

Making VRML Accessible for People with Disabilities

Sandy Ressler
National Institute of Standards
and Technology
Bldg. 220, Rm. A216
Gaithersburg MD 20899
sressler@nist.gov

Qiming Wang
National Institute of Standards
and Technology
Bldg. 220, Rm A216
Gaithersburg MD 20899
qwang@nist.gov

1. ABSTRACT

This paper describes a set of techniques for improving access to Virtual Reality Modeling Language (VRML) environments for people with disabilities. These range from simple textual additions to the VRML file to scripts which aid in the creation of more accessible worlds. We also propose an initial set of guidelines authors can use to improve VRML accessibility.

1.1 Keywords

VRML, virtual environments, navigational aids, accessibility, audio feedback, data access, speech input, user interfaces.

2. INTRODUCTION

In the introduction of a special section on Computers and People with Disabilities in *"Communications of the ACM"* [1] the authors point out:

"When one looks at the data, it is surprising to see how large a segment of the population of the United States has some form of disability. In 1990 the National Science Foundation formed a Task Force on Persons with Disabilities to determine how NSF could best promote programs in this area. As this report points out, over half a million Americans are legally blind - this means that their visual acuity, with correcting lenses, is no more than 20/200 in the better eye. Furthermore, of the approximately 5,000,000 scientists and engineers in the U.S., it is estimated that as many as 100,000 have some form of physical disability."

The world is becoming ever more interconnected and dependent on information available on the Internet. The use of Virtual Reality Modeling Language (VRML) [2] worlds as a mechanism for information access is a breakthrough. Access to that information for all people is critical. The challenge is to develop methodologies and guidelines for

making VRML accessible. VRML, now a draft international standard and what some have called the 2nd Web, opens up new possibilities but also erects new barriers. In this paper, we propose some guidelines, develop some techniques, and provide some tools to help create more accessible VRML worlds.

In general, a modest amount of additional work must be done in order to make the VRML world accessible. However, it is our experience that in making the worlds more accessible for people with disabilities, the worlds will be more usable by all. For example, in the course of conducting this research, we came upon the description field of the Anchor node. Adding the descriptions, for our VRML Miter Saw, solved a long standing annoyance of not clearly seeing what object the cursor was on. The additional text, now part of the description field, is displayed by the browser. The improved accessibility of our VRML world improved the user interface for all people.

3. BACKGROUND AND RELATED WORK

There are many types of disabilities, indeed it has been said that *"we are all disabled, it is just a matter of degree."* [3] Visual impairments, hearing loss or deafness, motor control impairments, speech impediments and cognitive disabilities are all problems for those afflicted. Different media types such as audio, graphics, and video can improve access to information depending on the particular disability.

As part of our ongoing research at NIST [4], we have examined the use of virtual environments for information access. Even in the early days of the Web, compelling demonstrations, by companies such as Papersoft (now Netscape) which let users obtain news from anywhere in the world by selecting geographic areas of a globe, illustrated a vision of how to provide intuitive access to massive amounts of information. More recently the ability to explore the Martian terrain, along with the Pathfinder, became a VRML information resource [5]. However what is a blind person to do if the information is important and only accessible via these worlds?

Graphical User Interfaces, the GUIs of the mid 1980s and 90s, have shifted the user's interaction with computers from a primarily text-oriented experience to a point-and-click experience. This shift, along with improved ease of use, has also erected new barriers between people with disabilities and the computer. There have been a variety of devices and software techniques that have been developed to improve access. The concept of auditory icons has been pursued for

over ten years [6]. Devices which are used to make PCs more accessible range from speech synthesizers and screen readers to magnification software and Braille output devices. A thorough collection of disabilities resources can be found at the WebABLE! [7] web site.

There is some concern over access to VRML for people with disabilities. However, it is quite scarce, with the most notable exception being the work of Treviranus and Serflek at the University of Toronto's Adaptive Technology Resource Centre (ARTC) and its web page "Accessibility and VRML" [8]. Many of the concepts such as aural introductions, and the use of embedded textual descriptions were addressed by their work.

4. TAXONOMY OF ACCESS TECHNIQUES

We propose the following VRML mechanisms as a starting point for improving access. These mechanisms fall into three categories: textual descriptions, audio cues and spoken descriptions, and keyboard input facilitation. All of these mechanisms use the inherent capabilities of the VRML specification to make the VRML world more accessible.

