COLLABORATIVE CONCEPTUAL MODELING USING THE SKETCH FRAMEWORK

JOSEPH J. LAVIOLA JR., LORING S. HOLDEN, ANDREW S. FORSBERG,
DOM S. BHUPHAIBOOL, and ROBERT C. ZELEZNIK

Brown University Site of the NSF Science and Technology Center
for Computer Graphics and Scientific Visualization
PO Box 1910, Providence, RI 02912 USA

ABSTRACT

This paper introduces NetSketch, an application that supports distributed conceptual design by providing tools for modelessly creating, manipulating and viewing 3D models in a shared virtual space. Inherent problems exist with collaborative design tools because of the simultaneous group interaction required for users to smoothly and effectively work together in the same virtual space. With NetSketch, we provide solutions to these problems by providing a fast and direct gesture-based user interface, a set of visual effects that better enable a user's awareness of operations done by other participants, and a set of tools for enhancing visual communication between participants.

KEYWORDS: Collaboration, 3D Modeling, Gestural Interaction, Visual Coherency, Ambient Cues

INTRODUCTION

Conceptual design involving geographically distributed participants requires not only tools that support a range of 3D model interactions but also techniques for resolving the inherent complexities of remote group-based model interaction. Currently, distributed design is based on conventional communication media like audio and video teleconferencing systems, speaker phones, and electronic mail. Although these media are valuable components of distributed collaboration, they do not address the critically important facility of enabling each participant to simultaneously create, modify, annotate and view a shared 3D model with other members of the design team. In fact, distributed design frequently degenerates into co-located design, when all team members travel to the same location in order to capture features that are not readily expressed with the traditional communication media. Therefore, what is required to help facilitate distributed collaborative modeling for conceptual design is a tool that will allow a group of participants to interactively design 3D models simultaneously while at physically separate locations.

This paper presents a 3D design tool, NetSketch, that allows small groups of remote collaborators to simultaneously interact with a virtual shared 3D model. In order to develop practical solutions, NetSketch addresses both fundamental issues relating to how multiple participants share a common 3D model, and also real-world problems associated with network limitations. In particular, NetSketch uses visualization techniques that not only mask network latency but also allow participants to identify and resolve object contention with each other. NetSketch further provides a range of group interactions that foster direct communication between individuals for purposes of feature highlighting, model annotation and attention focusing. Lastly, NetSketch utilizes ambient environmental cues to enable indirect sensing of the actions of other group members. When a users threshold for distraction is low enough, users will perceive information from these cues[1].

PREVIOUS WORK

Previous work in this area has been done in both 2D and 3D application domains. Ishii's ClearBoard system allows two remote participants to draw on a shared 2D drawing space for collaborative design[2][3]. This system has the advantage of having both users be able to visually see what the other is drawing and make eye contact due to a video steam of each participant superimposed on the drawing area. The drawback of this system is that it only supports two users and has no means to support 3D modeling. Another system that was developed for remote collaboration was LiveBoard[4]. LiveBoard is a large-area interactive display system that allows both remote and co-located participants to interact in a distributed design session This system also does not support 3D modeling tasks.

Kress's Shared 3D Viewer is a system that supports collaborative discussion of 3D models, specifically from CAD applications[5]. It helps to support communication in the design phase by allowing for shared camera views, three dimensional cursors and annotations to allow participants to discuss details of a product. The major problem with this system is that it only allows for viewing and annotating of 3D models and does not allow users to create or modify them.

Collaborative conceptual design often requires that participants visualize and understand changes made by other users in the design environment in real-time. One problem with obtaining this global environment awareness is the inherent network latency that accompanies essentially all networked applications. Recent work in this area has shown that there are a number of techniques that help to network lag. Conner's work uses various visual techniques, such as motion blurring[6] and transparency to help hide the effects of latency when users are manipulating objects[7]. Sharkey uses a perception model which ensures that each user views a "continuous version of the environment" such that object movement does not suffer from sudden jumps produced by lag[8].

We developed masking techniques for the related problem that arises when multiple people with different viewpoints interact with objects. While Conner's and Sharkey's work address visualizing object movements that are hindered due to system network lag, we address the issue of visualizing objects that disappear from view (because they were deleted, moved a large distance, etc.).

