner and E. Mucke. "Three-Dimensional Alpha Shapes", *ACM Transactions on Graphics*, Vol 13. 1994, pp. 43-72.

[Ehmann01] S. Ehmann and M. Lin. "Accurate Proximity Queries between Polyhedra Using Surface Decomposition", *Computer Graphics Forum (Proceedings of Eurographics),* 2001.

[Faugeras93a] O. Faugeras, *Three Dimensional Computer Vision*, The MIT Press, 1993.

[Faugeras93b] O. Faugeras, T. Vieville, E. Theron, J. Vuillemin, B. Hotz, Z. Zhang, L. Moll, P. Bertin, H. Mathieu, P. Fua, G. Berry, and C. Proy. "Real-time Correlation-Based Stereo: Algorithm, Implementations and Applications", INRIA Technical Report RR-2013, 1993.

[Foley96] J. Foley, A. Van Dam. S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice 2nd Edition in C*. Addison-Wesley Publishing Company. 1996, pp. 745-753.

[Garau01] M. Garau, M. Slater, S. Bee, and M.A. Sasse. The Impact of Eye Gaze on Communication Using Humanoid Avatars. In *Proceedings of the SIG-CHI Conference on Human Factors in Computing Systems*, March 31- April 5, 2001, Seattle, WA USA, pp. 309-316.

[Hand97] C. Hand. A Survey of 3-D Interaction Techniques, *Computer Graphics Forum*, Vol. 16, No. 5, pp. 269-281 (1997). Blackwell Publishers. ISSN 1067-7055.

[Heeter92] C. Heeter. "Being There: The Subjective Experience of Presence", Presence, Teleoperations and Virtual Environments, Vol. 1, No. 2, pp 262-271.

[Hilton00] A. Hilton, D. Beresford, T. Gentils, R. Smith, W. Sun, and J. Illingworth. "Whole-Body Modelling of People from Multiview Images to Populate Virtual Worlds", *The Visual Computer*, Vol. 16, No. 7, pp. 411-436, 2000. ISSN 0178-2789.

[Hinckley94] K. Hinckley, R. Pausch, J. Goble, and N. Kassell. "Passive Real-World Interface Props for Neurosurgical Visualization", In *Proceedings of the 1994 SIG-CHI Conference*, pp 452-458.

[Hoffman97] H. Hoffman, A. Carlin, and S. Weghorst. "Virtual Reality and Tactile Augmentation in the Treatment of Spider Phobia", *Medicine Meets Virtual Reality 5*, San Deigo, California, January, 1997.

[Hoffman98] H. Hoffman. "Physically Touching Virtual Objects Using Tactile Augmentation Enhances the Realism of Virtual Environments", In *Proceedings of the IEEE Virtual Reality Annual International Symposium '98*, Atlanta GA, p. 59-63. IEEE Computer Society, Los Alamitos, California.

[Hoff01] K. Hoff, A. Zaferakis, M. Lin, and D. Manocha. "Fast and Simple 2-D Geometric Proximity Queries Using Graphics Hardware", *2001 ACM Symposium on Interactive 3-D Graphics,* pp. 145-148, 2001.

[Hollnagel02] E. Hollnagel, Handbook of Cognitive Task Design. To be published by Lawrence Erlbaum Associates, Inc. 2002.

[Insko01] B. Insko. "Passive Haptics Significantly Enhances Virtual Environments", Department of Computer Science, UNC-Chapel Hill, 2001, Unpublished dissertation.

[Levoy00] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. "The Digital Michelangelo Project: 3-D Scanning of Large Statues", In *Proceedings of ACM SIGGRAPH 2000*, pp. 131-144, 2000.

[Lindeman99] R. Lindeman, J. Sibert, and J. Hahn. "Hand-Held Windows: Towards Effective 2D Interaction in Immersive Virtual Environments", In *IEEE Virtual Reality*, 1999.

[Kakadaris98] I. Kakadiaris and D Metaxas. "Three-Dimensional Human Body Model Acquisition from Multiple Views", *International Journal of Computer Vision 30*, 1998.

[Kutulakos00] K. Kutulakos. "Approximate N-View Stereo", In *Proceedings, 6th European Conference on Computer Vision*, Dubin, Ireland, pp. 67-83, 2000.

[Laurentini4] A. Laurentini. "The Visual Hull Concept for Silhouette-Based Image Understanding", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 16, No. 2, 150-162, February 1994.

[Lin98] M. Lin, S. Gottschalk. "Collision Detection between Geometric Models: A Survey", In *Proceedings of the IMA Conference on Mathematics of Surfaces*, 1998.

[Lok01] B. Lok. "Online Model Reconstruction for Interactive Virtual Environments", In *Proceedings 2001 Symposium on Interactive 3-D Graphics*, Chapel Hill, N.C., 18-21, March 2001, pp. 69-72, 248.

[Maringelli01] F. Maringelli, J. McCarthy, A. Steed, M. Slater and C. Umiltà. "Shifting Visuo-Spatial Attention in a Virtual Three-Dimensional Space", Cognitive Brain Research, Vol. 10, Issue 3, January 2001, pp. 317-322.

[Matusik00] W. Matusik, C. Buehler, R. Raskar, S. Gortler and L. McMillan. “Image-Based Visual Hulls”, In *Proceedings of ACM SIGGRAPH 2000*, Annual Conference Series, pages 369-374.

[Matusik01] W. Matusik, C. Buehler, and L. McMillan. "Polyhedral Visual Hulls for Real-Time Rendering", In *Proceedings of Twelfth Eurographics Workshop on Rendering,* London, England, June 2001, pp. 115-125.

[Meehan01] M. Meehan. “Physiological Reaction as an Objective Measure of Presence in Virtual Environments”, Department of Computer Science, UNC-Chapel Hill, 2001, Unpublished dissertation.

[Moezzi96] S. Moezzi, A. Katkere, D. Kuramura, and R. Jain. “Reality Modeling and Visualization from Multiple Video Sequences”, *IEEE Computer Graphics and Applications*, Vol. 16, No. 6, pp. 58-63, 1996.

[Mortensen02] J. Mortensen, V. Vinayagamoorthy, M. Slater, A. Steed, B. Lok, and M. Whitton. “Collaboration in Tele-Immersive Environments”, In *Proceedings of Eighth Eurographics Workshop on Virtual Environments* (EGVE 2002) on May 30-31, 2002.

[Niem97] W. Niem. "Error Analysis for Silhouette-Based 3D Shape Estimation from Multiple Views", *Proceedings on International Workshop on Synthetic - Natural Hybrid Coding and Three Dimensional Imaging* (IWSNHC3DI'97), Rhodos, 6-9 September, 1997.

[Pabst02] T. Pabst and L. Weinand. “PC Graphics Beyond XBOX – nVidia Inroduces GeForce4”, Retrieved March 28, 2002 from http://www6.tomshardware.com/graphic/02q1/020206/index.html.

[Pertaub01] D. Pertaub, M. Slater, and C. Barker. “An Experiment on Fear of Public Speaking in Virtual Reality”, *Medicine Meets Virtual Reality 2001*, pp. 372-378, J. D. Westwood *et al*. (Eds) IOS Press, ISSN 0926-9630.

[Potmesil87] M. Potmesil. “Generating Octree Models of 3-D Objects from Their Silhouettes in a Sequence of Images”, *Computer Vision, Graphics and Image Processing*. Vol. 40, pp. 1-29, 1987.

[Raskar98] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. “The Office of the Future: A Unified Approach to Image-Based Modelling and Spatially Immersive Displays”, *Computer*

Graphics. M. F. Cohen. Orlando, FL, USA (July 19 - 24), ACM Press, Addison-Wesley: pp. 179-188.

[Razzaque01] Razzaque, S. Z. Kohn, M. Whitton. "Redirected Walking", In *Proceedings of Eurographics 2001*, September 2001.

[Rehg94] J. Rehg and T. Kanade. "Digiteyes: Vision-Based Hand Tracking for Human-Computer Interaction", In J. Aggarwal and T. Huang, Eds., *Proceedings of Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 16-22, IEEE Computer Society Press, November, 1994.

[Satalich95] G. Satalich. "Navigation and Wayfinding in Virtual Reality: Finding Proper Tools and Cues to Enhance Navigation Awareness", Masters Thesis, Department of Computer Science, University of Washington, 1995.

[Seitz97] S. Seitz and C. Dyer. "Photorealistic Scene Reconstruction by Voxel Coloring", In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 1997, pp. 1067-1073.

[Simone99] L. Simone. Poser 4 (Review). Retrieved March 26, 2002 from http://www.zdnet.com/products/stories/reviews/0,4161,2313739,00.html.

[Slater93] M. Slater and M. Usoh. "The Influence of a Virtual Body on Presence in Immersive Virtual Environments", *VR 93, Virtual Reality International, Proceedings of the Third Annual Conference on Virtual Reality*, London, Meckler, 1993, pp 34-42.

