Jonathan C. Silverstein*
Fred Dech
Department of Surgery
The University of Chicago
Chicago IL 60637-1470 USA

*Correspondence to
jcs@uchicago.edu

Precisely Exploring Medical Models and Volumes in Collaborative Virtual Reality

Abstract

We describe a virtual-reality widget library and two medical applications built on the widget library. These two applications, education using surface models and radiological volume visualization, make use of collaborative interaction techniques. These techniques support a high degree of precision with respect to manipulation of data and data parameters. The 3D widgets instantiated in these applications are synchronized between clients in order to facilitate the high degree of interactivity necessary for productive investigation of shared medical models and volume data. We discuss challenges that face the investigator in an immersive 3D environment as opposed to that of a 2D desktop environment. We describe how these differences have led us to criteria for development of the shared 3D Virtual Reality (VR) graphical user interfaces (GUIs) used in the biomedical applications presented. We review our educational validations already conducted for the surface model exploration application and preview our future work toward a single advanced biomedical collaboration environment.

1 Introduction

Traditional user interaction with medical diagnostic and educational applications involves a keyboard and mouse, with an operating system or window manager containing various tools. Such 2D environments are most often displayed on a desktop monitor. It is routine on these desktop applications to manipulate multiple parameters that in turn affect the presentation of the medical data being examined. A high degree of accuracy and precision can be maintained through keyboard and mouse input and manipulation of 2D widgets in an application's user interface. While the keyboard and traditional mouse may still be utilized in nonimmersive or fish-tank VR, they are no longer suitable in immersive VR. Our definition of immersive VR is a system that uses a sufficiently wide field of view, stereo imaging, and tracked, viewer-centered perspective. Collaborators collectively immersed in such a computer-generated environment must use a different set of tools in order to interact.

We make a distinction between two different methods of operating in VR. One method places the users within a space to be navigated and explored. In this method, remote collaborators are often far apart from one another in virtual space. Anthropomorphic avatars of varying degrees of sophistication typically represent them. They must have some means of navigating through the large virtual space that surrounds them. A second method places the users to-

Presence, *Vol. 14, No. 1, February 2005, 47–59*

gether with an object or objects in space to be examined. Here, the investigators are not exploring a space, per se, but rather they are congregated together in virtual space to examine and manipulate a data object. This is typical of biomedical investigation in VR. The data representations in virtual space are rarely much larger than the clinicians conducting the investigation, and are often smaller. The need for navigation in this second method or approach is less important or completely absent, since there is literally no place else to go. All remotely connected users are in close virtual proximity to one another. The data being investigated are already virtually within touching distance to all participants. The first model occurs predominantly in industrial, architectural, and art-related applications. The second model occurs most commonly in clinical medicine or biomedical research. We will refer to the first method as a virtual reality space method, or a *VRSM,* and to the second method as a virtual reality object method, or a *VROM.*

In the last decade there has been increased interest and activity in immersive VR in biomedicine, particularly for teaching, motivated by proof-of-concept works by several leaders in the field (Satava, 1994, Coleman et al. 1997, Hoffman & Vu, 1997, Ackerman, 1999, Stredney et al., 1999). At the annual Medicine Meets Virtual Reality international conference (NextMed, Inc., 2004), now in its 12th year, numerous new immersive and nonimmersive “virtual-reality” applications are presented, but few are ever evaluated by users due to the complex nature of biomedical teams and integrating new software into clinical evaluation/practice. In fact, in the biomedical domain, rigorous evaluations of immersive environments are just appearing (Seymour et al., 2002) while evaluations of immersive collaborative environments are almost unheard of except in neuropsychology (Tarr & Warren, 2002). In short, immersive collaboration is an enabling infrastructure that has not yet been seriously evaluated in biomedicine. Our work is based on the premise that if these and other advanced collaboration technologies are evaluated and successfully integrated into research, curricula, diagnostics, and treatment, we can overcome the difficulties in communicating complex 3D structure information among biomedical team members.

We have chosen a workdesk-style VR system for most of our biomedical research. The ImmersaDeskC1 and ImmersaDeskR2 (Fakespace Systems Inc.) are widescreen, rear-projected systems reminiscent of a drafting table (Czernuszenko et al., 1997). They utilize active frame-sequential stereo, and head and hand tracking via an electromagnetic tracking system. The screen has a display area approximately 6 ft × 4 ft and is elevated to about waist height. The screen is set at an angle with the top approximately 45° back. This angle enables simultaneous down and out viewing. For medical applications, we prefer the Desk to the CAVE (Fakespace Systems Inc.) (Cruz-Neira, Sandin, & DeFanti, 1993). While the CAVE is far superior for VRSM applications, the Desk is at least as good, and often better, for the medical VROM applications on which our research is focused, due to higher brightness, easier user acceptance, and less costly deployment. As an input device, we use the Wand or V-Wand (Fakespace Systems Inc.). Both devices have three buttons and a joystick-like control. The hand-tracking sensor is normally attached to, or embedded in, the wand. The wand is a 6 degrees of freedom (DOF) interaction device and is in many ways analogous to the 2 DOF mouse.

2 Design Goals and Implementation

The need for a reusable library of 3D widgets capable of precise interaction in VR environments is strongly apparent when conducting biomedical VR application development. Many ad hoc GUIs and even some with a high degree of reusability are commonly in use in VR applications. These are often perfectly adequate, especially in VRSM applications where frequently the primary goal is navigating through a space, triggering events through proximity sensors or other means, and picking and then transforming objects. Systematic investigation of biomedical information, on the other hand, requires the ability to manipulate a vast array of parameters—some with a high degree of precision. Additionally, maintaining GUI- and data-state synchronization across multiple sites that are connecting and disconnecting to the application is critical if the collaborative VR investigation is to remain scientifically valid.