4.1 Textual Descriptions

Additional textual information about a VRML world may be added, in two different nodes, the WorldInfo and Anchor nodes. The WorldInfo node contains the label for the overall world and provides information about individual objects. A VRML browser such as CosmoPlayer uses the title field of the first WorldInfo node it encounters to place a label on a portion of the browser window. WorldInfo nodes should be placed inside of Groups to document individual objects with supplemental text. JavaScript functions may be defined which use the additional WorldInfo nodes to change the label displayed by the browser. The setDescription JavaScript method is supported by compliant browsers. VRML pre or post-processors, such as our speakWorldInfo utility, can perform additional actions with the text, such as speech synthesis.

Another place for additional text is in the description field of an Anchor node. This is a particularly useful way of adding value to a world because VRML browsers display the text from the description field when the cursor is placed over the object. In our miter saw example, illustrated in Figure 1, description fields enable the browser to display the descriptions of the parts comprising the saw.

The WorldInfo node is analogous to the ALT property of the HTML image tag, IMG. The ALT property, takes a textual string as its value. This string can be read by speech synthesizers or in some browsers, such as Internet Explorer, the text appears when the user simply holds the cursor on the image. Similarly VRML provides a general mechanism, via the WorldInfo node, for documenting and supplementing object information.

If browser builders would allow access and use the WorldInfo nodes associated with objects, then world builders would be encouraged to include more descriptive textual information in their worlds. One approach to using

WorldInfo nodes now, is to use some Java code via the External Authoring Interface (EAI) to read the contents of the WorldInfo node and display the text or send it to a speech synthesizer.

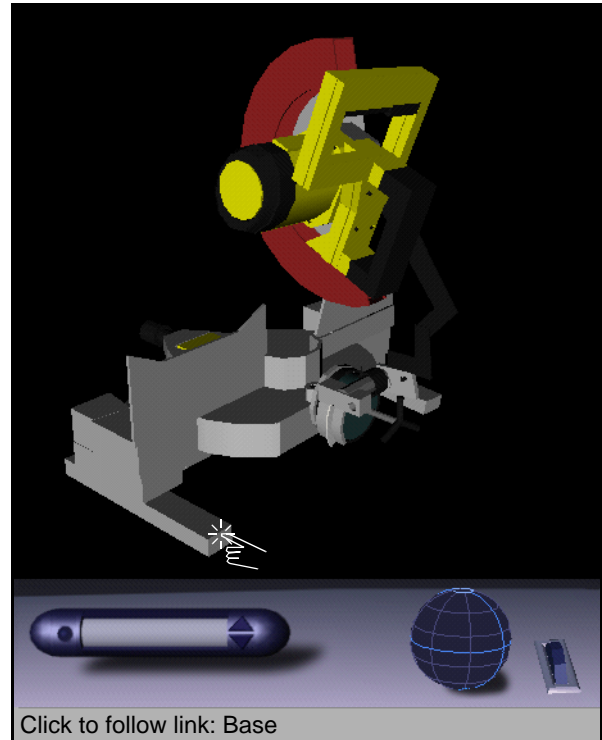


Figure 1. VRML world with WorldInfo description for object (note figure altered for printing purposes)

4.2 Audio Cues and Spoken Descriptions

Audio provides a set of rich capabilities to improve access to a VRML world. The three types of audio we examine here are: ambient background music, spoken descriptions, and speech synthesis. Ambient music can play and change as the user moves from one room to another, providing a subtle yet intuitive location cue. Spoken descriptions of objects can play as the viewer moves close to an object. Speech synthesizers can "read" embedded text. Given the availability of a speech synthesizer, text from Anchor node descriptions or WorldInfo nodes can be spoken. (We demonstrate this with our speakWorldInfo utility described in the section VRML Access Utilities.) Internet accessible speech synthesizers such as the accessibility Labs Text-to-Speech system [9] provide easy access to speech synthesizers.

One under-utilized capability is the description field in the AudioClip node. AudioClips contain the pointer to actual sound files and in addition the node contains a description field which can be used as a textual description of the sound. Unfortunately VRML browsers currently implemented do not take advantage of this information.