[Slater94] M. Slater and M. Usoh. "Body Centred Interaction in Immersive Virtual Environments", in N. Magnenat Thalmann and D. Thalmann, Eds., *Artificial Life and Virtual Reality*, pp. 125-148, John Wiley and Sons, 1994.

[Sutherland65] I. Sutherland. "The Ultimate Display", In *Proceedings of IFIP* 65, Vol 2, pp 506, 1965.

[Thalmann98] D. Thalmann. "The Role of Virtual Humans in Virtual Environment Technology and Interfaces", In *Proceedings of Joint EC-NSF Advanced Research Workshop*, Bonas, France, 1998.

[Turk94] G. Turk and M. Levoy. "Zippered Polygon Meshes From Range Images", In *Proceedings of ACM SIGGRAPH 1994*, pp. 311-318, 1994.

[Usoh99] M. Usoh, K. Arthur, *et al*. "Walking > Virtual Walking> Flying, in Virtual Environments", *Proceedings of SIGGRAPH 99*, pp. 359-364, Computer Graphics Annual Conference Series, 1999.

[Usoh00] M. Usoh, E. Catena, S. Arman, and M. Slater. "Using Presence Questionnaires in Reality", *Presence: Teleoperators and Virtual Environments*, Vol. 9, No. 5, pp. 497-503.

[Ward01] M. Ward. "EDS Launches New Tool To Help Unigraphics CAD/CAM Software Users With Earlier Detection Of Product Design Problems", Retreived March 26, 2002 from http://www.apastyle.org/elecgeneral.html.

[Wechsler39] D. Wechsler. The Measurement of Adult Intelligence, 1st Ed., Baltimore, MD: Waverly Press, Inc. 1939.

[Welch96r] R. Welch, T. Blackmon, A. Liu, A. Mellers, and L. Stark. "The Effect of Pictorial Realism, Delay of Visual Feedback, and Observer Interactivity on the Subjective Sense-of-presence in a Virtual Environment", *Presence: Teleoperators and Virtual Environments*, Vol. 5, No. 3, pp. 263-273.

[Welch01g] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, and D. Colucci. "High-Performance Wide-Area Optical Tracking: The HiBall Tracking System", *Presence: Teleoperators and Virtual Environments* Vol. 10, No. 1: pp. 1-21, 2001.

[Zachmann01] G. Zachmann and A. Rettig. "Natural and Robust Interaction in Virtual Assembly Simulation", *Eighth ISPE International Conference on Concurrent Engineering: Research and Applications (ISPE/CE2001)*, July 2001, West Coast Anaheim Hotel, California, USA.

Appendix A User Study Documents

Pre-experience	Consent Form (A.1) Health Assessment (A.2) Kennedy-Lane Simulator Sickness (A.3) Guilford-Zimmerman Spatial Ability (A.4)
During Experience	Participant Experiment Record (A.5)
Post-experience	Debrief Form (A.6) Interview (A.7) Kennedy - Lane Simulator Sickness (A.3) Steed - Usoh - Slater Presence Questionnaire (A.8)

Appendix A.1 Consent Form

Task Performance and Presence in Virtual Environments

Introduction and purpose of the study:

We are inviting you to participate in a study of effect in virtual environment (VE) systems. The purpose of this research is to measure how task performance in VEs changes with the addition of visually faithful avatars (a visual representation of the user) and natural interaction techniques. We hope to learn things that will help VE researchers and practitioners using VEs to train people for real-world situations.

The principal investigator is Benjamin Lok (UNC Chapel Hill, Department of Computer Science, 361 Sitterson Hall, 962-1893, email: lok@cs.unc.edu). The Faculty advisor is Dr. Frederick P. Brooks Jr. (UNC Chapel Hill, Department of Computer Science, Sitterson Hall, 962-1931, email: brooks@cs.unc.edu).

What will happen during the study:

We will ask you to come to the laboratory for one session lasting approximately one hour. During the session, you will perform a simple task within the VE. During the experiment, you will wear a helmet containing two small screens about three inches in front of your eyes. You will also be wearing headphones in order to receive instructions. In the traditional VE condition, you will wear data gloves on your hands, and in the hybrid you'll wear generic white gloves. We will use computers to record your hand, head, and body motion during the VE experience. We will also make video and audio recordings of the sessions. You will be given questionnaires asking about your perceptions and feelings during and after the VE experience. Approximately 30 people will take part in this study.

Protecting your privacy:

We will make every effort to protect your privacy. We will not use your name in any of the data recording or in any research reports. We will use a code number rather than your name. No images from the videotapes in which you are personally recognizable will be used in any presentation of the results, without your consent. The videotapes will be kept for approximately two years before they are destroyed.

Risks and discomforts:

While using the virtual environment systems, some people experience slight symptoms of disorientation, nausea, or dizziness. These can be similar to "motion sickness" or to feelings experienced in wide-screen movies and theme park rides. We do not expect these effects to be strong or to last after you leave the laboratory. If at any time during the study you feel uncomfortable and wish to stop the experiment you are free to do so.

Your rights:

You have the right to decide whether or not to participate in this study, and to withdraw from the study at any time without penalty.

Payment:

You will be paid $10 for your participation in this study, regardless of completion of the task. No payment will be given to an individual who does not meet the criteria specified in the signup sheet or who does not meet the criteria which are determined on-site at the time of the experiment regarding health, stereo vision, and comfort and ease of use of the HMD.

Institutional Review Board approval:

The Academic Affairs Institutional Review Board (AA-IRB) of the University of North Carolina at Chapel Hill has approved this study. If you have any concerns about your rights in this study you may contact the Chair of the AA-IRB, Barbara Goldman, at CB#4100, 201 Bynum Hall, UNC-CH, Chapel Hill, NC 27599-4100, (919) 962-7761, or email: aa-irb@unc.edu.

Summary:

I understand that this is a research study to measure the effects of avatar fidelity and interaction modality on task performance and sense-of-presence in virtual environments.
I understand that if I agree to be in this study:

- I will visit the laboratory one time for sessions lasting approximately one hour.
- I will wear a virtual environment headset to perform tasks, my movements and behavior will be recorded by computer and on videotape, and I will respond to questionnaires between and after the sessions.
- I may experience slight feelings of disorientation, nausea, or dizziness during or shortly after the VE experiences.

I certify that I am at least 18 years of age.
I have had a chance to ask any questions I have about this study and those questions have been answered for me.
I have read the information in this consent form, and I agree to be in the study. I understand that I will get a copy of this consent form.

_____ _____

Signature of Participant Date

I am willing for videotapes showing me performing the experiment to be included in presentations of the research. ☐ Yes ☐ No

Appendix A.2 Health Assessment & Kennedy-Lane Simulator Sickness Questionnaire

Participant Preliminary Information

1. Are you in your usual state of good fitness (health)?
   YES
   NO

2. If NO, please circle all that apply:

Sleep Loss	Hang over	Upset Stomach	Emotional Stress	Upper Respiratory Ill.
Head Colds	Ear Infection	Ear Blocks	Flu	Medications

Other (please explain) _____

3. In the past 24 hours which, if any, of the following substances have you used? (circle all that apply)

None	Sedatives or Tranquilizers	Decongestants
Anti-histamines	Alcohol (3 drinks or more)	

Other (please explain) _____

4. For each of the following conditions, please indicate how you are feeling right now, on the scale of "none" through "severe". If you do not understand any of the terms, please consult the glossary at the bottom of this page or ask the experimenter.

1. General discomfort	***none***	***slight***	***moderate***	***severe***
2. Fatigue	***none***	***slight***	***moderate***	***severe***
3. Headache	***none***	***slight***	***moderate***	***severe***
4. Eye Strain	***none***	***slight***	***moderate***	***severe***
5. Difficulty Focusing	***none***	***slight***	***moderate***	***severe***
6. Increased Salivation	***none***	***slight***	***moderate***	***severe***
7. Sweating	***none***	***slight***	***moderate***	***severe***
8. Nausea	***none***	***slight***	***moderate***	***severe***
9. Difficulty Concentrating	***none***	***slight***	***moderate***	***severe***
10. Fullness of Head	***none***	***slight***	***moderate***	***severe***
11. Blurred Vision	***none***	***slight***	***moderate***	***severe***
12. Dizzy (with eyes open)	***none***	***slight***	***moderate***	***severe***
13. Dizzy (with eyes closed)	***none***	***slight***	***moderate***	***severe***
14. Vertigo	***none***	***slight***	***moderate***	***severe***
15. Stomach Awareness	***none***	***slight***	***moderate***	***severe***
16. Burping	***none***	***slight***	***moderate***	***severe***
17. Hunger	***none***	***slight***	***moderate***	***severe***

Explanation of Conditions

Fatigue:	*weariness or exhaustion of the body*
Eye Strain:	*weariness of soreness of the eyes*
Nausea:	*stomach distress*
Vertigo:	*surroundings seem to swirl*
Stomach Awareness:	*just a short feeling of nausea*

Scoring

For each question, a score of 0 (none), 1 (slight), 2 (moderate), or 3 (severe) is assigned The scores are then combined as follows [Kennedy93]. See Appendix B.5 for results.

