

When Worlds Collide - Interactions between the Virtual and the Real

Sandy Ressler Brian Antonishek
Qiming Wang Afzal Godil Keith Stouffer

{sressler | antonis | qwang | godil}@nist.gov Information Technology Laboratory
kstouffer@nist.gov Manufacturing Engineering Laboratory
National Institute of Standards and Technology

Abstract

This paper explores issues surrounding interaction with virtual (computer generated) objects which are interfaced to real world devices. In addition a virtual room corresponding to a real physical room has been created to allow collaborative meetings with those physically and virtually present.

In particular we have created a set of VRML (Virtual Reality Modeling Language) cameras, which interface to real computer controlled cameras in a “smart” room. Interacting with the virtual camera controls the real camera. A second interaction method is through a Java applet, which appears on a Web page as an image of the remote control for the camera. The VRML camera object can also function as a status display. When someone moves the camera in the real world via the real remote control, the position of the VRML camera updates to reflect the actual status.

We have two methods of viewing the video from the camera, one a C program executing on the CPU with the camera which quickly updates the images that can be viewed on the web via a server-push HTML page. The second method is to stream the video through video streaming software (commercial off-the-shelf).

The entire VRML view of the “smart” room with cameras is placed within the context of a multi-user VRML world. This world can be visited by whomever desires to participate in meetings, held in the physical “smart” room. Our intent is to create intuitive user interfaces to the cameras, and other facilities of the room. We allow people not physically able or willing to attend the physical meeting to participate “virtually”. People attending meetings remotely are represented as avatars.

The room and objects in it are used for control and status. Depending on which, the dialog between the user and the displays change. When controlling real devices feedback must be rapid and sufficient to allow meaningful control. When using the virtual devices of the entire room only for status the user is more passive and can focus on other aspects of the meeting such as the actual content of the talk while still getting a feeling of the physical nature of the meeting space.

Additional Keywords and Phrases: virtual environments, user interfaces, device control, VRML

1 BACKGROUND AND RELATED WORK

The Virtual Reality Modeling Language (VRML) is a recently ratified ISO standard (ISO 14772) [8] file format for the description of geometry and behavior of 3D computer graphics. VRML is the most widely used open file format for integrating 3D computer graphics with Web based content. We use VRML as the primary interface technology for the representation of interactive virtual worlds. Our motivation is simply to use a widely available 3D technology that is already deployed throughout the general Web community.

Our VRML work is part of a larger Systems Integration for Manufacturing Applications (SIMA)[7] program at the National Institute of Standards and Technology (NIST) performing research in advanced manufacturing and the applications of information technology. VRML provides a robust foundation upon which one can build 3D applications that are portable and distributable via the World Wide Web.

There has of course been significant research performed on the use of a variety of devices and integration techniques relating such environments as offices and auditoriums. The 1998 AAAI Spring Symposium on Intelligent Environments[1], contains a rich selection of papers on similar topics. These papers cover the use of individual inputs such as vision, spatialized audio as well as inferred input via face detection and body position.

Our work differs from these systems by the use of the same 3D environment both for status and control of real world devices. In addition our use of VRML is based on the desire to create collaborative meeting functionality of the existing infrastructure of the Web. This particular mixture of real and virtual devices leads to innovative and sometimes nonsensical results. On the innovative side, we can use a real physical remote control to control virtual objects. Nonsensical control occurs when virtual cameras track avatars while controlling a real world camera, watching nothing.

2 SYSTEM OVERVIEW

The overall system consists of a typical Web client/server arrangement. The end user interacts through the framework of a Web browser, with some additional 3D (VRML) plug-ins, or Java applets. Figure 1 illustrates the user’s view of one version of the system.

DISCLAIMER: Mention of trade names does not imply endorsement by NIST.

