An Immersive Tool for Wide-Area Collaborative Design

Brook Conner[1], Matt Cutts[2], Russ Fish[3], Henry Fuchs[2], Loring Holden[4], Marco Jacobs[2], Brian Loss[3], Lee Markosian[4], Rich Riesenfeld[3], Greg Turk[5]

The National Science Foundation Center for Computer Graphics and Scientific Visualization[6]

[4]Brown University

[2]University of North Carolina at Chapel Hill

[3]University of Utah

1. ABSTRACT

This paper describes a virtual reality application, *vrapp*, for telecollaborative engineering design. The prototype system interoperates with existing software and networks (e.g. vic, Open Inventor, Alpha_1 modeling system, the Internet) while leveraging greater capabilities where available, such as the Science and Technology Center's dedicated T-1 line for high-bandwidth video and local immersive displays (e.g. head-mounted displays or FakeSpace booms). The modularity of vrapp permitted the architects of this application to write, to test, to debug, and to use vrapp for real applications without the primary code writers ever meeting face to face. By providing a rich set of extensible tools in a shared, virtual environment, vrapp gives participants a better sense of communication and presence.

Keywords

collaborative design, virtual reality, object-oriented design

2. INTRODUCTION

To rapidly create, modify, and review designs interactively, it is necessary to assemble appropriate experts, users, and managers. Typically, experts are gathered for a design session by physical travel, telephone or video conferencing, or via a shared whiteboard application. Most shared whiteboards are not conducive to conversations, making it hard to communicate tradeoffs in a design to experts from outside one's own field. A goal of our immersive collaborative design environment is to have the designed objects included in the context of their environment, making possible judgments about spatial relations, and to have participants see each other, talk amongst themselves, and share interactions (interactive joint creation and modification) with design models. Our focus is on real design of real parts for real applications.

For example, mechanical engineering and manufacturing experts at the University of Utah and a video and optics specialist at the University of North Carolina at Chapel Hill (UNC) faced the task of jointly designing a new head-mounted display (HMD). They needed a convenient way to review and modify design choices together.

3. PREVIOUS WORK

A large number of research groups share our goal of being able to collaborate across wide distances using computer technology. An exhaustive list of related work would be prohibitively long, so we mention here only some of the more relevant work.

---

[1] Current address: Sony Corporation of America, 550 Madison Avenue, 5th floor, New York, New York, 10028, brook_conner@sonyusa.com

[2] Department of Computer Science, Sitterson Hall, CB #3175, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA, +1 919 962 1700, {cutts, fuchs, jacobs, turk}@cs.unc.edu

[3] Department of Computer Science, 3190 Merrill Engineering Bldg, University of Utah, Salt Lake City, UT 84112 USA, +1 801 581 8224, {bloss, fish, rfr}@cs.utah.edu

[4] Computer Science Department, Box 1910, Brown University, Providence, RI 02912 USA, +1 401 863 7600, {dbc, lsh, lem}@cs.brown.edu

[5] Current address: Georgia Institute of Technology, Atlanta, GA, 30332-0280

[6] The authors' schools are three of five partners in the National Science Foundation Science and Technology Center for Computer Graphics and Scientific Visualization. The other partners are the California Institute of Technology and Cornell University.

The Shastra project [2] at Purdue is directed at research in computer supported cooperative work such as geometric modeling, simulation, interrogative visualization and design prototyping environments. That system runs on a multimedia desktop. The system detailed in [1] combines the ideas of distributed virtual environments and collaboration. That system allows users to cooperate on furniture placement for interior design but lacks representations of the users within the virtual space.

The ATR group in Japan has used wide-projection screens to deliver video to a participant in a shared design system [7]. They simulated the facial expressions of one of the participants by warping the geometry of that participant's face (captured with a laser scanner off-line) according to the movement of tracked dots on that person's face.

The only other design system that we are familiar with that also places the user in an immersive environment is the work at ATR, thus it is worth noting differences between our project and theirs. Perhaps the most important difference is in our choices of interaction paradigms for design operations. ATR's system uses gestures (with a tracked glove) and speech input to specify actions such as command selection and object placement. Because design of mechano-optical parts is a task that demands both precision and a wide variety of operators, we chose to retain Alpha_1's (see below) powerful LISP-based syntax as our way of performing modeling operations. This means that one user types into a text window for changes to the model. Changes to an object's position for viewing (an operation that does not need to be precise) can be performed by grabbing the object with a 3D mouse.

4. FUNCTIONALITY

Two primary goals of vrapp are to provide a viable environment for collaborative design and to give a better sense of presence among colleagues at a distance.

For design, the vrapp system interfaces to Alpha_1 [5], the University of Utah's system for geometric modeling, visualization, and manufacturing. In Alpha_1's suite of tools, models are parameterized according to their design features. The vrapp system currently allows one user to control an Alpha_1 model. That user can highlight and modify parts of the assembly. As the controlling user alters the values in the model's parameterization, changes are propagated to the other viewers. A locking mechanism ensures that any participant, one at a time, can rotate, translate, and scale the model.