Column 1 = Sum (1, 6, 7, 8, 9, 15, 16)

Column 2 = Sum (1, 2, 3, 4, 5, 9, 11)

Column 3 = Sum (5, 8, 10, 11, 12, 13, 14)

NAUSEA = Column 1 x 9.54

Oculomotor Discomfort = Column 2 x 7.58

Disorientation = Column 3 x 13.92

Total Severity = (Column 1 + Column 2 + Column 3) x 3.74

Appendix A.3 Guilford-Zimmerman Aptitude Survey – Part 5 Spatial Orientation

This test is a copyrighted work, which we used with permission from. Please contact them for reproductions of the test.

Consulting Psychologists Press, Inc.

3803 E. Bayshore Roard,

Palo Alto, CA 94303

Appendix A.4 Participant Experiment Record

Participant Experiment Record

User ID: _____ Date: _____

Real Space	Time A	Time B	Incorrect	Notes
Small Patterns				
Pattern #1 (ID:)				
Pattern #2 (ID:)				
Pattern #3 (ID:)				
Large Patterns				
Pattern #1 (ID:)				
Pattern #2 (ID:)				
Pattern #3 (ID:)				
Virtual Environment:				
Small Patterns				
Pattern #1 (ID:)				
Pattern #2 (ID:)				
Pattern #3 (ID:)				
Large Patterns				
Pattern #1 (ID:)				
Pattern #2 (ID:)				
Pattern #3 (ID:)				

Additional Notes:

Appendix A.5 Debriefing Form

Debriefing

Virtual environments are used to help bring people and computers together to explore problems from medicine to architecture, from entertainment to simulations. Researchers have made strong advances in rendering, tracking, and hardware. We look to explore an approach to two components that are not currently largely overlooked: (a) visually faithful user representations (avatars) and (b) natural interactions with the VE.

The purpose of this study is to test whether inserting real objects, such as the participant's arm and the blocks, into the virtual environment improves task performance compared to doing an "all virtual" environment (where everything is computer generated). The second purpose is to test whether having a visually faithful avatar (seeing an avatar that looks like you) improves a sense-of-presence over a generic avatar.

To test this hypothesis, we included 4 conditions, with the same block manipulation task in each: (a) On a real table, in an enclosure, without any computer equipment; (b) in an all virtual condition where the participant wore tracked gloves and manipulated virtual blocks; (c) in a hybrid environment where the user wore gloves to give a generic avatar but manipulated real blocks; (d) in a visually faithful hybrid environment where the participant saw their own arms and could naturally interact with the environment. Subjects did the real space and then one of the purely virtual, hybrid environment, or the visually faithful hybrid environment. From the findings, we hope to expand on the capabilities and effectiveness of virtual environments.

I would like to ask you to not inform anyone else about the purpose of this study. Thank you for participating. If you have questions about the final results, please contact Benjamin Lok (962-1893, lok@email.unc.edu), Dr. Fred Brooks (962-1931, brooks@cs.unc.edu).

If you are interested in finding out more about virtual environments, please read the following paper:

Brooks,Jr., F.P., 1999: "What's Real About Virtual Reality?" IEEE Computer Graphics and Applications,19, 6:16-27.

or visit:

http://www.cs.unc.edu/Research/eve

Are there any questions or comments?

References

Slater, M., & Usoh, M. (1994). Body Centred Interaction in Immersive Virtual Environments, in N. Thalmann and D. Thalmann (eds.) Artificial Life and Virtual Reality, John Wiley and Sons, 1994, 125-148.

Appendix A.6 Interview Form

VE Research Study: Debriefing Interview

Debrief by:_____ Date:_____

Questions	Comments
How do you feel? – sickness – nausea	
What did you think about your experience?	
What percentage of the time you were in the lab did you feel you were in the virtual environment? ? >50% or <50% of the time?	
Any comments on the environment? - what made it real - what brought you out - what objects did you see	
Any comments on your virtual body? - Behavior - identified with it	
Any comments on interacting with the environment? - manipulating the blocks? - Was it difficult? - Was it natural?	
How long did it take for you to get use to the virtual environment? - grabbing and moving objects - the "rules" of the system	
What factors do you think: - helped you complete the task - hindered your completing the task	
	Any additional comments:

Appendix A.7 Kennedy-Lane Simulator Sickness Post-Experience Questionnaire

Participant Health Assessment

(To be completed after the experiment.)

For each of the following conditions, please indicate how you are feeling right now, on the scale of "none" through "severe." Circle your response.

1. General Discomfort	None	Slight	Moderate	Severe
2. Fatigue	None	Slight	Moderate	Severe
3. Headache	None	Slight	Moderate	Severe
4. Eye Strain	None	Slight	Moderate	Severe
5. Difficulty Focusing	None	Slight	Moderate	Severe
6. Increased Salivation	None	Slight	Moderate	Severe
7. Sweating	None	Slight	Moderate	Severe
8. Nausea	None	Slight	Moderate	Severe
9. Difficulty Concentrating	None	Slight	Moderate	Severe
10. Fullness of Head	None	Slight	Moderate	Severe
11. Blurred Vision	None	Slight	Moderate	Severe
12. Dizzy (with your eyes open)	None	Slight	Moderate	Severe
13. Dizzy (with your eyes closed)	None	Slight	Moderate	Severe
14. Vertigo	None	Slight	Moderate	Severe
15. Stomach Awareness	None	Slight	Moderate	Severe
16. Burping	None	Slight	Moderate	Severe
17. Hunger	None	Slight	Moderate	Severe

In the space below, please list any additional symptoms you are experiencing (continue on the back if necessary).

Appendix A.8 Steed-Usoh-Slater Presence Questionnaire

S.U.S. Questionnaire

I. Personal Info

Gender:	**Please tick against your answer**
1. Male	1
2. Female	2

My status is as follows:	**Please tick against your answer**
1. undergraduate student	1
2. Masters student	2
3. PhD student	3
4. Research Assistant/Research Fellow	4
5. Staff member - systems/technical staff	5
6. Faculty	6
7. Administrative staff	7
8. Other (please write in)...	8

1. Have you experienced "virtual reality" before?

I have experienced virtual reality…	**Please tick against your answer**
1. never before	1
2.	2
3.	3
4.	4
5.	5
6.	6
7. a great deal	7

2. To what extent do you use a computer in your daily activities?

I use a computer...	**Please tick against your answer**
1. not at all	1
2.	2
3.	3
4.	4
5.	5
6.	6
7. very much so	7

3. When you played PC/video games the most (the past few years), how much did you play?

I play or played computer or video games …	**Please tick against your answer**
1. never	1
2. less then 1 hour per week	2
3. between 1 and 5 hours per week	3
4. between 5 and 10 hours per week	4
5. more then 10 hours per week	5

II. The following questions relate to your experience

1. How dizzy, sick or nauseous did you feel resulting from the experience, if at all? Please answer on the following 1 to 7 scale.

I felt sick or dizzy or nauseous during or as a result of the experience...	**Please tick against your answer**
1. not at all	1
2.	2
3.	3
4.	4
5.	5
6.	6
7. very much so	7

2. Please rate *your sense of being in the* virtual room with the blocks, on the following scale from 1 to 7, where 7 represents your *normal experience of being in a place*.

I had a sense of "being there" in the virtual room…	**Please tick against your answer**
1. not at all	1
2.	2
3.	3
4.	4
5.	5
6.	6
7. very much	7

3. To what extent were there times during the experience when the virtual room was reality for you?

There were times during the experience when the virtual room was the reality for me...	**Please tick against your answer**
1. at no time	1
2.	2
3.	3
4.	4
5.	5
6.	6
7. almost all of the time	7

4. When you think back about your experience, do you think of the virtual room more as *images that you saw*, or more as *somewhere that you visited*?

The virtual room seems to me to be more like...	**Please tick against your answer**
1. images that I saw	1
2.	2
3.	3
4.	4
5.	5
6.	6
7. somewhere that I visited	7

5. During the time of the experience, which was the strongest on the whole, your sense of being in the virtual room, or of being in the physical laboratory?

I had a stronger sense of...	**Please tick against your answer**
1. being in the lab	1
2.	2
3.	3
4.	4
5.	5
6.	6
7. being in the virtual room	7

6. Consider your memory of being in the virtual room. How similar in terms of the *structure of the memory* is this to the structure of the memory of other *places* you have been today? By 'structure of the memory' consider things like the extent to which you have a visual memory of the virtual room, whether that memory is in color, the extent to which the memory seems vivid or realistic, its size, location in your imagination, the extent to which it is panoramic in your imagination, and other such *structural* elements.