COLLABORATIVE 3D MODELING ISSUES

When creating a collaborative 3D modeling application, there are inherent issues that must be addressed to make the tool intuitive and efficient. A collaborative 3D modeling application's interface must be both fast and direct. We believe that the Post-WIMP interface metaphor will give users the speed and directness that the application requires[9]. Specifically, a gesture based interface allows for fast interaction because it does not require users to have to constantly move between a traditional WIMP(windows, icons, menus, and point and click) interface and the model being designed. A gesture based interface also allows users to directly perform the intended tasks and operations. This directness not only increases interaction speed, but also allows participants to keep their focus on the design session so they can remain aware of what others are doing in the design environment.

Although having a fast and direct interface is important in collaborative applications, there are still many important distributed interaction problems that must be addressed. Some of these problems are generic in that they apply to both 2D and 3D collaborative applications and some are inherently specific to 3D collaborative tools. The basis of these generic problems is how to show participants that something in the virtual work space has changed. Showing other participants that an object has been created, deleted, or moved quickly must be done so that users can have cues to perceive and derive information about the changing environment[7]. Secondary visual effects that both smooth the jarring and also provide a comprehensible record of user actions are a useful method of providing such cues[1]. Without these ambient cues, objects would simply appear or disappear with no warning that the environment is about to change. Another important issue is what happens when two or more participants try to interact with the same object. A mechanism must be created that allows a participant to freely interact with a given object and not affect what other participants are trying to do with the same object.

Collaborative 3D applications lend themselves to a unique set of interaction difficulties. Because 3D models and scenes can be too large to see in a single view, users usually have to move around the virtual world, by methods of rotation, zooming and panning. This camera manipulation can give participants different viewpoints when discussing a 3D model. In many instances, communication can break down when one participant tries to describe certain aspects of a 3D model when both participants are looking at two different views. As a result, viewpoints must be coordinated to enhance communication. Another issue that stems from a collaborative 3D application, is the ability to enhance the design discussion through the use of annotations. Annotations in 3D are difficult to create and use because as viewpoints change, the annotation must maintain its connection with the object that it is attached to.

SYSTEM BACKGROUND

Object interactions in NetSketch are derived from the gestural interaction metaphor presented in the SKETCH system[10]. SKETCH's interface is based solely on a 3 button mouse (no menus or 2D interface widgets are used) that allows users to draw lines on the image plane of a 3D view that the SKETCH system interprets as geometric operations and primitives. This metaphor is appropriate for collaborative environments because it allows 3D geometry to be manipulated at the speed of conversation. The cost of this style of interaction as compared to CAD systems is that the functionality of NetSketch is more limited, and some operations are less precise. However, these costs are acceptable in collaborative environments where the most critical issue is that communication bandwidth be maximized. For example, people are able to communicate effectively with hand gestures or scribbled drawings since they are able to generate multiple iterations of their "models" and are able to offload the missing "model" information to other modalities, especially speech.

EXTENSIONS FOR COLLABORATION

In this section, we discuss the various solutions to the 3D collaborative modeling issues that were discussed in the "COLLABORATIVE 3D MODELING ISSUES" section.

We discuss the visual effects we explored for object creation and deletion. We talk about our preliminary work on object contention. We then talk about how we let users know that an object has moved quickly by a remote user. We discuss our viewpoint control mechanism, and finally we talk about the types of annotations we use to enhance communication.

Typically, if a remote participant creates or deletes an object in a shared world it instantly appears or disappears. In NetSketch, these operations are animated over time to give a participant “ambient cues” to show something is changing in their shared environment. We experimented with various types of visual effects for object creation and deletion. One effect is a fade in upon creation and a fade out upon deletion. This effect is done by varying the objects transparency over a given time interval (typically 1 second). Another type of effect we explored is object scaling upon creation and deletion. This effect is created in a similar way to the previous effect except, in this case, we vary the objects transformation matrix over a given time interval by changing the scaling parameters of the matrix. We found that both visual effects work well in smoothing the jarring motion of objects that simply pop in or pop out of existence in the virtual environment.