The collaborative applications to be discussed in this paper make use of a generalized VR widget toolkit, vwLib (Dech, 2004), which we have developed in order to facilitate 3D VR GUI construction. The 3D VR GUI is an integral aspect of our applications. Not only does it make precise manipulation possible on a local level, but also it is tightly coupled with the telecollaborative libraries used. This results in complete GUI and data synchronization across remote clients. It allows for multiple clients to interact with the data in a simultaneous manner. Any user at any site can immediately control the GUI to modify the data representation. No specific gesture or interface action is required to take control. In fact all users technically have complete privilege to manipulate the environment at all times. This does not turn out to be chaotic. The interactions end up working in much the same way that multiple persons carry on a polite group conversation. One user (remote site) has the floor and others observe and interact in a constructive manner until control is yielded (socially—not technically) to another site. Shared channel voice communication (with echo canceling) via streaming audio facilitates the cooperative approach.

2.1 Interface Library Description

In designing vwLib, our primary goals were reusability, efficiency, and ease of use. We use an object-oriented approach written in C++. For event generation and propagation, we use a combination of an observer pattern with a mediator (Gamma, Helm, Johnson, & Glissades, 1995). It is important to stress that these widgets are part of the VR environment. They are represented as polygonal geometry in the shared virtual space. All widgets have a geometric component, and therefore constitute a portion of the OpenGL Performer (SGI) scene graph that all of our VR applications utilize. CAVELib (VRCO Inc.) is the library used to drive the VR devices within our applications (Pape, 1996; Pape, Sandin, & DeFanti, 1999). CAVElib is a library for developing OpenGL and OpenGL Performer VR applications. With CAVElib, applications become integrated into VR environment platforms in a straightforward fashion. CAVElib supports both the CAVE and ImmersaDesk. The hardware-specific tasks (e.g., tracking, multiscreen display) needed for these devices are handled by the CAVE library and remain transparent to the application.

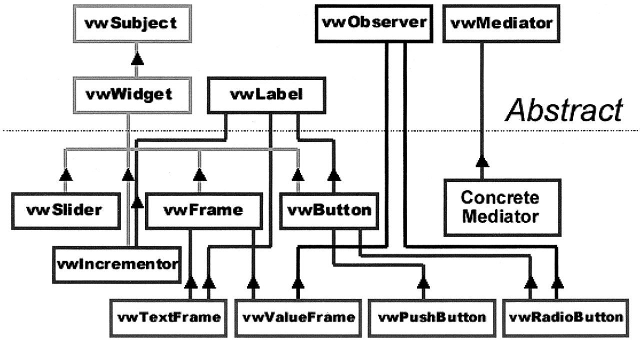

Figure 1. *This inheritance diagram illustrates the class structure of vwLib.*

The interface library, vwLib, was therefore built as a layer on top of both OGL Performer and CAVElib. It is made up of four generic widget classes: vwButton, vwSlider, vwFrame, and vwIncrementor (see Figure 1 for the inheritance diagram). An event occurs when a concrete subject calls the notify() method. All nonabstract subclasses of vwObserver (concrete observers) that have been attached, via attach(), to the notifying subject will generate an update() call. A subclass of vwMediator (concrete mediator) manages the subject-observer mappings and handles the event propagation. All widgets have a Performer grouping node associated with them (pfDCS in the case of vwButton and vwSlider and pfSCS in the case of vwIncrementor and vwFrame). Widgets can act as both vwSubjects and vwObservers. The vwRadioButton is a good example of this. An application can have numerous nonwidget concrete observers. Through their update() methods, they may take direct action or act as intermediaries between an interface and other portions of the application.

2.2 Teleimmersive Collaboration

CAVERNSoft G2 (Park et al., 2000) is a C++ toolkit designed specifically to enable interactive, networked collaborative applications. Rather than incorporating telecollaborative methods directly into the widgets, we have developed a GUI-state class that utilizes CAVERNSoft G2 to enable our networked interaction. This is a straightforward choice because the requirement for random and complete GUI synchronization among remote clients makes encapsulation of network behavior inside individual widgets problematic. Thus, multiuser functionality is not directly embedded into vwLib objects per se, yet the objects behave as networked elements. We can achieve multiuser behavior due to the client-server model used.

Each connected client maintains a netGUI object of the current state information of that client. When a local client's state changes (e.g., a local user moves a slider), the event cascade results in a modification of the netGUI's data structure. When this occurs, the netGUI is transmitted to the server application, which in turn broadcasts the new state information to all other clients. Synchronization is therefore constantly preserved. If a new client connects to the server during an existing collaborative VR session, a copy of the current netGUI database is retrieved from the server and the new client attains complete synchronization with all the other clients (see Figure 2). An initial client-server connection is established using TCP. This connection information is reflected from the server to clients that are already actively collaborating. These clients add the new client to their personal user database and then respond with a TCP message so that the new client can add them to its user database. The local user databases are used to create and manage avatars that represent the remote collaborators. When possible, a separate CAVERN sound-server process is also used. A sound-client utility is then run on a machine patched into the ImmersaDesk audio system. This CAVERN utility streams audio over a UDP socket or sockets to the server, which reflects the data to all other audio clients.