I think of the virtual room as a place in a way similar to other places that I've been today...	**Please tick against your answer**
1. not at all	1
2.	2
3.	3
4.	4
5.	5
6.	6
7. very much so	7

7. During the time of the experience, did you often think to yourself that you were actually in the virtual room?

During the experience I often thought that I was really standing in the virtual room...	**Please tick against your answer**
1. not very often	1
2.	2
3.	3
4.	4
5.	5
6.	6
7. very often	7

8. How much did you associate with the visual representation of yourself (your avatar)?

During the experience I associated with my avatar...	**Please tick against your answer**
1. not very much	1
2.	2
3.	3
4.	4
5.	5
6.	6
7. very much	7

9. How realistic (visually, kinesthetically, interactivity) was the visual representation of yourself (your avatar)?

During the experience I thought the avatar was…	**Please tick against your answer**
1. not very realistic	1
2.	2
3.	3
4.	4
5.	5
6.	6
7. very realistic	7

10. Overall, how well do you think that you achieved your task?

I achieved my task...	**Please tick against your answer**
1. not very well at all	1
2.	2
3.	3
4.	4
5.	5
6.	6
7. very well	7

11. Further Comments

Please write down any further comments that you wish to make about your experience. In particular, what things helped to give you a sense of 'really being' in the virtual room, and what things acted to 'pull you out' of this?

Reminder - all answers will be treated entirely confidentially.

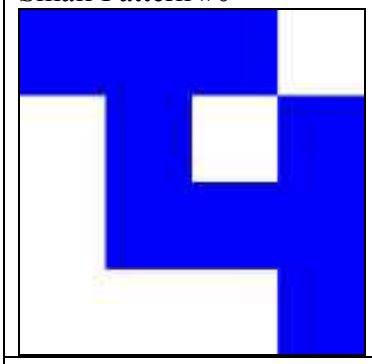
Thank you once again for participating in this study, and helping with our research. Please do not discuss this with anyone for two weeks. This is because the study is continuing, and you may happen to speak to someone who may be taking part.

Scoring

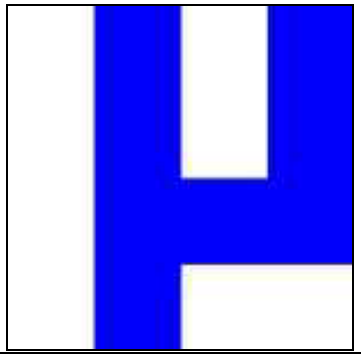
The UCL Presence Questionnaire is scored by counting the number of "high" scores, in our case, five, six and seven responses. See Table Appendix B.3 for results.