Object contention is an important part of collaborative modeling systems. In NetSketch, we chose to allow multiple users to interact with the same object in order to prevent bottlenecks that typically occur with strict lock-based mechanisms[11]. When multiple users grab the same object, NetSketch automatically detects this conflict and seamlessly creates personal copies of the object for each user. A user’s personal copy of an object can not be modified by other users. However, a user can see the personal objects controlled by other participants. Because each user can modify their personal copy of an object, there must be some resolution regarding the appearance of the object after the release of all personal copies. In NetSketch, the shared version of the object is replaced with the personal object owned by the last user to release their personal object. Frequently, modifications to a personal object are temporary and users do not mind if they are lost. However, in other situations multiple users may want to keep their changes. Instead of some algorithmic form of resolving this situation such as “merging” the changes together, NetSketch replaces the original shared object with one of the user’s personal copy. The users must come to an agreement through spoken communication as to which personal object will persist.

It is important that a participant have an idea for what has happened to an object when it is modified by another participant. This situation can occur when participants move or delete an object. As described above, deleting is visualized by fading an object out over a period of time and is accompanied by a “delete sound”. In NetSketch, we have experimented with various visual effects that provide “ambient cues” for users to better track the locations of quickly

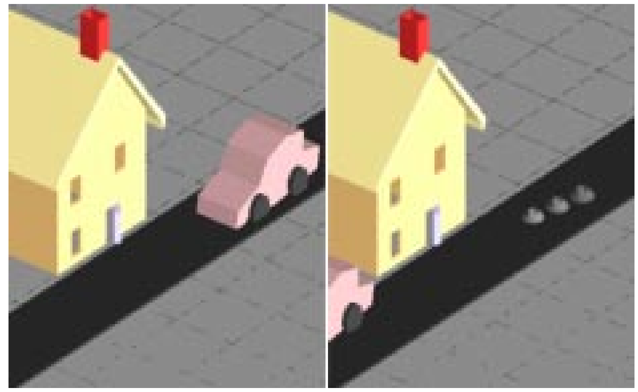

Figure 1: These to images show the smoke puff effect. The image on the left shows the car before it is moved. The image on the right shows 3 smoke puffs that the car left behind due to its quick movement.

moving objects. This first effect presents users with a ghost-like transparent copy of the given object that moves in the direction of the real object. After a brief time interval, the transparent copy will appear and follow the real object. The second effect produces streaks of lines from the initial position of the object to where the object gets placed. These lines give the user cues on the direction in which the object disappeared. Lastly, we created a “smoke puff” effect (see Figure 1). When an object is moved quickly, smoke puffs are created in its path similar to the techniques shown in cartoons.

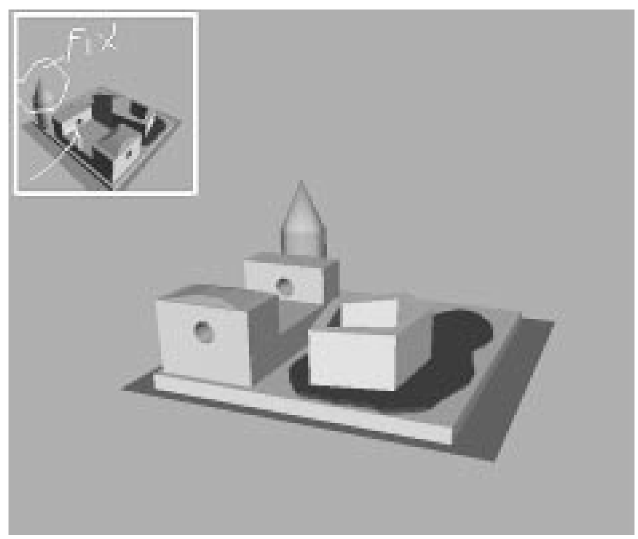

Figure 2: A remote participant of the design group. has created a view lens showing his/her viewpoint. The user also annotates his/her view lens to enhance communication.

When participants are discussing a given 3D model, it is important that they are able to share viewpoints when necessary. In the real world, this is similar to asking a collaborator to physically look over your shoulder to view a feature from your point of view or to turn the object away from yourself so that they can view the feature. In NetSketch, we take a slightly different approach to coordinating viewpoints. In NetSketch, users can show other participants their respective viewpoints by invoking a command that creates a magic lens on the screen of the other participants. This view lens, implemented like a tool glass[12], looks like a picture-in-picture. To invoke view lenses on remote participants screens, a user simply hits a shortcut key. View lenses can also be used to present a zoomed in viewpoint by drawing a lasso around the area of interest and invoking the view lens.