Regarding a sense of presence, we contend that there are many non-verbal cues that collaborators make that are important to communication in a number of settings, including engineering design. Some of these cues include facial expressions, direction of gaze, apparent level of attention, and pointing and gesturing both with hands and the head. We feel that ultimately the best way to represent these visual cues is to place in the shared virtual environment visual representations of one's collaborators that are as nearly real as possible.

Avatars

In our prototype system, embodiments of collaborators are represented by static and video images. These surrogates of individuals are called “avatars.” An avatar can take any form. When the user moves or changes viewing direction, the avatar mimics that behavior in the virtual environment.

Static and dynamic textures can be incorporated into the environment and onto avatars. Figure 1 shows Brook Conner at Brown giving a short talk with Powerpoint™ slides as wallpaper. Unchanging images can convey information, but rarely give subtle nuances like a frown or a confused expression. To help convey these nonverbal cues, vrapp incorporates "animated textures." This feature employs a wrapper around a network video tool called vic [4] to provide texture-mapped video from a remote source. With this feature a video stream can be brought into the environment. The system uses this to map a live video feed onto one of the avatars. The video can also be useful to show, for instance, machine parts at remote sites.

Figure 1. An example image from the environment. A CAD/CAM model rests on the table. An avatar from a boom viewer at Brown University “sits” over one chair. The other avatar shows a video stream from the University of Utah. Powerpoint™ slides hang on the walls of the meeting room. The screen shows what an HMD viewer at UNC sees.

Another capability of vrapp is multiple viewing platforms. Since vrapp renders the 3D view with Open Inventor (see next section), it can be easily ported to any machine with support for that toolkit. Each site wrote an Open Inventor viewer for the hardware it had locally. To date, the authors have added viewers for the FakeSpace boom viewer and some HMDs via UNC's trackerlib package. Thus the environment can be viewed not only from a regular workstation, but also via a spectrum of virtual reality display devices. The latter options permit users to collaborate in an immersive environment.

To help boost the sense of sharing the same space, participants work in a model of a real meeting room in Sitterson Hall (the building which houses the computer science department at UNC). The room is accurately modeled down to the same chairs and whiteboards as the real room. The model includes radiosity shading to provide a more realistic environment.

5. IMPLEMENTATION

The implementation of vrapp emerged from the following specific software goals:

- rapid development
- easy integration of code from different sites and from public domain sources
- extensible, dynamically loaded modules for behavior

In order to realize these goals, the schools chose Open Inventor [6] as a prototyping platform. Open Inventor is a software package from Silicon Graphics, Inc., that provides flexible scene modeling in OpenGL, a standard graphics library. Scenes in Open Inventor are modeled with a graph of nodes that can describe geometry, material properties, lights, and cameras. Open Inventor includes shape nodes for geometry, property nodes that modify attributes of objects, group nodes to connect nodes into graphs, and camera and light nodes. Avatars are represented by Open Inventor nodes. Any valid Open Inventor scene graph can be an avatar or object in the vrapp model.

Moreover, user-written nodes can be written to extend the functionality of Open Inventor. In vrapp, we used an animated texture node that texture maps a stream of video onto geometry. Also, we used an Alpha_1 node that displayed geometry that was received from an Alpha_1 server. Other parts of vrapp rely on standards like network sockets as well as locally written packages like UNC's trackerlib for HMDs.

Putting code into modular nodes eases the task of managing integration. Nodes can be distributed in binary form, as was the node for interacting with Alpha_1. Likewise, some viewers can be local to a site. For example, local viewers suffice for specialized hardware like boom viewers and HMD interfaces. Code can also be sent out as source and compiled at each site (this was the case for the animated texture nodes). These procedures lessen the need to integrate programming interfaces and reduce source interdependencies.

Another simplifying factor is the Open Inventor feature to dynamically load nodes at run time. When an unknown node is encountered, Open Inventor searches for a *node.so* file, much like shared libraries for programming languages. This allows the behaviors of nodes to be updated or changed without recompiling the whole program.

Despite the strengths of node-based code, Open Inventor remains merely an extensible graphics toolkit and lacks any notion of client/server interaction. The vrapp architecture provides arbitration of objects among avatars over a network. Communication is peer-to-peer with TCP sockets (one socket for each pair of participants). This model worked for simple purposes, but would not scale to tens of participants or more. The vrapp application is derived from a C++ base class called NetApp. NetApp defines a prototype distributed application which does routine tasks like manage connections to peers within a session.

Managing Objects

The subclass VRapp maintains a list of objects in the virtual world. These are two basic kinds of objects: real and ghost. Anything in the three dimensional scene, such as chairs, Alpha_1 models, and avatars, must have some representation within each peer process. To maintain consistency, a single representation of a given object is the master copy (i.e. "real one"), and all the other representations just mirror that one (i.e. they are "ghost" copies). The "real" copy of an avatar resides in its process, and it causes ghosts to be created within each of the other peer processes. When a user moves, her avatar sends updated position information to each of its ghosts. Then the other participants see the avatar move.