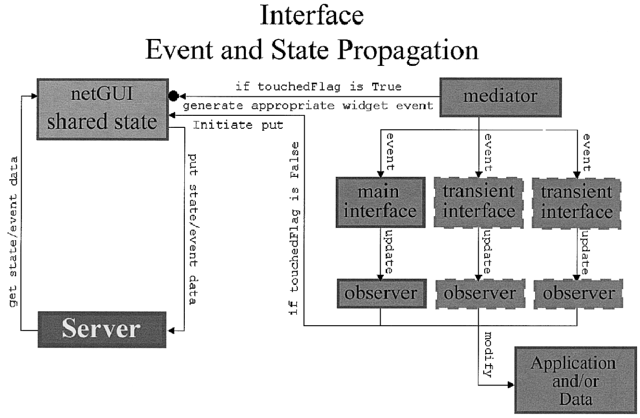

Figure 2. *This flow diagram illustrates the manner in which one client's GUI and application data representation maintain synchronization with the server. All participating clients maintain this flow-diagram structure with the server.*

An excellent review of applications based on CAVERNsoft and issues associated with teleimmersive collaboration was published by Leigh, Johnson, DeFanti, and Brown (1999). That paper reported the retrofitting of a powerful scientific visualization environment, the Virtual Chesapeake Bay (Wheless et al. 1996), to be used as a collaborative virtual environment including user avatars with eyes, bodies, and arms, sophisticated graphical interfaces controlled by wand devices, and private data spaces that could be interactively shared or isolated from other users (see http://www.evl.uic.edu/akapoor/cave6d/Accessed 9/9/2004). Given the substantial limitations noted in managing such complex interfaces and avatars intuitively, the substantial overhead in managing such a complex software environment with an Information Request Broker for each client, and our own design goals, we chose simplicity over sophistication in our interface without sacrificing sophistication in collaboration features. Thus, our interface design is as simple as possible in order to maximize user understanding in the medical domain, where immersive collaborative environments are extremely rare. Still, all users of our applications have complete simultaneous interface control at all times while user-centered perspective is maintained. In this way, the model and interface visualization is "personal" but the state of all interface elements and avatars is shared among n-clients.

In particular, for our VROM applications we have found that enabling navigation of clients' local CAVE coordinates through the virtual space is a hindrance rather than an asset. Therefore, as long as all clients are on desk VR devices, we can safely assume that everyone is facing in more or less the same direction in virtual world coordinates. While we do collect head-tracker information from the clients in order generate viewer-centered perspective, our avatars use only the hand (wand) tracker information. For vwLib based GUI manipulation, we have found that a slender cone avatar emanating from the front of the wand works well. Such a slender cone also makes an excellent device by which individual remote clients can point out items of interest to the other remote collaborators. This holds true, as well, for passive participants local to the tracked person who is doing the pointing. Each participating client gets a uniquely colored pointing device that can be seen by all connected sites. The base of the cone is at the front of the wand in virtual space and it points along the vector emanating from the wand (i.e., it points like a laser pointer).

2.3 Interface Usability

Pointing a slender cone at 3D widgets in VR is in some ways similar to using a laser pointer to illuminate items on a projected image in a conference room. There are a number of issues that make this somewhat difficult. Hand-jitter is an obvious culprit (Olsen & Nielson, 2000). Most people would agree that sliding a mouse over a 2D plane and letting it come to rest over a 2D widget is a much easier method for selecting widgets. The mouse has 2 DOF and it doesn't move around much when the user's hand is resting on it. Moreover, the mouse does not move once it is let go. Wand manipulation in VR applications is much more difficult than mouse manipulation, largely due to the fact that the wand user has 6 DOF to deal with instead of 2. There is no haptic feedback from either a mouse pad or wand, and there is nothing analogous to the mouse pad/table to steady the user's hand (Mine, Brooks, & Sequin, 1997; Lindeman, Sibert, & Hahn, 2000; Lindeman, Sibert, & Templeman, 2001). One cannot *release* the wand as one can release the mouse from one's hand. The wand therefore has no straightforward position and orientation memory. For this reason, cascading menus were not even considered as a widget type in vwLib. Tracker calibration and latency issues are also potential pitfalls. However, these can be overcome to the point where their significance is minimal (Czernuszenko, Sandin, & DeFanti, 1998).

One of the most obvious factors influencing target or widget selection by means of a pointing device is the amount of distance in virtual space between the wand and the widget or GUI. We minimize this distance by situating the widgets on the virtual plane that corresponds to the physical plane of the projection screen. By design, the ImmersaDesk requires that users be as close to the physical device as possible. The closer one is to the screen, the wider the peripheral image footprint becomes. Therefore, in most instances, it becomes detrimental to attempt to bring the widgets further out than the screen in virtual space. This would risk putting the GUI behind the user's viewpoint in virtual space. During a collaborative session, the application GUI is generally fixed in virtual space and its virtual location is identical for all participating clients.

When a researcher at one site "has the floor" and is manipulating the interface, other collaborators see that remote researcher's pointer manipulating the interface/data in a natural way (i.e., they see that person's pointer modifying the interface). We have found that nonactive, tracked participants normally lower their wands unless they are attempting to point out or ask about something in the medical data. The unpleasant specter of five remote clients simultaneously waving their wands around and through the data is needless to say not an issue.

Both applications to be described have been tested on countless occasions. Formal evaluation results for teaching effectiveness and desirability are reported below in Section 3.1. In our experience, clinicians and biomedical-oriented research personnel adjust quickly to using the GUIs, requiring only minimal practice. For example, in our formal evaluations, anatomy teaching assistants and surgical educators were each comfortable teaching their students/residents with the system (a substantially higher barrier of usability than a single users using it for their own purposes) after only one 45 min technology overview and (group) practice. An example of a successful three-client session across long distances without training was conducted in the spring of 2001 between Chicago, Chile, and a second Chicago site (Dech, 2001).

In many cases the VR widgets making up the collaborative GUI need to be sized larger than one would initially expect in order for the user to manipulate them effectively. The reason for this is straightforward. It is a means to compensate for the hand-jitter that does not occur in mouse-driven applications. For this and other reasons, such as visual coherence, we have chosen an application interface model that consists of a main menu, usually composed of one or more groups of radio buttons (they behave together as radio buttons—only one in a group can be pressed down at a time). Some of these buttons, when selected, bring up child interfaces whose buttons may create second level child interfaces, and so on. Once the interface or interfaces are no longer required, deselecting their parent button destroys them. Interface cluttering in collaborative applications is ordinarily a concern. The transient child-interface model we have used keeps such distracting clutter to a minimum.