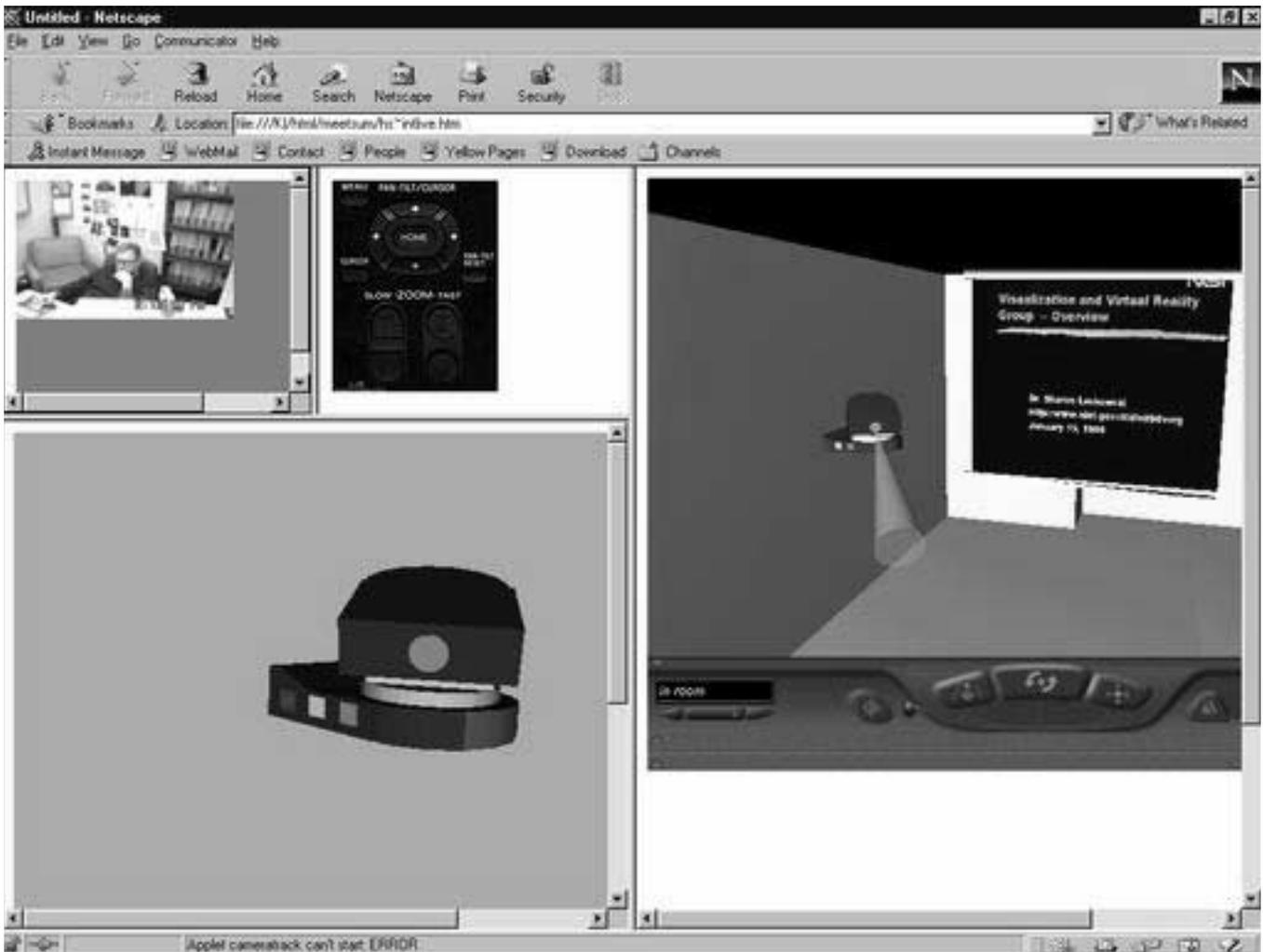


Figure 1. Live Video Stream with Camera Control and Status Displays

The user can control the camera either with the Java applet presented to the user as simply an image of the camera's remote control, or by moving the 3D VRML version of the camera. The video displayed in the upper left corner of the display is streamed.

The camera server must run on the same physical machine as the Web server because of the security restrictions imposed by the use of a Java applet. Our camera server is quite robust and interfaces both with the VRML interface and the Java applet, and can handle cameras on multiple servers.