Appendix A.9 Patterns

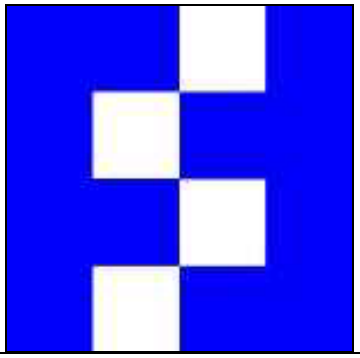
Small Pattern #0



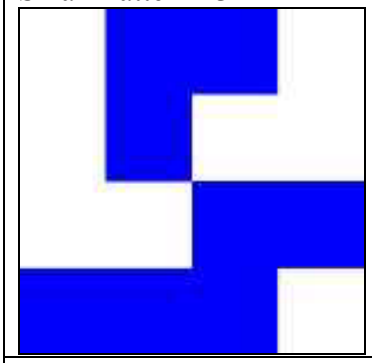
Small Pattern #1



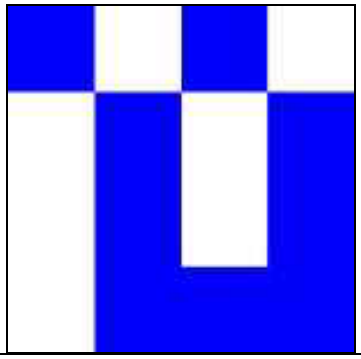
Small Pattern #2



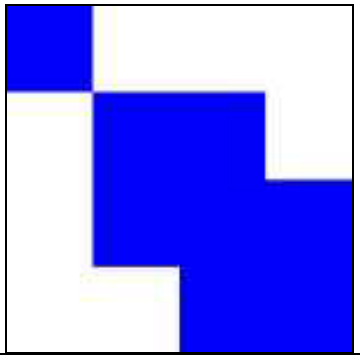
Small Pattern #3



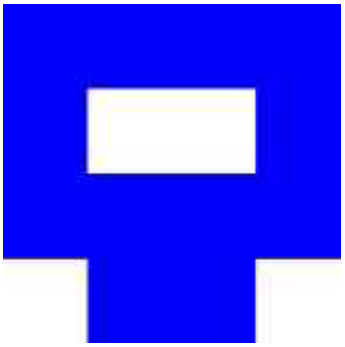
Small Pattern #4



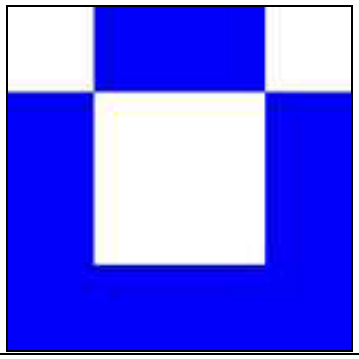
Small Pattern #5



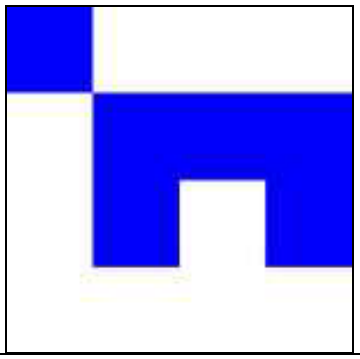
Small Pattern #6



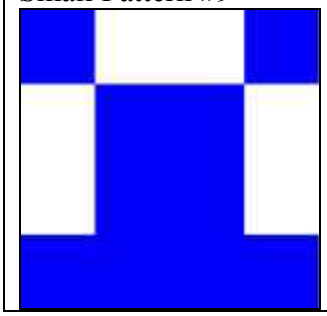
Small Pattern #7



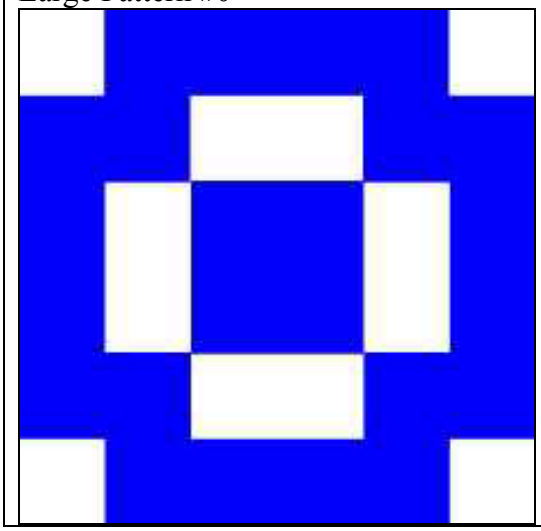
Small Pattern #8



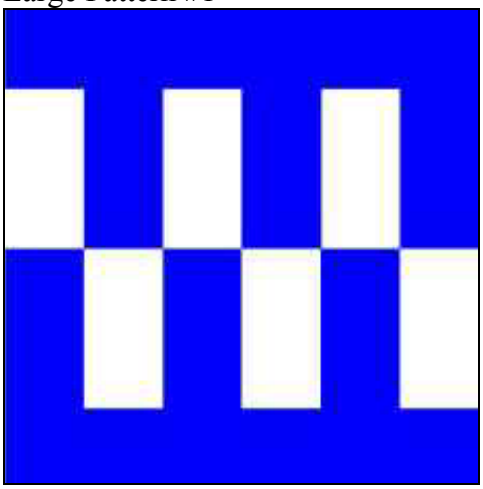
Small Pattern #9



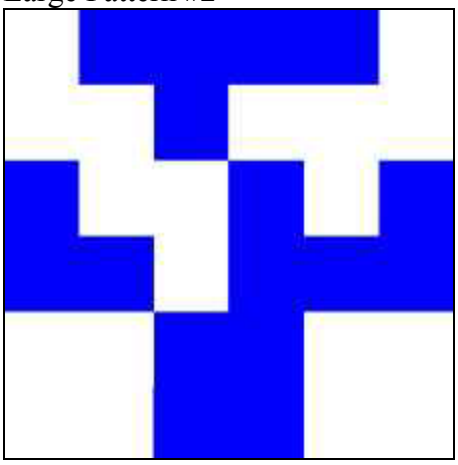
Large Pattern #0



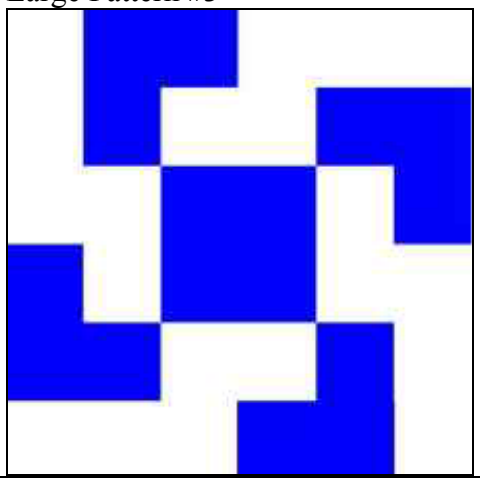
Large Pattern #1



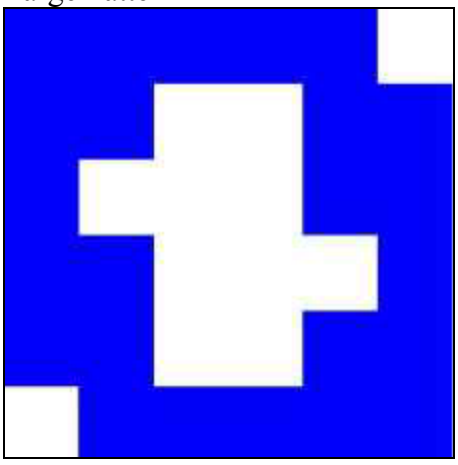
Large Pattern #2



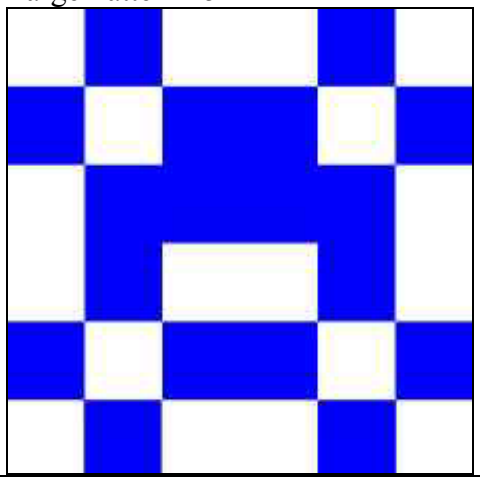
Large Pattern #3



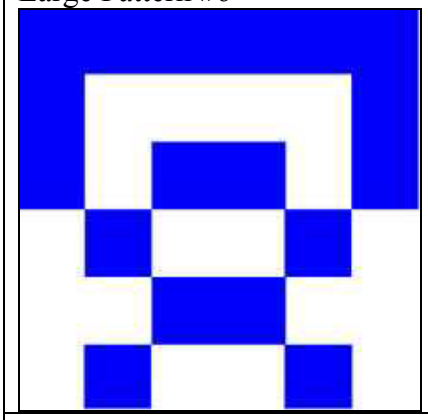
Large Pattern #4



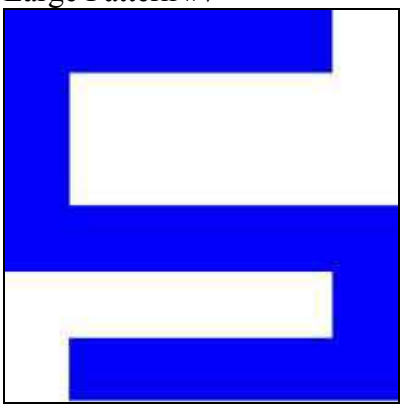
Large Pattern #5



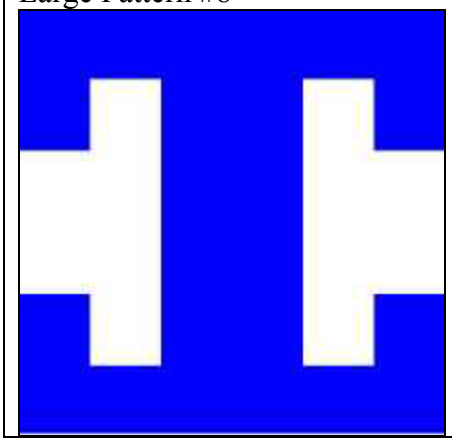
Large Pattern #6



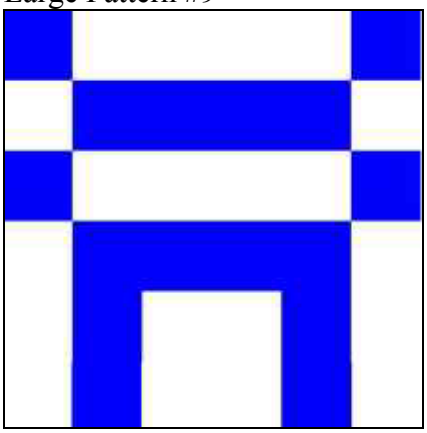
Large Pattern #7



Large Pattern #8



Large Pattern #9



Appendix B User Study Data

Participants #1-14 – PVE, #15-30 – VFHE, #31-44 HE.
Participant #12 got nauseated during PVE. The system broke down for Participants #29 & 30.
There was a data collection error for Participant#37.

Appendix B.1 Participant Data

Gender: 0. female, 1. male
Status: 1. Ugrad, 2. Masters, 3. PhD, 4. Rsch Asst/Fllw, 5. Staff, 6. Faculty, 7. Admin, 8. Other
VR Experience: 1. Never before… 7. A great deal
Computer use: 1. Not at all… 7. Very much so
Computer game play: I play or played computer or video games (per week):
1. Never, 2. < 1 hour, 3. >1 and <5 hours, 4. >5 and <10 hours, 5. >10 hours

ID #	Gender	Status	VR experience	Computer use	Computer game
1	1	2	1	7	2
2	1	1	1	7	4
3	1	1	2	7	3
4	0	1	1	7	2
5	1	1	3	7	2
6	0	1	1	7	2
7	1	1	1	7	4
8	1				
9	1	1	1	7	5
10	1	1	1	7	3
11	1	8	1	5	2
12	1	1	1	7	5
13	1	5	1	7	2
14	1	1	1	7	4
15	1	1	1	7	5
16	1	1	1	7	3
17	1	1	1	7	2
18	1	1	1	7	5
19	1	1	2	5	1
20	1	1	2	5	2
21	1	1	2	7	3
22	1	1	1	7	5
23	0	2	1	3	1
24	0	1	1	7	3
25	1	1	2	7	4
26	1	1	1	5	3
27	1	1	1	3	1
28	0	8	2	7	3
29	1				
30	0				
31	1	1	4	7	3
32	1	3	1	7	2
33	0	2	1	4	1
34	1	3	1	7	2
35	1	3	2	7	3

	36	1	1	1	7	4
	37	1	1	1	5	2
	38	1	1	1	7	3
	39	1	1	2	6	3
	40	1	1	2	5	3
	41	0	1	2	6	1
	42	1	1	1	7	3
	43	1	1	1	7	5
	44	0	1	1	7	1