Finally, it is worth noting that hand-jitter and lack of support can make slider-pips difficult to select. Despite this, we are able to use sliders in our collaborative VR applications through several techniques. Once selected, the slider-pips remain selected as long as the left wand button remains depressed. Also, the left wand button can be held down, and as soon as the wand vector intersects the slider-pip geometry, the slider becomes selected and attached to the researcher's wand. Thus, slider movement is handled in such a way that does not require continual intersection with the slider-pip. In cases where high precision is desirable or required, we often attach an observer slider and an observer incrementor to one another. Thus, when the wand vector is intersecting the vwIncrementor, a left-button click decrements its value by an application-defined amount. A right wand-button click increments its value by the same predetermined amount. In this way, a slider's value can be *nudged* up and down.

3 Medical Applications

Two telecollaborative applications will be discussed. Current testing and research has remained confined to ImmersaDesk-type systems, although in principal these applications should work in CAVE and CAVE-like environments with minimal modification. Also, collaborative mixtures between CAVEs and IDesks, while not yet tested, do not present any prima facie roadblocks. As was discussed earlier, human interaction with the data in VR is achieved through the use of sophisticated VR GUIs. As new clients attach to the shared VR environments, they inherit the current GUI and data set of the application. If an existing client opens submenus, these submenus become open to all other attached clients and new clients bring up their application with the GUI in the same state as that of the other preexisting clients. If a different remote client takes over GUI/data manipulation, synchronization continues seamlessly. When one client drags a slider, for example, that slider is dragged at all other remote sites. It is a completely shared VR user interface that all can see and all can manipulate.

Again, current avatars consist simply of slender cones, with each client having a distinct color. The cone represents where remote clients are and what they are pointing at. Since navigation is disabled, each client sees one or more avatars pointing more or less from the same general position toward the shared GUI and the medical data object, which typically occupies most of the display. Both these applications share common transformation modes. In the translation and rotation mode, as the left wand button is depressed and held, the biomedical data object is attached to the changes in the position and orientation of the wand. The data remains in its final position and orientation when the wand button is released. In the scale mode, the data object scales up by an incremental value as the right wand button is depressed and held. It scales down by this incremental value as the left button is held down. A middle-button press in this mode resets the model to its initialized real-scale value.

3.1 Education Using Surface Models

In 1998, recognizing challenges of surgical education including rapid expansion of knowledge, limited availability of biological materials, and limited availability of expert educators, we began a multidisciplinary federally funded contract to build and assess educational applications using collaborative virtual reality. Both the surface and volume applications discussed here are a result of that effort. The key concept driving the educational application using surface models is (in response to the challenges above) to improve educational efficiency by enabling the surgical anatomy expert to teach students effectively at multiple VR sites simultaneously. Anatomic regions were selected for being highly complex and critical to understanding common problems, particularly where surgeons' conceptual visualization is difficult to achieve with lectures, 2D illustrations, or photos, or where cadaver dissection is difficult. The temporal bone, pelvic floor, and liver were selected. Geometries of these anatomic regions were generated using standard 3D illustration and modeling techniques from slice sections.

This application makes use of drop-down style menus that allow for transparency manipulation of the specific model-elements in the anatomic region presented (see Figure 3). The drop-down menus work by selecting a button from the main menu (e.g., LIVER). This produces a second-level child menu (drop-down menu) consisting of an overall transparency slider and a set of other buttons. These new buttons can be selected to further refine manipulation of transparency to individual elemental structures (e.g., segments I, II, III, IVA, IVB, V, VI, VII, and VIII). Once again, as these drop-down menus are created by one client (in most cases by the master surgeon), they are simultaneously created at all other participating client stations. The drop-down child menus are then removed as their parent widget is either deselected, or another radio button is selected in its stead. All the separate anatomies for which there are models are represented in the application's menus.

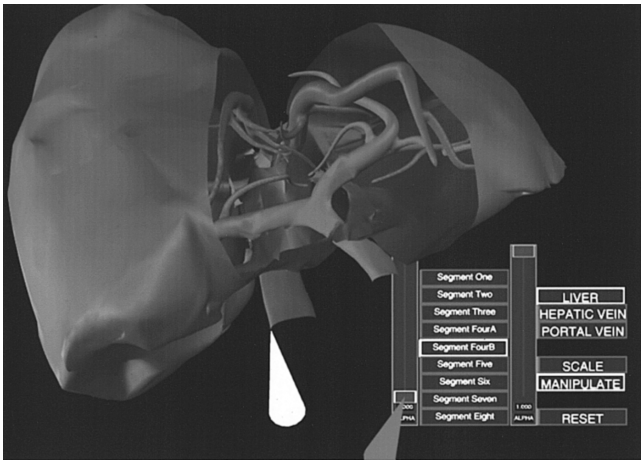

Figure 3. *This image illustrates the education application being used to teach surgical anatomy of the liver. The hepatic and portal vasculature can be seen within the liver geometry. Two user avatars are shown. One user has rendered segments 4A and 4B completely transparent via its alpha slider while the other user points to an area of interest.*

The anatomic structures under investigation in Figure 3 are the liver, the hepatic veins, the portal veins, and elemental submodels of each (such as liver segments). The educator is demonstrating the complexity of the central liver (segments 4A and 4B) and pointing out the plane of resection for a right lobectomy. To generate this image, after loading the application (preconfigured for the liver geometries), the users collectively have clicked SCALE to enlarge the models, clicked MANIPULATE to modify the models' orientation with a subsequent click-and-drag, clicked LIVER to bring up the liver segments' child interface, clicked SEGMENT FOURA, and moved the (segment 4A) slider to render that segment completely transparent, clicked SEGMENT FOURB, and moved the (segment 4B) slider to render that segment completely transparent, and finally waved the wand to demonstrate the resection plane of interest.