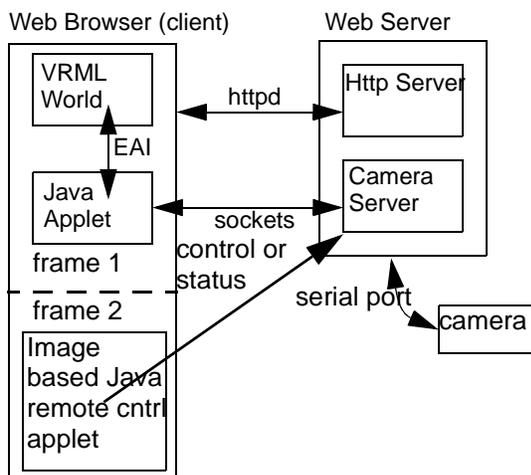


Figure 2. Camera Interface Architecture

3 CAMERA INTERACTION

3.1 VIRTUAL CAMERA FOR DELAY MITIGATION

One of the methods we use to display live video is to stream it with a RealMedia server running on the same machine that hosts the camera. The RealMedia method is suitable for Internet viewing but usually has fairly slow frame rates.

The streaming method introduces a new problem, a long delay due to a 30 second buffer created to enable the video stream. All streaming systems create some sort of buffer from which the display is created to remove network latency. For the usual application of live video a 30 second buffer doesn't matter, however for

interactive live video the buffer is problematic. If you push on the control button to move the camera it takes 30 seconds until you see the result. Not terribly satisfactory.



Figure 3. Camera and Remote Control

In order to compensate for the large delay from streamed video a new interaction method is introduced. We use the VRML camera for feedback. The position of the camera is displayed via the virtual camera instantaneously. As the user moves the camera by clicking on buttons of the Java applet the virtual camera moves showing the user the position of the real camera. After the 30 second delay the image is updated. While not nearly as satisfactory as instant image updates this method does however provide useful feedback for the user.

We can further enhance the feedback provided by the virtual camera by displaying a cone or ray from the camera. This additional visual cue gives the user a better idea of where the camera is actually looking in the room (albeit a virtual room).

3.2 CAMERA TRACKING

The Sony camera (EVI-D30) has a built in tracking capability. The camera is simply pointed at a particular area, of hopefully distinctive coloring, and it attempts to keep that area in the center of the camera. It is particularly interesting watching the VRML status camera move according to the motion of a tracked individual.

In Figure 4, the white rectangle in the center of the screen tracks the subject as he moves around the room. Depending on the camera mode the camera attempts to keep the tracked subject in the center of the screen. Areas of high contrast appear to work better however the camera does function surprisingly well.

One obvious arrangement is to have the camera track a real person and create an avatar in the shared virtual world of the real person. Unfortunately this is somewhat problematic with our current hardware as we can not get any Z values for the distance from the tracked subject to the camera. This can be remedied either with additional hardware or by computing a Z value by triangulating from tracking values of 2 cameras. It remains a task for future work.



Figure 4. Camera Display of Tracked Area

3.3 VIEWING VIDEO

We have two methods of viewing the video from the camera, one a C program executing on the CPU with the camera which quickly updates the images that can be viewed on the web via a server-push HTML page. The second method is to stream the video through a RealMedia Server [4].

The server push method provides approximately a 10-15 frame-per-second rate however the high data transfer is not suitable for Internet access. The RealMedia streamed video is of significantly lower quality but is usable over the Internet. As discussed earlier the streamed video also introduced a long delay due to the buffering required to achieve smooth video over the Internet.

4 Blended Reality

Some of the more interesting issues arise from mixtures of real world devices and the virtual world analogs. We call this new type of environment "Blended Reality." Blended Reality (BR) mixes the use of real physical devices with a virtual representation of those devices and/or the spaces they occupy. A BR system integrates many devices with virtual representations which can be modified versions of the real device. Camera's can have additional controls placed on them in the virtual world. Virtual representations of objects can be made to control and display the status of real-world physical devices. This blending of reality and virtual representations offers users new ways of interacting with information appliances.

4.1 REAL REMOTE CONTROL of VIRTUAL WORLD

One twist in the collision between the virtual and the real is the use of a real remote control device to control objects in the virtual world. Recall that we have a virtual camera presenting the status, (the position) of a real camera. The real camera can be controlled via it's remote control. If we simply ignore the camera (i.e. pay no

attention to the man behind the curtain) and look at the virtual camera we are, in effect, controlling the virtual camera with the real remote control.