Appendix B.2 Task Performance

ID# 1-14 = PVE, 15-30 = VFHE, 31-44 = HE

ID #	RSE Small Average	RSE Large Average	VE Small Average	VE Large Average	VE – RSE Small	VE – RSE Large	Ratio Small	Ratio Large	RS Total Incorrect	VE Total Incorrect	Total Incorrect
1	20.17	50.13	46.10	127.00	25.93	76.87	2.29	2.53	0	0	0
2	17.53	34.97	49.50	123.20	31.97	88.23	2.82	3.52	0	0	0
3	11.37	43.80	39.30	103.65	27.93	59.85	3.46	2.37	0	0	0
4	14.07	42.17	45.20	126.20	31.13	84.03	3.21	2.99	0	0	0
5	15.80	32.33	41.20	114.10	25.40	81.77	2.61	3.53	0	0	0
6	16.30	37.43	43.10	92.60	26.80	55.17	2.64	2.47	0	0	0
7	25.13	57.13	42.00	117.25	16.87	60.12	1.67	2.05	0	0	0
8	21.10	35.93	63.50	135.20	42.40	99.27	3.01	3.76	0	0	0
9	12.73	24.87	46.15	70.20	33.42	45.33	3.62	2.82	0	0	0
10	47.37	45.30	47.10	148.55	-0.27	103.25	0.99	3.28	0	0	0
11	10.90	25.70	43.55	97.65	32.65	71.95	4.00	3.80	0	0	0
12	16.47	48.17									
13	15.77	44.77	73.55	192.20	57.78	147.43	4.66	4.29	1	1	2
14	18.17	31.53	33.85	73.10	15.68	41.57	1.86	2.32	1	1	2
15	14.50	38.43	25.00	68.50	10.50	30.07	1.72	1.78	0	1	1
16	21.50	52.23	24.90	100.80	3.40	48.57	1.16	1.93	0	0	0
17	12.40	38.00	46.00	76.05	33.60	38.05	3.71	2.00	1	1	2
18	16.37	34.03	39.90	89.10	23.53	55.07	2.44	2.62	0	0	0
19	12.60	31.70	26.35	58.50	13.75	26.80	2.09	1.85	1	0	1
20	8.77	23.90	20.20	56.35	11.43	32.45	2.30	2.36	1	0	1
21	20.67	38.10	27.00	68.75	6.33	30.65	1.31	1.80	0	0	0
22	18.10	50.00	35.30	77.45	17.20	27.45	1.95	1.55	2	0	2
23	17.03	29.37	26.50	74.75	9.47	45.38	1.56	2.55	1	0	1
24	14.87	36.13	34.60	62.35	19.73	26.22	2.33	1.73	0	0	0
25	17.77	29.35	22.75	54.60	4.98	25.25	1.28	1.86	1	1	2
26	11.60	30.57	20.45	104.50	8.85	73.93	1.76	3.42	0	1	1
27	20.57	57.20	31.90	51.60	11.33	-5.60	1.55	0.90	3	1	4
28	13.53	30.63	23.40	69.10	9.87	38.47	1.73	2.26	0	0	0
29	21.40	42.87							1	0	1
30	17.60	25.17							0	0	0
31	13.47	28.20	34.10	62.25	20.63	34.05	2.53	2.21	0	1	1
32	13.83	33.53	28.90	109.30	15.07	75.77	2.09	3.26	0	0	0
33	14.07	41.60	35.95	93.65	21.88	52.05	2.56	2.25	0	1	1
34	12.97	40.37	38.50	86.20	25.53	45.83	2.97	2.14	1	0	1
35	12.57	31.20	23.90	61.15	11.33	29.95	1.90	1.96	0	1	1
36	9.97	25.90	29.00	100.40	19.03	74.50	2.91	3.88	1	0	1
37	13.50	46.77							0	0	0
38	11.73	24.17	20.20	56.65	8.47	32.48	1.72	2.34	0	0	0
39	14.80	32.93	29.85	82.50	15.05	49.57	2.02	2.51	0	1	1
40	15.63	40.33	31.50	93.40	15.87	53.07	2.01	2.32	0	0	0
41	12.77	48.13	39.25	99.95	26.48	51.82	3.07	2.08	0	0	0
42	21.80	25.27	34.25	62.90	12.45	37.63	1.57	2.49	0	0	0
43	20.13	40.90	29.10	66.55	8.97	25.65	1.45	1.63	0	0	0
44	30.13	37.23	37.30	153.85	7.17	116.62	1.24	4.13	0	0	0

Appendix B.3 SUS Sense-of-presence

Q1. I felt sick or dizzy or nauseous during or as a result of the experience (1. Not at all… 7. Very Much So)
Q2. I had a sense of "being there" in the brick room (1. Not at all… 7. Very much)
Q3. There were times during the experience when the brick room was the reality for me (1. At no time… 7. Almost all of the time)
Q4. The brick room seems to me to be more like (1. Images that I saw… 7. Somewhere that I visited)
Q5. I had a stronger sense of (1. Being in the lab… 7. Being in the brick room)
Q6. I think of the brick room as a place in a way similar to other places that I've been today (1. Not at all… 7. Very much so)
Q7. During the experience I often thought that I was really standing in the brick room (1. Not very often… 7. Very often)
Q8. During the experience I associated with my avatar (1. Not very much… 7. Very much)
Q9. During the experience I though the avatar was (1. Not very realistic… 7. Very Realistic)
Q10. I Achieved my task (1. Not very well at all… 7. Very well)
SUS: Score 1 for each response ≥5, from Q2-Q7.

ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	**SUS**	Comments
1	2	5	6	4	5	4	4	5	3	4	3	Colors and space were realistic. The hand movement and interference brought me out
2	2	2	1	1	1	1	1	1	1	4	0	What brick room? (Didn't answer any of the other questions because the didn't know the brick room = VE), ammended: I never even noticed that it was supposed to be a brick room! I focused completely on the view I first saw, that of the blocks, and never looked around. The way those goggles are set up made it nearly impossible to have a sense of 'really being' in the brick room. The images were coming from two small squares that were far enough from my eyes to leave much of the structure of the goggles, as well as my arms, feet, and the floor below me clearly within my field of view. So long as I received those constant visual cues telling me that I was not within a brick room, it was impossible to experience immersion.
3	3	4	5	6	5	3	2	5	3	4	3	My avatar helped to give me a sense of really being in the brick room. I think that if I had some time to walk around (or just look around0 the room, I would have felt more like I was actually there.
4	1	6	7	7	7	7	7	6	7	5	6	I really felt like I was in the brick room. The only thing that reminded me that I wasn't was the weight on my head, and not being comfortable moving the blocks. I somewhat had a difficult time manuevering the blocks. Visually, I just thought I was wearing weird glasses in the brick room.
5	1	4	3	4	4	5	3	3	3	5	1	If the environment froze, that obviously took me out of it. My hands acted and responded very well. Seemed lifelike for the most part.
6	1	7	7	6	7	6	7	6	6	5	6	Everything moved well; when I moved my hand, my virtual hand performed the same action. The headgear pulled me out of it along with the inability to move the blocks with two hands
7	3	5	5	3	5	3	5	4	4	6	4	Things that helped: hands, being surrounded by walls, things that hurt: headmount gre heavy, there is a little delay when you move, the fingers on hand didn't move,

												outside people talking
8	6	4	2	1	1	6	1	2	2	3	1	The spatial representation of items in the room was very good (the lamp, the mona lisa, the tables). This increased my sense of 'really being' in the room. The blocks and hands were not quite so accurate so they seemed 'less real' to me. I was amazed at how quickly I was affected by physical symptoms (sweating/nausea) as a result of the VR.
9	1	4	3	2	2	4	2	6	3	6	0	The only thing that really gave me a sense of really being in the brick room was the fact that the hands moved when mine moved, and if I moved my hand, the room changed to represent that movement. Things htat pulled me out were that the blocks floated, but this did help in acutally solving the puzzles easier.
10	1	5	4	2	5	7	3	5	5	5	3	When my hands were resting within the blocks, I could see what looked like a blue flame eminating from where my hands were sticking out of blocks
11	1	5	5	6	4	3	3	5	3	5	3	Seeing my hands was helpful, but feeling the actual blocks would have helped a lot more. The image was sometimes somewhat sturdy.
12	4	5	5	6	6	5	5	6	4	4	6	The motion and random virtual objects made the room real. The slight motion sickness began to make me think I was in a nazi torture chamber. The room seemed very real, perhaps if I had explore the room more it would have seemed even better.
13	1	5	5	3	6	4	3	3	3	3	3	Had the headset not place as much strain as it did on my neck, I might have done better. Something about too much perfection is distracting in the case of the environment.
14	1	6	5	5	5	6	5	5	4	5	6	Movement of hands and ability to interact w/ blocks and move them around helped
15	1	6	4	4	5	3	5	6	7	7	3	Being there -> mona lisa, lamp with appropriate lighting, sign on wall, vodeo of own hands, and actual blocks on table. Pulled out -> video 'noise' around my hands/blocks if it was clean around the images of my hands I would be totally immersed.
16	1	5	4	7	5	5	5	5	5	5	5	The fact that things moved when I moved helped me believe that I was really there. It was hard to seeattimes
17	1	5	3	2	4	6	2	6	5	5	2	It would have been almost completely believable if there wasn't the little bit of noise and discoloration when I saw myself. When I looked around the room it was nearly flawless
18	1	6	7	7	7	6	7	4	4	6	6	The plant was a great touch to the brick room. The total immerson of things that I knew that were truly not in the lab helped to make me forget that I was in the lab. I also suspended disbelief to help me accomplish my task. This in itself allowed for full immersion in the rooml. Factors such as lag and noise kept this from being a true/realistic environment, but it was very close.
19	2	4	5	5	5	4	4	4	4	6	3	The objects around the room helped as well as the relationship between my moving physical objects and seeing it in the room. I was however, aware of my "dual" existance in two rooms
20	2	3	4	3	5	6	2	6	4	5	2	I felt most like I was in the room when I was engaged in activities within the room (doing the puzzles). I felt least