The students and teacher communicate verbally with one another through the integrated audio server that streams audio to and from the various clients. The surgical educator uses his or her wand-pointer avatar not only to select and manipulate widgets, thereby adding, subtracting, and otherwise manipulating the anatomic objects, but also to point out anatomical features to the remote collaborators. This application works adequately on an Octane SE (SGI).

This anatomic model visualization application is in fact a general-purpose tool. It can easily be customized to accept a variety of specific surface models that were preconstructed for the purposes of telecollaborative education. In fact, applying it to new data requires only including the new hierarchical list of structures and the associated geometry files. The first proof-of-concept version was the original Virtual Temporal Bone application (Mason et al., 2000) but it was never formally tested. Subsequently, as we describe in the next three paragraphs, versions of this software have been tested for three different regions of the anatomy in three separate formal educational tests, all demonstrating statistically significant educational benefits and high marks for usability.

The first formal test of our teleimmersive education applications was done on the Virtual Pelvic Floor in a series of short workshops (Dobson et al., 2003). Training on the Virtual Pelvic Floor produced substantial and statistically significant improvements from pre- to posttest scores of 13 senior surgical residents on a 10-question test ($p < .001$, approximately from 60% to 90% correct). Resident evaluations after the workshops (13 of 13) also confirmed their desire for further workshops using the application, comfort with the technology at the conclusion of the workshop, and perceived effectiveness of understanding pelvic anatomy using the virtual reality. No differences were detectable in performance between those at the site of the instructor and remote participants.

Next, the liver application we've described in detail above (for teaching surgical liver anatomy) was formally tested with six other senior surgical residents at two sites (Silverstein et al., 2002). A 24-question examination was administered before and after a roughly 45-min workshop and again six months later. The workshop produced significant improvements in the mean test scores between the pre- and posttests (49% to 66%, $p < .02$). Most importantly, six-month delayed testing demonstrated complete retention of new knowledge for all residents, while some residents' knowledge of surgical liver anatomy substantially improved on delayed testing. This surprising finding was reportedly because the intense VR experience gave the residents a new framework for understanding the complex 3D relationships that make up the liver. This allowed them to read advanced textbooks that were previously too difficult to understand. We found no differences between residents who were with the instructor and those at the remote location. Subsequent resident classes have demanded the session be given and more than a dozen other residents among several residency programs have participated (without additional formal tests).

As further evidence that the application is of general-purpose use, a newer human temporal-bone application was constructed using the same code as for the liver sessions but configured for newer inner-ear anatomy models. One feature of interest in this usage was that one key slice from the original cadaver histological data set (used to generate geometry files) was included as a model (single-textured plane) in the final application. This was used in a semitransparent fashion to demonstrate the relationship between 3D and 2D anatomy by allowing students to visualize both the anatomy and essentially the standard diagnostic images of the anatomy (slice sections) simultaneously in the same coordinate space.

In spring of 2003, the application was formally used and evaluated in the first year inner-ear anatomy curriculum at the University of Chicago Pritzker School of Medicine (Silverstein et al., forthcoming). After a single training session for both the anatomy (45 min) and the technology (45 min), anatomy teaching assistants gave 20-min workshops to 49 first-year medical student volunteers in groups of five or fewer before or after a traditional 3 h lecture/laboratory session. Twenty-seven students who volunteered for only the traditional session served as a comparison group. A 15-question examination designed to evaluate understanding of complex 3D relationships and knowledge retention was administered before and after each workshop or session. Also, user satisfaction questionnaires were administered and discussion groups held following each use of the application.

The average improvement from pre- to posttests demonstrated statistically significant advantage of the

Table 1. *Groups of Student Volunteers*

Group	Monday	Tuesday	Wednesday	Thursday
Traditional (27 students)	Pretest		Lecture/Lab 1st Post-Test	
Avg. ± *SD* (%)	3.2 ± 2.0 (21.3%)		6.7 ± 1.8 (44.7%)	
VR First (19 students)	Pretest	VR Session 1st Post-Test	Lecture/Lab 2nd Post-Test	
Avg. ± *SD* (%)	3.1 ± 1.6 (20.7%)	8.6 ± 2.1 (57.3%)	9.6 ± 2.6 (64%)	
VR Second (30 students)	Pretest		Lecture/Lab 1st Post-Test	VR Session 2nd Post-Test
Avg. ± *SD* (%)	3.5 ± 1.6 (23.3%)		6.7 ± 1.9 (44.7%)	9.2 ± 2.1 (61.3%)
Excluded (11 students)	Pretest		No lecture/lab OR No post-test	

Total of 87 students from a class of 95 volunteered to participate. Group average scores ± *SD* and (%) are presented.

brief immersive virtual reality session alone over the traditional 3 h lecture/laboratory session alone (5.5 pts. vs 3.2 pts, p = .002 two-tailed paired t-tests). The overall improvement for those who were exposed to both the traditional and immersive virtual-reality sessions was also statistically better than those exposed only to traditional methods (6.1 pts vs 3.5 pts, p = .0002 two-tailed paired t-tests). See Table 1.

We concluded that it enhanced students' understanding of complex 3D relationships similarly whether conducted before or after the traditional session and that it was more efficient than the traditional session. Satisfaction questionnaires and discussion groups also demonstrated that the students found the immersive teaching method efficient, effective, and enjoyable (4 or greater on all questions on 5-point Likert scale) and that the students favored further use of the technology 45 to 4. See Table 2.