Let's take this one step farther. It would be useful to be able to pick which of several virtual objects we are controlling via the real remote control. All that is needed is some mechanism for selecting which object to control. In our particular case the real remote control has buttons for preset camera positions, numbered one through six. We can use the buttons as a generic signal to the VRML world to conveniently select between any of six objects to control.

One more step into this realm generalized the button pushing of the remote control to any (of the six available) touch sensors we use for the control of a virtual world. For example in the VRML work for our manufacturing project[5] we set up a small set of controls as a HUD (heads up display). These controls are a set of spheres and a cube that float in the lower left corner of the virtual world. Touching the spheres or cube causes a humanoid figure in the scene to start several actions. We can set this up as a set of remote control buttons that cause the actions to occur. A much more usable method for giving presentations of the manufacturing demo.



Figure 5. Remote Control of a VRML World (real camera out of picture)

We can implement the concept of real remote controls for virtual worlds via another direction, that of a flexible programmable remote. The PalmPilot III contains an infrared output device which can be programmed[3] to learn the infrared signals from any remote control device. We simply train the PalmPilot on the remote control of interest and voila we have a universal remote control interacting with our virtual world.



Figure 6. PalmPilot Remote Control

Taken to it's logical conclusion the display on the PDA could be another 3D object which we can manipulate. The utility of this is, yet again, unclear.

4.2 VIRTUAL WORLD CONTROLLING REAL WORLD

In our attempt to make a virtual world as "realistic" as possible some interesting actions arise. For example as we use the real camera to track a real person and display the virtual camera the reverse seems natural. An avatar in the virtual world, is tracked by the virtual camera, and we can control the real camera accordingly. Of course it makes no sense to have the camera track a virtual ava-

tar in the real world, there is nothing there! It does however have a reality blurring effect to sit and watch the camera panning across the room tracking some avatar.

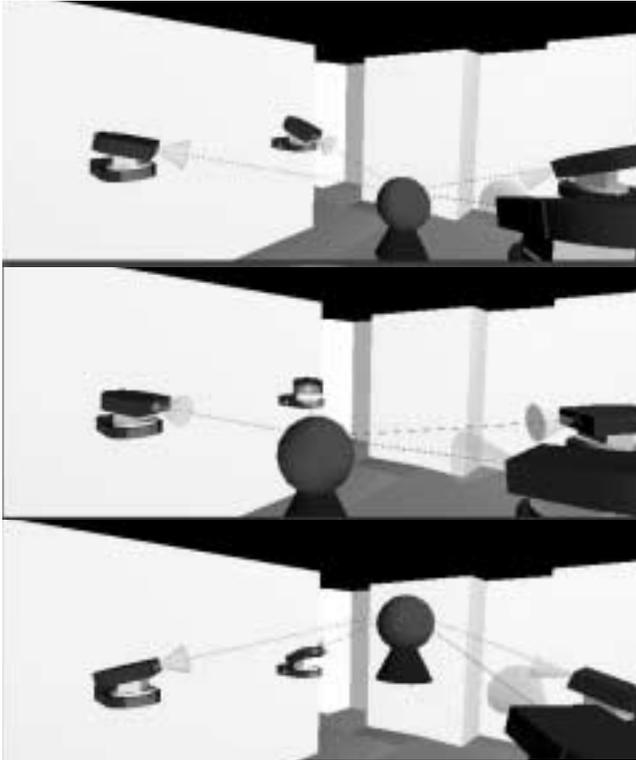


Figure 7. Virtual cameras tracking avatar and controlling real camera.

We must ask is there any value to a real camera tracking a non-existent object. Probably not. If however we were to move a projected image or some other visual or auditory indicator users could get the “feeling” of the motion, presence and activities of other meeting participants. This would aid in the creation of subconscious implicit knowledge of the meeting participants, perhaps a valuable aspect.

4.3 SHARED VIRTUAL WORLDS

We are using the blaxxun multi-user server and client software[2] to create a shared virtual world for a collaborative meeting space. We have initially taken the VRML version of the “smart room” and made it a multi-user world. Meeting participants appear as avatars in the virtual space. One issue, do we simply place the avatars in chairs and force them to be well behaved or do we let people wander around the virtual room?