												that I was in the room because I couldn't touch walls, paintings, etc. Also, I had on a heavy head piece and could see out of it peripherally into the real world. Also, I felt most like I was in the room when I could see myself (my hands)
21	1	5	3	2	3	7	2	4	2	5	2	The things that helped me to 'be there' in the brick room were the things that were different from the lab. This includes the painting, the plant, the pattern and difference in the wall surface. My avatar 'pulled me out' because it didn't integrate well with the virtual world. Updating the hands took too long, relative to updating the rest of the world, making them stay separate making me realize I wasn't actually in the brick room.
22	2	2	1	1	2	1	2	2	2	6	0	The pixel artifacts, the limited field of vision, and the delay between moving an object all acted to hinder immersion. The room itself, w/the picture and lamp and such were very well done, and enhanced immersion.
23	1	6	5	3	7	1	4	7	5	5	3	I didn't know what the black image on the grid on the table was, it interrupted my vision of the blocks.
24	1	4	2	2	2	3	3	2	2	7	0	In terms of the brick room, the environment felt very real. However, the fact I could not see my feet and the fact that there were wires, etc. on the floor which did not appear on the virtual floor, 'pulled me out' since I knew things were there but couldn't really see them in the virtual environment. In terms of the blocks, I never really felt a sense that they were real in the virtual environment because I could not see them very clearly, and when I rotated them, there were many delays in the virtual environment. I know I was not seeing what I was actually doing (accurately). Overall I thought the room was very well done, but the blocks portion of the VE needs some improvement. Good luck with the study! I had fun :)
25	1	3	1	2	5	4	1	5	5	3	1	The visual shearing pulled me out of it, the hands put me back in, the inability of being able to readjust the angle of my head to facilitate comfort pulled me out again.
26	1	5	7	5	5	4	5	7	7	6	5	My hands looked very realistic. The only thing that really took away from the sense of reality was the fuzziness around the blocks and the limited peripheral vision
27	2	4	4	2	3	4	4	4	4	4	0	Really being: movement (mobility), spatial relationships, Pulled you out: knowing there were physical objects present in the room before putting headset on. Would have been more realistic if instructor had instructed me to focus on specific things more before the task
28	3	4	6	3	4	4	1	3	7	6	1	Pulled out: Small visual area, fuzziness surrounding visual area, weight of helmet, feeling the wires.Pulled in: feeling the blocks, seeing textures on blocks and hands
29												
30												
31	1	4	3	2	2	4	3	5	4	5	0	The fact that there were objects in the room that looked realistic helped fulfill the sense of being in the room. If the refresh rate was high or the delay and jumpiness ewhen moving was lowered so that that movement was

												smooth, that would be great. When I was able to think clearly as opposed to focusing on the block task I was drawn away
32	2	5	6	5	6	5	5	6	6	6	6	Seeing my hands at work, distortion in the images; delay in the transmission of images->cannot rotate the blocks as fast as I want to. 2-3 times I missed a rotation and had to later go back and change the blocks again. The virtual image of the pattern also reminds one of being in a virtual rm. But that is partly due to unrealism of the img
33	1	2	2	1	1	2	1	3	2	7	0	Pulled me out: the time dleay between what I was doing with my hands and what I saw, the image of the blocks was not as sharp aand live as the picture of the room with the mona lisa (that felt really real)
34	1	6	6	7	7	7	5	6	7	7	6	Really being there: being albe to move around 360 and see everything in the room, pulled out: voice commands, upon completing tasks
35	2	4	2	4	4	5	1	5	5	4	1	Picture on the wall, boxes on the table
36	2	3	2	1	1	2	1	3	2	2	0	When I looked around for a couple seconds it felt like I was in the brick room, but then when I looke down at my body, stuck out my hand and coulnt' see anything I felt more in the lab room, just kind of watching tv or something. Also, when I was doing the blocks, the noise on the screen made it seem a lot less realistic
37	1	6	6	3	5	2	3	4	2	6	3	Putting me in: Full field of vision at first, not seeing lab room, pulled me out: color variationb etween room and objects/hands while doing task. Different reaction time from reality something to get used to
38	1	5	6	2	6	4	3	6	4	5	3	Having the image of my hands and the blocks spliced into the brick room reality drew me away from the brick room and made me feel more in the lab. If my hands and the blocks had been rendered in the brick room graphics, then I might have felt even more a part of the reality.
39	2	4	4	5	4	3	3	6	6	3	1	Bad: blocks fuzzy and even unviewable on edges, color distortion, good: crisp representation of patterns and static objects, also hands seemed good
40	1	4	3	4	3	5	3	4	4	6	1	The glitches in the visuals were the only thing pulling me out of the experience. Everything else seemd fairly realistic. Movement of my hand would cause glitches, which removes me from connecting with my avatar
41	1	4	4	5	5	6	5	6	5	6	4	Helped: Looking around room, moving my arms and objects up and down rather than just left-right, hindered: glitches, feeling the headset move or touching the table, etc and remember where I was
42	2	4	3	1	4	2	2	5	5	4	0	The accurately represented movement helped, but the lack of peripheral vision, noise and choppiness/frame rate pulled me out
43	3	4	3	3	4	3	3	2	5	5	0	The objects in the room and knowing that if I turned to the side that they would be there helped. But the static distorted vision pulled me back into the laboratory.
44	5	3	1	3	3	5	2	6	7	4	1	Being able to see my hands moving around helped with the sense of "being there". The fuzziness of the blocks and lack of ability to glance up at the block pattern w/ my eyes only pulled me out. Having to move my entire head to look was unnatural

Appendix B.4 Debriefing Trends

	PVE *n* = 13	HE *n* = 13	VFHE *n* = 14	Total *n* = 40
1. How do you feel				
R1. Fine	8	9	9	26
R2. Neck/Back is sore	5	2	6	13
R3. Dizzy/Nausea	4	2	3	9
R4. Headache	1	1	0	2
R5. Eyes are tired	0	0	2	2
2. What did you think about your experience?				
R1. Fun	4	6	6	16
R2. Interesting	8	5	6	19
R3. Frustrating	3	0	0	3
R4. New experience	3	2	0	5
R5. Surprised at difficulty	1	1	0	2
R6. Weird	0	0	2	2
R7. Unimpressed	0	1	0	1
3. What percentage of the time you were in the lab did you feel you were in the virtual environment?	67.3	61.4	70.0	66.3
R1. Noticed tracking failed	1	0	0	1
R2. Very focused on task (100%)	0	2	4	6
4. Any comments on the environment that made it feel real				
R1. When head turned, so did everything else (made real)	4	3	2	9
R2. Took up entire FOV (made real)	1	1	1	3
R3. Virtual Objects (mona lisa, plant, etc) (made real)	5	8	6	19
R4. Seeing Avatar (made real)	3	4	4	11
R5. Concentrating on a task	3	1	3	7
R6. Tactile feedback	0	1	4	5
R7. Virtual objects looked like real objects	0	0	1	1
R8. Real objects	0	2	0	2
R9. Goal pattern was easy to see	0	1	0	1
4B. What brought you out				
R1. Tracker failing (brought out)	2	0	0	2
R2. Sounds (talking/lab) (brought out)	4	1	2	7
R3. Seeing under shroud (brought out)	1	3	2	6
R4. Floating blocks/snapping (PV)	4	0	0	4
R5. Headmount (weight/fitting)	1	2	3	6
R6. Blocks didn't really exist (PV)	1	0	0	1
R7. Hand could pass through blocks (PV)	1	0	0	1
R8. Environment looked computer generated	2	1	0	3
R9. Reconstruction noise (HE/VFHE)	0	11	10	21
R10. Couldn't touch virtual objects	0	0	1	1
R11. Blocks looked fake	0	0	1	1
R12. Presence of physical objects (blocks/table)	0	2	1	3
R13. Wires	0	0	1	1
R14. Lag	0	2	0	2
R15. Reconstruction rate	0	1	0	1
R16. Lack of peripheral vision	0	1	0	1

R17. Working on a task	0	1	0	1
5. Any comments on your virtual body				
R1. Fine	9	11	9	29
R2. Movement Matched	2	2	1	5
R3. Noticed arm detached from hand	1	0	0	1
R4. Mismatch of model <-> reality. Different Hand positions/Fingers didn't respond/Fingernails	5	1	2	8
R5. No Tactile Feedback	3	0	0	3
R6. Shadows were weird	1	0	0	1
R7. Looked Real	0	4	9	13
R8. Lag	0	4	6	10
R9. Noisy Images	0	2	5	7
R10. Color was a bit off	0	1	1	2
R11. Didn't notice hands	0	2	0	2
R12. Looked like video	0	2	0	2
6. Any comments on interacting with the environment				
R1. Took more thinking	2	0	0	2
R2. Rotation took a larger arc than usual	8	0	0	8
R3. Frustrating	5	1	0	6
R4. Learned to use whole hand instead of fingers	1	0	0	1
R5. Had trouble using two hands	4	1	0	5
R6. Lag made things harder	0	6	6	12
R7. Used sense of feel to assist vision	0	1	2	3
R8. Low FOV hurt grabbing	0	5	9	14
R9. Interaction was natural	0	4	3	7
R10. Interaction was hard (hard to see/pick up blocks)	0	2	1	3
7. How long did it take for you to get used to the VE?	2.4	2.0	1.5	2.0
8A. What factors helped you complete your task				
R1. Blocks in mid-air (PV)	8	0	0	8
R2. Two handed interaction (PV)	1	0	0	1
R3. Seeing an avatar	2	2	1	5
R4. Block snapping (PV)	2	0	0	2
R5. Gridding the pattern	0	3	1	4
R6. Practice in Real space	0	3	3	6
R7. Location of sample pattern	0	0	2	2
R8. Playing plenty of video games	0	0	1	1
8B. What factors hindered your completing your task				
R1. Not having complete hand control	1	0	0	1
R2. Not being able to feel	1	0	0	1
R3. Highlights were hard to see	2	0	0	2
R4. Blocks didn't go where they thought they would/snapping	6	0	0	6
R5. Hard to see pattern (in blocks)	1	1	1	3
R6. View registration	1	2	0	3
R7. Headset was heavy	1	0	1	2
R8. Display Errors	0	2	3	5
R9. Couldn't see pattern + blocks all in one view	0	5	3	8
R10. Poor headset fit/focus settings	0	1	0	1
R11. Had trouble distinguishing between blue and white faces	0	1	0	1
Since block manipulation was slower, had to learn relationship between sides as opposed to real space where it was so fast to spin the blocks, they didn't have to.	2	1	0	3