Table 2. *Results of Survey (Complete for All 49 VR Participants)**

Survey question	Average result on Likert scale
I found the instructor easy to understand.	4.0
I found the teleimmersive technology an enjoyable way to learn this material.	4.4
The teleimmersive technology helped me to better master the subject material.	4.3
I feel that I know more about the material from using this technology than I would have under traditional methods.	4.0
The teleimmersive technology is an efficient way to learn the subject matter.	4.0
I would like to take additional courses using this technology.	Yes = 45, No = 4

*Note: Likert scale was used where 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree.

3.2 Radiological Volume Visualization

We have developed a tool for the investigation of volumetric radiological data (e.g., CT, MR) in a telecollaborative VR environment (see Figures 4 through 7). This application's primary purpose is telecollaborative

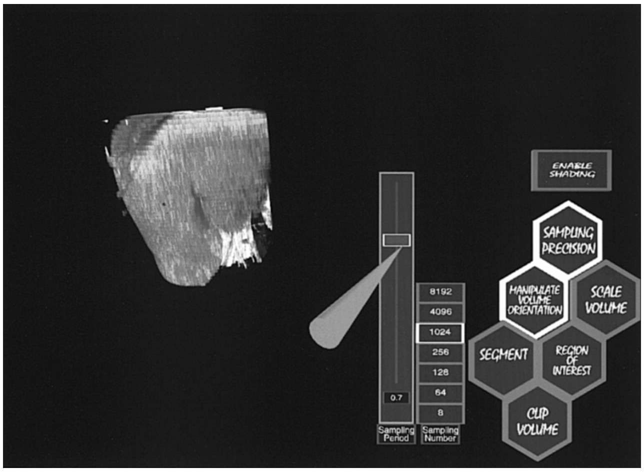

Figure 4. *This image illustrates the sampling-precision interface. The wand pointer is intersecting the slider widget controlling the sampling period (set in this case to 0.7). The sampling number has already been set to create 1024 translucent polygons for the visualization.*

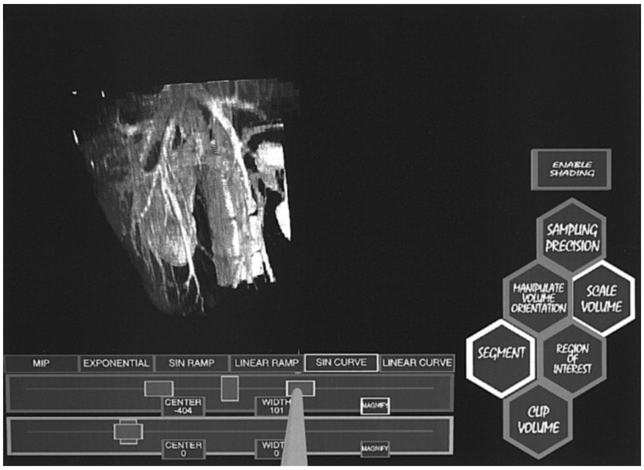

Figure 5. *This image illustrates the segmentation interface. The wand pointer is intersecting the slider widget controlling the data window width. Magnification is on.*

examination of CT and MR scans; however, we have found it useful in visualizing many other types of slice-section volumetric data sets. It shows promise in the areas of medical education and potentially as a clinical tool to assist in diagnosis and in preoperative planning. It has not yet undergone user evaluation (funded, but not yet performed). This application directly imports DICOM3 compatible data on the fly, typically in under 1 min even for large data sets. Radiological data is transmitted automatically from the data server to clients as they connect to the telecollaborative application (Dech, 2001). Clients are typically Onyx2 IR2 (or higher) visualization computers (SGI).

Once the data have been loaded into shared memory, they are converted and segmented into *memory bricks* within an OGL Performer geometry class. This class contains all of the OpenGL Volumizer (an OpenGL C++ volume rendering library) construction routines (Grzeszczuk, Henn, & Yagel, 1998). Volumizer is much faster than other approaches (e.g., ray casting) because computations are performed by the dedicated texture-mapping hardware rather than by the CPU. The volume is reduced to a series of texture-mapped, translucent polygons. Current data types supported are 8-bit and 16- (12-) bit grayscale, along with 8-bit and 16-bit pseudocolor. On a fully configured IR2, a 2-byte 512 × 512 × 244 volume can remain resident in texture memory, producing frame rates upwards of 30 fps. This remains well within the range of interactive speed. Much larger volumes can be studied if the volume geometry visualized in texture memory is subselected from the actual voxel dimensions of the volume in main memory. Using this approach, one can interactively roam through small subsets of the volume. This method is known as *region-of-interest* (ROI) manipulation.

The main menu consists of two groups of radio buttons. One group allows the user to select between one of two object-transformation modes previously discussed: manipulation and scale. The other main-menu radio-button group consists of buttons that create four child interfaces (segmentation, region of interest [ROI], sampling precision, and an arbitrary clipping plane (see Figures 4 through 7). The square button at the top adds gradient shading to the display.

The most complex child interface is the segmentation interface (see Figure 5). It consists of two identical windowing and leveling tools that, if used together, generate a segmentation curve mapping the higher of each of the two levels along the dynamic range of the data set

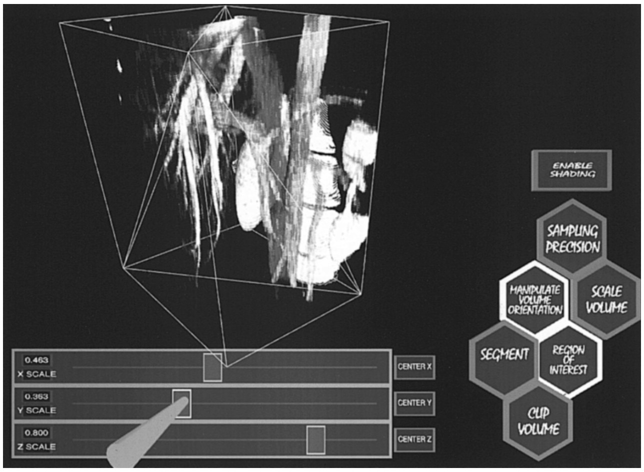

Figure 6. *This image illustrates the region of interest interface. The wand pointer is intersecting the slider widget controlling the data window width in the y dimension. The region of interest box creates 6 orthogonal clipping planes that persist in the data object's local coordinate system.*