In addition to the world itself there are a number of challenges to delivering a shared space. The objects in the space may be interactive or in our case may provide status and control of real-world devices. We are using the shared event capability built into the blaxxun server to create worlds where all participants see the same actions.

When one user moves the camera, all see it move. Similarly when the camera position is updated due to a change in the position of the real camera all users see the changes.



Figure 8. Multiuser shared world with controlling camera and avatars.

An additional problem, when dealing with device control, is allocating the resources in a manner acceptable to the users. Users must be told they have control of a particular device, or that the device is busy and being used by another user. We have added some buttons to the base of the camera for our additional controls and are still experimenting with feedback mechanisms.

In an ideal implementation the virtual world would represent both the position and actions of real people in the “smart room” and people wishing to attend the meeting, “virtually”. One advantage of this type of configuration is to allow collaborators access to several spaces during a short period of time in which non-of the real spaces would allow colocation.

For example in the manufacturing applications we are exploring, a welding environment which is the analog of a real welding cell which can be visited “virtually” by any of the meeting participants. Simply walking out one door of the virtual smart room leads to the welding cell. This is not a mirror of reality where the weld-

ing cell is in fact in a different building from the smart room. The ability to dynamically reconfigure spaces for the topic of the day is one advantage of the virtual meeting space.



Figure 9. Welding Cell from Virtual NAMT in Multiuser Environment

The VRML weld cell contains a model of a Robotics Research Corporation (RRC) robot that is controlled by the real world robot controller. This is accomplished by a socket connection between the client's web browser and the real world controller located at NIST. The current joint angles of the robot, which are stored in a world model buffer in the controller, are collected by a Java applet on the web page. The applet then updates the VRML model of the robot via the External Authoring Interface (EAI) of the VRML plug-in.

Development is currently underway to create dynamic weld sections (VRML cylinders) based on the current robot position and the calculated weld quality of a section of weld obtained from the real world controller. A VRML section of weld is created every 100ms via the EAI. The color of the VRML weld is based on the quality of the weld in that 100ms interval. The VRML weld is colored gray for a good section of weld and red for a bad section of weld. This allows a client to visually determine where a bad section of weld has occurred without being present in the physical welding lab. For this application the VRML world functions as a status and data visualization display. Collaborative meetings after the fact can take place in the virtual welding cell to discuss the results.

5 CONCLUSIONS AND FUTURE WORK

We have demonstrated the integration of real and virtual worlds exhibiting novel properties. One outcome was the use of real world devices to control virtual objects. The use of these devices reduces the explicit nature of the user interface reducing the cognitive load on the user. The user just grabs the remote control and can manipulate the object. In addition we believe that providing blended reality configurations provides users with

additional information about the environment without overwhelming them with information. The use of a virtual device representation provides immediate feedback and helps mitigate some of the delay problems for our streamed video.

Clearly we have raised many more questions than we have provided answers. In the future we will develop closer integration with the welding cell and conduct actual meeting gathering user feedback about the experience. We also will be establishing a virtual world which will remain "open" as a resource in which people can experiment with the camera controls. .

6 ACKNOWLEDGEMENTS

We would like to thank the continuing sponsorship of the Systems Integration for Manufacturing Applications (SIMA) program for supporting this work.

References

- [1] AAI Spring Symposium on Intelligent Environments, 1998. <http://www.aaai.org/Symposia/Spring/spring-symposia.html>
- [2] Blaxxun Interactive <http://www.blaxxun.com>
- [3] Pacific Neo-Tek OmniRemote <http://www.pacificneotek.com/download.htm>
- [4] RealNetworks <http://www.real.com>
- [5] S. Ressler, A. Godil, Q. Wang, G. Seidman. A VRML Integration Methodology for Manufacturing Applications *Proceeding VRML99 Fourth Symposium on the Virtual Reality Modeling Language*, Paderborn Germany, pages 167-172, 1999.
- [6] Sony EVI-D30 Operating Instructions.
- [7] Systems Integration for Manufacturing Applications (SIMA) Program <http://www.nist.gov/sima>
- [8] VRML. *VRML 2.0 Specification ISO/IEC CD 14772*, 1996.