In addition to the sounds themselves the way in which a sound is started, the triggering mechanism is also important. Actions such as the start of speech, sound effects, or other external programs (such as a speech synthesizer or Braille printer) can be initiated based on different types of triggers. Three triggering mechanisms, supported by VRML are:

- **proximity - execution based on viewer position**
- **viewpoint - execute when selecting viewpoint**
- **touch - execute on viewer clicks**

Triggers define the events when a sound is to play. For example a Proximity Sensor is used to cause an event when the user is inside the area defined by the sensor. A sound describing an object plays when the viewer moves close to that object as illustrated in Figure 2.

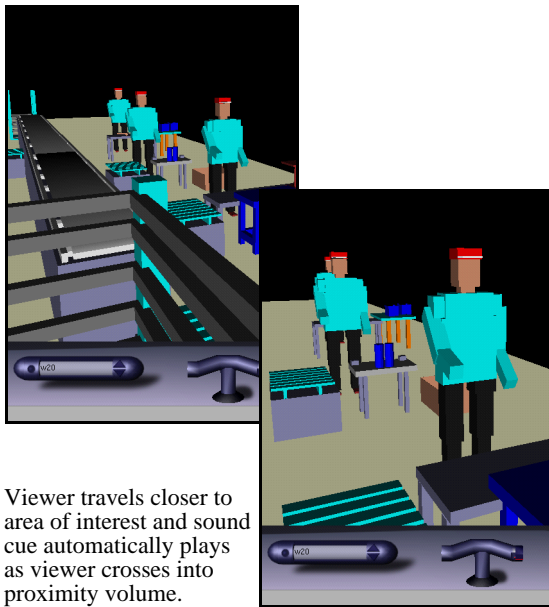


Figure 2. An overhead view of line with proximity sensors.

Triggers can be used for more than playing sounds clips. They can cause a speech synthesizer to read text, print Braille, and via programmed external utilities execute virtually any type of code. Triggers can initiate the execution of arbitrary code and direct devices to perform desired actions.

In Figure 3 we see, in an overhead view of the entire assembly line, the bounding boxes of the four proximity sensors. When the viewer's position enters any of these volumes the appropriate audio is triggered.

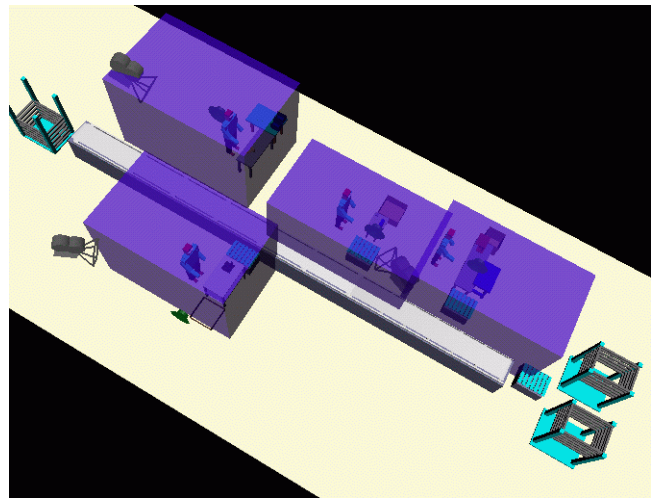


Figure 3. An overhead view of line with proximity sensors.

The audio in VRML is also spatialized a feature we have not taken advantage of yet. However sounds can be associated directly with object and triggered as deemed appropriate. Users, may be able to locate objects by sound alone. Figure 4 illustrates the sphere (which can actually be an ellipsoid) within which the audio is heard at full volume with a small icon in the center indicating direction. The proximity sensor surrounding that particular sound has been removed for the illustration, and the other proximity sensor volumes can be seen in the distance.