Appendix B.5 Simulator Sickness

ID #	Total	Total	Difference
1	0	0	0
2	1	1	0
3	2	5	3
4	0	0	0
5	2	1	-1
6	0	0	0
7	1	6	5
8	1	13	12
9	1	1	0
10	2	3	1
11	0	1	1
12	2	8	6
13	3	6	3
14	1	1	0
15	0	0	0
16	0	0	0
17	0	1	1
18	1	2	1
19	0	2	2
20	3	3	0
21	4	3	-1
22	2	2	0
23	1	1	0
24	0	0	0
25	5	5	0
26	1	2	1
27	1	6	5
28	2	6	4
29	1		
30	0		
31	2	2	0
32	1	4	3
33	4	0	-4
34	5	9	4
35	0	7	7
36	5	6	1
37	3	0	-3
38	5	3	-2
39	1	4	3
40	3	4	1
41	4	2	-2
42	2	8	6
43	3	7	4
44	0	3	3

Appendix B.6 Spatial Ability

ID #	Highest question attempted	Skipped	Wrong	Right	Final Score	Percentage
1	41	0	7	26	24.25	78.79
2	41	0	9	24	21.75	72.73
3	43	0	3	32	31.25	91.43
4	30	0	4	18	17	81.82
5	22	0	0	14	14	100.00
6	30	0	4	18	17	81.82
7	25	1	2	14	13.5	87.50
8	39	0	4	27	26	87.10
9	42	0	4	30	29	88.24
10	30	1	16	5	1	23.81
11	26	0	2	16	15.5	88.89
12	48	0	18	22	17.5	55.00
13	33	0	8	17	15	68.00
14	32	0	1	23	22.75	95.83
15	58	0	1	49	48.75	98.00
16	58	0	20	30	25	60.00
17	36	1	5	22	20.75	81.48
18	34	0	10	16	13.5	61.54
19	43	0	4	31	30	88.57
20	67	0	6	53	51.5	89.83
21	52	0	8	36	34	81.82
22	39	1	12	18	15	60.00
23	25	0	2	15	14.5	88.24
24	25	0	1	16	15.75	94.12
25	28	0	6	14	12.5	70.00
26	39	1	2	28	27.5	93.33
27	29	0	4	17	16	80.95
28	44	0	6	30	28.5	83.33
29						
30						
31	36	0	10	18	15.5	64.29
32	27	2	3	14	13.25	82.35
33	43	3	3	29	28.25	90.63
34	41	0	11	22	19.25	66.67
35	54	0	6	40	38.5	86.96
36	35	0	10	17	14.5	62.96
37	28	0	3	17	16.25	85.00
38	54	0	4	42	41	91.30
39	50	0	12	30	27	71.43
40	38	0	6	24	22.5	80.00
41	29	0	6	15	13.5	71.43
42	50	0	6	36	34.5	85.71
43	53	0	1	44	43.75	97.78
44	21	0	2	11	10.5	84.62

Appendix C NASA Case Study Surveys

Appendix C.1 Pre-Experience Survey

Pre Experience Survey

Brief description of your role in payload development:

What payload development tasks do you potentially see VR technologies aiding?

What are general types of tasks, such as attaching connectors and screwing fixtures, are common to payload assembly?

Specific to the task I just explained:

How much space between the **TOP** of the PMT and the **BOTTOM** of the second payload is necessary? ____ CM

How much space would you actually allocate? ____ CM

Appendix C.2 Post-Experience Survey

Post Experience Survey

Specific to the task you just experienced:

After your experience, how much space do you feel was necessary between the **TOP** of the PMT and the **BOTTOM** of the second payload is necessary? ____ CM

How much space would you actually allocate? ____ CM

How much time would such a spacing error cost if discovered during the final payload layout?

How much money would such a spacing error cost if discovered during the final payload layout?

After your experience, what ***additional*** payload development tasks do you potentially see VR technologies aiding?

Please write down some issues or problems you currently have with a specific payload development tasks and what tool, hardware, or software would assist you?

Appendix C.3 Results

Pre-Experience Survey:

1) Brief description of your role in payload development:

1: I have worked in both flight software and hardware development. Work as an electronics engineer involves decisions about connector placement, cable routing, hardware placement, etc.
2: Integration and test management. Design and implement testing of payload before and after satellite integration
3: I design ground system software, plan mission ops scenarios, and write system test and mission commanding/monitoring software
4: System design & flight payloads. Primary Instrument in PC Board Design & Fabrication

2) What payload development tasks do you potentially see VR technologies aiding?

1: Tasks I mentioned above: connector placement (sufficient access for example), where to place cables throughout the payload, how to orient subsystem boxes.
2: Container design; ergonomic training for cable layout and connector fitting; training for mechanical adjustments of payload.
3: Use it in the payload design/planning stage to determine if payload components will fit within spacecraft constraints.
4: Form Fit Factors. Multiple Player design & development (Private + Government).

3) What are general types of tasks, such as attaching connectors and screwing fixtures, are common to payload assembly?

1: Cable routing, cable moounting/demounting (see above).
2: Cable layout; mechanism adjustments; GSE fit and location; layout of hardware (both flight & GSE) in environmental testing (Thermal/Vac Chamber, etc.).
3: Attaching to predefined spacecraft connectors, mounting hardware, etc. Fitting with spacecraft enclosure space constraints.
4: Connector, cable assembly. Instrumentation installation in shuttle environment.

Specific to the task I just explained:
How much space between the **TOP** of the PMT and the **BOTTOM** of the second payload is necessary?
____ CM

1: 14 cm
2: 14.2 cm
3: 15-16 cm
4: 15 cm

How much space would you actually allocate? ____ CM

1: 21 cm
2: 16 cm
3: 20 cm
4: 15 cm

Post-Experience Survey:

After your experience, how much space do you feel was necessary between the **TOP** of the PMT and the **BOTTOM** of the second payload is necessary? ____ CM

1: 15 cm
2: 22.5 cm
3: 22 cm
4: 17 cm

How much space would you actually allocate? ____ CM

1: 18 cm
2: 16 cm (redesign tool)
3: 25 cm
4: 23 cm

How much time would such a spacing error cost if discovered during the final payload layout?

1: This could be measured in days or months depending on the problem solution. A tool could be fashoned in days. If a box was demated, regression could take months.
2: 30 day at the least due to disassembly and retest. Could be more.
3: Could be extremely long - could cause partial disassembly/reassembly, or even redesign of physical layout! Partial disassembly/reassembly would be several days to weeks, but redesign could cost months.
4: Months of effort due to critical design considerations.

How much money would such a spacing error cost if discovered during the final payload layout?

1: A marching army of personnel waiting on a fix could cost hundreds of thousands of dollars. Launch delays would push this into millions of dollars.
2: Least cost in $, but a huge hit in schedule which is $.
3: Unable to estimate - depending on delay, could cost well over $100K to over $1M, and such delays and cost overruns could cause launch slip, mission reschedule or even project cancellation.
4: Could cost in the hundreds of thousands.

After your experience, what ***additional*** payload development tasks do you potentially see VR technologies aiding?

1: mechanical latches.
2: All pieces mounted; clearance of cable & connectors during GSE & flight cable use (do connector savers change the configuration?); remove before launch items (enough clearance?).
3: Any tasks where physical size/location of objects is an issue.
4: A to Z

Please write down some issues or problems you currently have with a specific payload development tasks and what tool, hardware, or software would assist you?

1: Fitting the integrated CALIPSO model into the clean shipping container. How do we orient the payload in the container? Where do we place access panels (for people) and cable feed-thrus?
2: Location of cable & connector interfaces.
3:
4: 1) My biggest concern (as mentioned) could be continuity between multiple players (private & government). Being on the same page when in the design phase. 2) When VR is in a refined state, I believe the benefits are enormous. (Cost savings, Time, & Minimize Gotchas).