To better enhance communication, NetSketch provides two annotation mechanisms that allow users to point out particular features in a 3D model. The first mechanism allows participants to make 2D drawings in the view lens of another participant (see Figure 2). This allows users to describe and annotate parts of the 3D model using another user's viewpoint. The second mechanism is an annotation arrow. When a user draws a 2D gesture that looks like a arrow, a 3D arrow object is created and placed in the world. The tip of the arrow is positioned directly on top of the tip of the 2D drawn arrow. This 3D arrow can be moved around and will always snap to the normal of a given face of an object. This annotation tool allows participants to indicate a particular feature from any viewpoint, unlike the 2D drawn gestures above.

CONCLUSION

In this paper, we have presented NetSketch, a first step at making a collaborative conceptual 3D modeling tool. NetSketch provides tools for creating, manipulating, viewing, and annotating 3D models in a shared virtual space. The advantages of our prototype is that we provide the user a fast and direct gesture based interface for rapid creation and discussion of 3D models. The system also provides ambient cues that enable participants to understand what is happening in the virtual environment with respect to others. Because this application is a work in progress, there are some limitations. The application's object contention mechanism is very simple and does not take into account some of the more complex operations that a single user could do. The enhancement of this mechanism is an area of future work.

In order to minimize network latency and because we target groups of less than six people, we chose a peer-to-peer network topology instead of a hierarchical topology or client server model[13]. A peer-to-peer topology is easier to implement but makes it harder to guarantee consistency since system messages from one participant must be sent to all other participants. As a result, one of the major limitations of our prototype is that we cannot always guarantee consistency. Thus, as we continue to develop this application, we plan to experiment with various consistency control mechanisms such as dead reckoning and clock synchronization to guarentee consistency. Another area of future work is the types of visual effects used. We plan to both improve with our existing visual effects and develop new ones to find better ways to help participants understand what others are doing in the design environment.

REFERENCES

[1] H. Ishii and B. Ullmer. Tangible Bits: Toward Seamless Interfaces Between People, Bits, and Atoms. *Proceedings of CHI'97 Human Factors in Computing Systems*, 1997, 234-241.

[2] H. Ishii and M. Kobayashi, ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact. *Proceedings of CHI'92 Human Factors in Computing Systems*, 1992, 525-532.

[3] H. Ishii, M. Kobayashi and K. Arita, Iterative Design of Seamless Collaboration Media. *Communications of the ACM*, 37(8), 1994, 83-97.

[4] R. Elrod, R. Bruce et al. Liveboard: a Large Interactive Display Supporting Group Meetings, Presentations, and Remote Collaboration. *Proceedings of CHI'92 Human Factors in Computing Systems*, 1992, 599-607.

[5] H. Kress and B. Anderson. HP Shared 3D Viewer. *Ein verteiltes 3D Visualisierungssystem fur Produktdaten*, unix/mail, Carl Hanser Verlag Munchen, 1996, 329-334.

[6] R.C. Zeleznik and M. Wloka. Interactive Real-Time Motion Blur. *Proceedings of CGI'94:Insight Through Computer Graphics*, 1994, 74-85.

[7] B. Conner and L. Holden. Providing A Low Latency User Experience In A High Latency Application. *Proceedings of 3D Symposium on Interactive 3D Graphics*, 1997, 45-48.

[8] P.M. Skarkey, M.D. Ryan and D.J. Roberts. A Local Perception Filter for Distributed Virtual Environments. *Proceedings of Virtual Reality Annual International Symposium* 1998, 242-249.

[9] A. van Dam. Post-WIMP User Interfaces. *Communications of the ACM*, 40(2), 1997, 63-67.

[10] R.C. Zeleznik, K. Herndon and J. Hughes. SKETCH: An Interface for Sketching 3D Scenes. *Proceedings of SIGGRAPH'96*, 1996, 163-170.

[11] M. Macedonia and M. Zyda. A Taxonomy for Networked Virtual Environments. *IEEE Multimedia*, March 1997.

[12] E.A. Bier et al. Toolglass and Magic Lenses: The See-Through Interface. *Proceedings of SIGGRAPH'93*, 1993, 73-80.

[13] T.A. Funkhouser. Network Topologies for Scalable Multi-User Virtual Environments. *Proceedings of the Virtual Reality Annual Interntional Sysmposium*, 1996, 222-228.