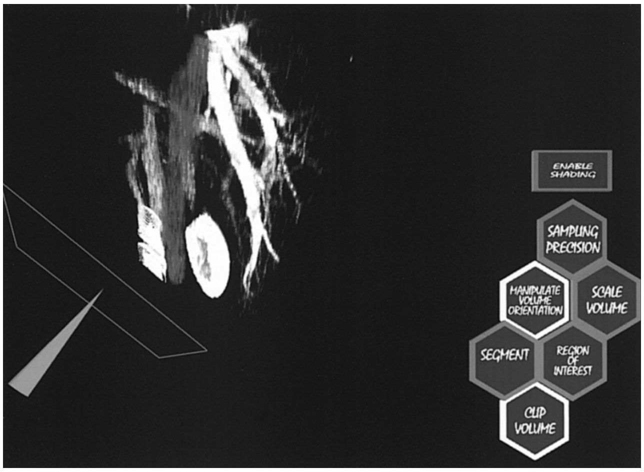

Figure 7. *This image illustrates the arbitrary clipping-plane mode while the left button is depressed. When the button is released, the clipping plane remains in the data object's local coordinate system.*

(normally 12 bits). Each tool has a level slider along with two window-width sliders. A group of radio buttons allows the user to select among four different segmentation-curve functions. Two other nonradio buttons allow for a maximum-intensity projection (MIP) mapping and a function that applies an exponential value to the segmentation curve before it is mapped to the data. The final button in the segmentation interface is the magnification button. This allows for precise control, no matter how small the data window. In order for this tool to work correctly, numerous subject-observer relationships are established among the various widgets.

Figures 4 through 7 demonstrate the four child interfaces being used in sequence to develop a visualization of the hepatic veins and portal system from a standard (5–7 mm slice) CT scan of the liver. In these examples, the user has already focused the region of interest and segmented the data to show the liver parenchyma. Figure 4 first demonstrates the sampling precision being increased. Then, after the data have been scaled larger, Figure 5 demonstrates the data being further segmented to identify the vasculature and other more dense features (such as the kidney and spine). Next, the volume orientation is changed and Figure 6 demonstrates focusing on an even finer region of interest. Finally, an arbitrary clipping plane is applied in Figure 7, ultimately removing the kidneys and spine to create the desired visualization.

4 Conclusions and Future Work

We have developed the infrastructure for a series of biomedical teleimmersive applications permitting distributed collaboration with desirable user interface tools. The two applications we have described allow the wand-controlling participants at all connected sites to manipulate any of the widgets. The interfaces are identical and synchronized at each location, such that a slider manipulation at one site, for example, is reflected at all other sites in the collaboration. Whenever a child interface is created or destroyed by one client, it is subsequently created or destroyed by all other clients. Clients can leave the collaboration by terminating their client application and rejoin at a later time, assuming that the server application is still running. Since the networked GUI database is persistent on the server, it is even possible for all clients to terminate their applications, and at a later time new client applications can reconnect to the server, with the GUI and data state being regenerated exactly as it was left when the final client disconnected from the earlier session.

We've demonstrated educational value, knowledge retention, and desirability of our surgical-anatomic education application for three different regional anatomies (pelvic floor, liver, and temporal bone) across the University of Illinois at Chicago and the University of Chicago. These studies have shown that this VR environment is effective for learning complex 3D relationships for both residents and medical students. We anticipate that a new educational framework can be developed in which difficult 3D relationships are presented in collaborative virtual reality before more traditional educational exercises. We have not, however, conducted a comparison study using the same models but different interface widgets (2D GUI and/or models, smaller displays, or web-based collaboration) to see which specific elements are responsible for the learning effects. We expect the learning effects are related to 3D model quality and to the widgets enabling intuitive manipulation in immersive collaborative environments (especially precise orientation, selection, and transparency).

A logical next step is to test across diverse platforms simultaneously (e.g., CAVE, IDesk, etc.). In the case of dissimilar devices, we would enable interactive GUI translation so that the GUI could be resituated for optimal usability depending upon which platform was manipulating the medical data. Recording and playback enhancements are also being planned. These features will allow particularly useful investigations to be reviewed again. This is thought to be valuable because the highly interactive nature of the interface makes it otherwise difficult to precisely recreate an interesting visualization. Ongoing work involves incorporating both surface and volume exploration features into an integrated visualization. This application will also include data fusion between anatomic surface atlases and radiological volumes.

Other features are being investigated, such as the capability of playback of videos and video teleconferencing in association with the virtual environment. These features will allow the surgical educator to play video captured during relevant operations and will allow other related diagnostic images and video to be shared in the radiology application. Specifically, we are embarking on a three-year project (titled Advanced Biomedical Collaboration) to rebuild these applications upon the Access Grid infrastructure (www.accessgrid.org) and test them in clinical use, thereby fully converging our biomedical virtual-reality and conferencing technologies.

Acknowledgments

This project has been funded in whole or in part with federal funds from the National Library of Medicine, National Institutes of Health, under contract N01-LM-9-3543, and under grant R01-LM-06756-01. We also acknowledge Peter Jurek and Dr. N. Joseph Espat for development of the liver model and Drs. Theodore Mason and J. J. Kempiners for directing development of the second virtual temporal bone model.

References

Ackerman, M. J. (1999). The visible human project: A resource for education. *Academic Medicine, 74*(6), 667–670.

Coleman, J., Savchenko, A., Goettsch, A., Wang, K., Bono, P., Littlefield, R., et al. (1997). TeleInViVo: A collaborative volume visualization application. *Studies in Health Technology and Informatics, 39,* 115–124.