Inanimate objects can be "grabbed." When a user clicks on a piece of "furniture" (e.g. the Alpha_1 model), she will have a ghost copy of the furniture on her machine. The ghost copy has to respond by agreeing to be grabbed or by refusing. The ghost copy actually sends a message to its real half, which makes the decision and sends the answer back to the ghost. Currently, if an object is not already grabbed, it says yes. The ghost gets the message, and responds to the user accordingly. If the grab succeeds, the user gets some kind of manipulator which allows translation, rotation, and/or scaling the object. While a participant moves the object, the ghost sends updates to the real copy, which forwards them on to all the other ghosts.

This way, the other people participating in the group see objects moving as the user manipulates it. The design of the system permits objects to be added at any time from a command line by specifying an Open Inventor file.

6. RESULTS

The creators of vrapp demonstrated the system recently for a set of designers and visitors. Russ Fish at the University of Utah and D'nardo Colucci at the UNC had already designed the mechanics and optics of a lightweight HMD.

The design of the HMD balances factors such as weight and placement of cameras, mirrors, and the LCD displays. Although the HMD wasn't designed in the vrapp system, it was parameterized in Alpha_1. After Alpha_1 and vrapp were integrated, the designers could go back and explore changes in the position and orientation of different components. The HMD designers commented that the system would be viable for design collaboration.

7. CONCLUSIONS

The vrapp environment achieves progress in its specific goal of collaborative design and modification. This driving application motivated the integration of many software packages and several diverse hardware platforms. Thus, vrapp serves as a useful test of integration for real tasks. A primary idea of vrapp is to avoid "reinventing the wheel." By adapting off-the-shelf component software, the schools were able to build a working prototype in a very short time. Existing applications software such as Alpha_1 is interfaced through "gateway nodes," rather than re-implemented in Open Inventor.

In summary, we have made an initial step towards developing a larger environment appropriate to the goals described in the introduction. In view of the substantial benefits and economies that a fully functioning system could provide, we feel that it is important to fully develop and test out the concepts.

8. FUTURE WORK

The authors are looking into ways of dealing with communication latency [3]. Several other methods of interacting during collaboration are also being considered: pointing to objects, shared whiteboards, and audio feedback. One piece of future work is to make this a truly shared collaborative environment where multiple participants can change the model. Another potential area of investigation is how to direct participants' attention to a particular object or feature within the environment. The prototype system also serves as a good testbed for user interface research in virtual reality.

9. ACKNOWLEDGMENTS

The authors wish to thank the National Science Foundation for supporting this research under Cooperative Agreement ASC-8920219 and the Principal Investigators of the Science and Technology Center for their vision and encouragement. We thank all the members of the Science and Technology Center for their help. This research was also supported under DARPA contract number DABT63-93-C-0048. Mark Bloomenthal at the University of Utah assisted in discussions for integrating Alpha_1 and Open Inventor.

Open Inventor™ and OpenGL™ are trademarks of Silicon Graphics, Inc. PowerPoint™ is a trademark of Microsoft, Inc. The vic tool was created by Lawrence Berkeley National Laboratory in collaboration with the University of California, Berkeley.

10. REFERENCES

1. Ahlers, Klaus H., Andre Kramer, David E. Breen, Pierre-Yves Chevalier, Chris Crampton, Eric Rose, Mirharan Tuceryan, Ross T. Whittaker, Douglas Greer. "Distributed Augmented Reality for Collaborative Design Applications." in Computer Graphics Forum, Vol. 14, No.3 (Eurographics '95, Maastricht, The Netherlands August 28 - September 1, 1995). pp. C-4 to C-14.

2. Anupam V. and Bajaj C. "Shastra: An architecture for development of collaborative appplications." Technical Report 93-033, Purdue University, 1993.

3. Conner, Brook and Loring Holden, "Providing a Low-Latency User Experience in a High-Latency Application". To appear in Proceedings of 1997 Symposium on Interactive 3D Graphics (Providence, Rhode Island, April 27-30, 1997).

4. McCanne, S. and V. Jacobson. vic: A Flexible Framework for Packet Video, In proceedings of the ACM Multimedia, 1995 (San Francisco, CA, USA, November 5-9, 1995) ACM, NY, 1995, pp 511-522.

5. Riesenfeld R., "Modeling with NURBS Curves and Surfaces", in Fundamental Developments of Computer Aided Geometric Design, L. Piegl (ed.), Academic Press.

6. Strauss, Paul S. and Rikk Carey. An Object-Oriented 3D Graphics Toolkit. Computer Graphics (SIGGRAPH '92 Proceedings) 26(2), pp. 341-350, July 1992.

7. Yoshida, Miko, Yuri A. Tijerino, Shinji Abe and Fumio Kishino, "A Virtual Space Teleconferencing System that Supports Intuitive Interaction for Creative and Cooperative Work", 1995 Symposium on Interactive 3D Graphics, Monterey, California, April 9-12, 1995, pp. 115-122.