Cruz-Neira, C., Sandin, D., & DeFanti, T. (1993). Surround screen projection-based virtual reality: The design and implementation of the CAVE. *Proceedings of SIGGRAPH '93 Computer Graphics Conference,* 135–142.

Czernuszenko, M., Pape, D., Sandin, D., DeFanti, T., Dawe, G., & Brown, M. (1997). The ImmersaDesk and infinity wall projection-based virtual reality displays. *Computer Graphics, 31*(2), 46–49.

Czernuszenko, M., Sandin, D., & DeFanti, T. (1998). Line of sight method for tracker calibration in projection-based VR systems. *Proceedings of 2nd International Immersive Projection Technology Workshop.*

Dech, F. (2001). The virtual temporal bone G2 and manipulation of volumetric patient data in a distributed virtual reality environment. *Science, Culture and Education over Internet2,* Valparaiso, Chile, Remote Demonstration from Chicago.

Dech, F. (2004). vwLib. Retrieved September 9, 2004, from http://www.evl.uic.edu/fred/vwLib/docs.

Dech F., Ai, Z., & Silverstein, J. C. (2001). Manipulation of volumetric patient data in a distributed virtual reality environment. *Medicine Meets Virtual Reality 2001,* 81, 119–125.

Dobson, H. D., Pearl, R. K., Orsay, C. P., Rasmussen, M., Evenhouse, R., Ai, Z., Blew, G., et al. (2003). Virtual reality: New method of teaching anorectal and pelvic floor anatomy. *Diseases of the Colon and Rectum, 46*(3), 349–352.

Gamma, E., Helm, R., Johnson, R., & Glissades, J. (1995). *Design patterns—Elements of reusable object-oriented software*. Reading, MA: Addison-Wesley.

Grzeszczuk, R., Henn, C., & Yagel, R. (1998). Advanced geometric techniques for ray casting volumes. *ACM SIGGRAPH '98 Course Notes,* 4.

Hoffman, H., & Vu, D. (1997). Virtual reality: Teaching tool of the twenty-first century? *Academic Medicine, 72*(12), 1076–1081.

Leigh, J., Johnson, A., DeFanti, T., & Brown, M. (1999). A review of tele-immersive applications in the cave research network. *Proceedings of IEEE VR '99,* 180–187.

Lindeman, R., Sibert, J., & Hahn, J. (2000). An intuitive VR user interface for design review. *Proceedings of the Working Conference on Advanced Visual Interfaces 2000,* 98–101.

Lindeman R., Sibert, J., & Templeman, J. (2001). The effect of 3D widget representation and simulated surface constraints on interaction in virtual environments. *Proceedings of IEEE Virtual Reality 2001,* 141–148.

Mason, T. P., Applebaum, E. L., Rasmussen, M., Millman, A., Evenhouse, R., & Panko, W. (2000). Virtual temporal bone: Creation and application of a new computer-based teaching tool. *Otolaryngology, Head and Neck Surgery, 122*(2), 168–173.

Mine, M., Brooks, F., Jr., & Sequin, C. (1997). Moving objects in space: Exploiting proprioception in virtual-environment interaction. *Proceedings of SIGGRAPH '97 Computer Graphics Conference,* 19–26.

NextMed, Inc. (2004). MMVR: Medicine meets virtual reality conference. Retrieved September 9, 2004, from http://www.nextmed.com/mmvr_virtual_reality.html.

Olsen, D., & Nielsen, T. (2000). Laser pointer interaction. *Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems,* 17–22.

Pape, D. (1996). A hardware-independent virtual reality development system. *IEEE Computer Graphics and Applications, 16*(4), 44–47.

Pape, D., Sandin, D., & DeFanti, T. (1999). Transparently supporting a wide range of VR and stereoscopic display devices. *Proceedings of the SPIE, Stereoscopic Displays and Virtual Reality Systems VI, 3639,* 346–353.

Park, K., Cho, Y., Krishnaprasad, N., Scharver, C., Lewis, M., Leigh, J., et al. (2000). CAVERNsoft G2: A toolkit for high performance tele-immersive collaboration. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology,* 8–15.

Satava, R. M. (1994). Emerging medical applications of virtual reality: A surgeon's perspective. *Artificial Intelligence in Medicine, 6*(4), 281–288.

Seymour, N. E., Gallagher, A. G., Roman, S. A., O'Brien, M. K., Bansal, V. K., Andersen, D. K., et al. (2002). Virtual reality training improves operating room performance: Results of a randomized, double-blinded study. *Annals of Surgery, 236*(4), 458–464.

Silverstein, J. C., Dech, F., Edison, M., Jurek, P., Helton, W. S., et al. (2002). Virtual reality: Immersive hepatic surgery educational environment (IHSEE). *Surgery, 132*(2), 274–277.

Silverstein, J. C., Ehrenfeld, J. M., Croft, D., Dech, F., Small, S., & Cook, S. (forthcoming). Tele-immersion: Preferred infrastructure for anatomy instruction. *Journal of Computing in Higher Education.*

Stredney, D., Crawfis, R., Wiet, G. J., Sessanna, D., Shareef, N., & Bryan, J. (1999). Interactive volume visualizations for synchronous and asynchronous remote collaboration. *Studies in Health Technology and Informatics, 62,* 344–350.

Tarr, M. J., & Warren, W. H. (2002). Virtual reality in behavioral neuroscience and beyond. *Nature Neuroscience, 5 Suppl.,* 1089–1092.

Wheless, G. H., Lascara, C. M., Valle-Levinson A., Brutzman, D. P., Sherman, W., Hibbard, W. L., et al. (1996). Virtual Chesapeake Bay: Interacting with a coupled physical/biological model. *IEEE Computer Graphics and Applications, 16*(4), 52–57.