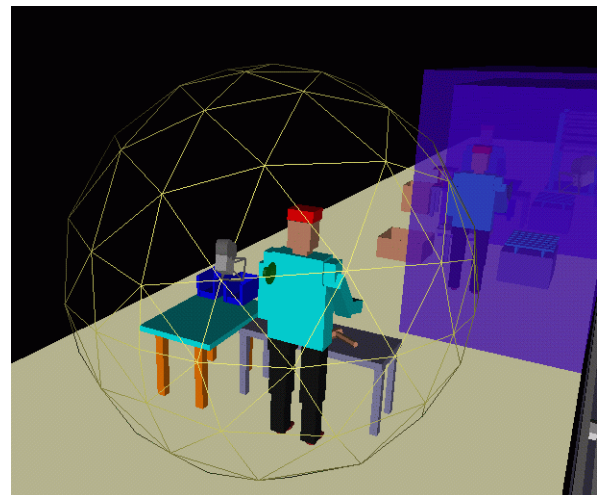


Figure 4. Spatialized audio with and proximity sensors.

4.3 Keyboard Input Facilitation

Keyboard mappings, the ability to perform application functions simply by using a keyboard, rather than a mouse, is an important enabling technology. VRML browsers provide some aid in this domain albeit minimal. A common

keyboard equivalence is to map the PageUp and PageDown keys to allow users to step to the next or previous viewpoint. Viewpoints, however, must be defined as part of the world an all to infrequent occurrence. In addition to viewpoint selection, the arrow keys can be used to rotate the object, when in examiner mode, and to travel, when in walk mode. The specific examples cited above are for CosmoPlayer; each VRML browser behaves slightly different. Consistent keyboard mappings and their subsequent behavioral effects in the VRML world can provide an important accessibility capability for a VRML browser.

Finally if a world will be used by individuals with motor impairments, such as no hand control, the world should have large control areas for easy access by alternate devices such as a voice controlled cursor. Again this type of interface is not only useful for people with disabilities, but for people with their hands busy, such as when driving a car or operating machinery. We successfully tried one such device the Dragon Dictate [10] voice operated cursor control and could select links and examine objects, albeit with some difficulty and ample patience.

5. AUTHORING GUIDELINES

As we have shown, there are several ways to make VRML worlds accessible by the visually and physically impaired. The addition of embedded text, sounds and assistive devices such as a speech recognition systems all contribute to more accessible virtual worlds. Web designers wishing to make their VRML worlds more accessible should:

- Add WorldInfo node descriptions both for the entire world and individual objects.
- Use the description field of Anchor nodes.
- Create Viewpoints because they are accessible via the keyboard PageUp and PageDown keys.
- Associate Sound nodes with spoken descriptions of objects of interest.
- Name significant objects to allow for automated sound hooks (as demonstrated with our addSndToVrml utility).
- Create large control areas for alternate input devices.

6. VRML ACCESS UTILITIES

We have developed several utilities to assist in the creation of accessible VRML worlds. (The source code for all of these are freely available to the public on our web site [11]) They are: showVP, addSndToVrml, and speakWorldInfo. Following are descriptions of each utility:

6.1 showVP

SYNOPSIS: showVP input.wrl

DESCRIPTION: Takes the **input.wrl** VRML file and adds a small sphere to the scene. Clicking on the sphere causes the browser description window pane to display the current Viewpoint location and orientation. These values can be used to define new ViewPoint nodes. This utility is intended for use as input to the addSndToVrml utility. The result is sent to STDOUT.

6.2 addSndToVrml

SYNOPSIS: addSndToVrml mapfile input.wrl

DESCRIPTION: This utility adds Sound nodes to a VRML file such that when the user travels from one ViewPoint to another the sound plays. The intent is to allow guided tour types of scenarios. The **showVP** utility can be used as an aid in the definition of the ViewPoint node parameters, alternately a VRML authoring system can also be used. The author is responsible for the creation of the sound files, the viewpoint location and to name the objects in the VRML file. The **mapfile** provides the association between the viewpoint and the sound file. Each row of the mapfile consists of an object name, sound file name and the seven values for location and orientation of a viewpoint. The result of running **addSndToVrml** is a new VRML file, sent to STDOUT (see Figure 5).

Five steps for adding sound files as access aids.

1. Create sound files.
2. Name objects and areas of interest.
3. Use showVP (or other tool) to determine Viewpoints.
4. Create mapfile.
5. execute addSndToVRML mapfile input.wrl > output.wrl

Figure 5. Steps for adding proximity triggered sound files.

The changes **addSndToVrml** makes in creating a new VRML file are: Creates new Proximity nodes at the Viewpoint location defined in the **mapfile**; Creates new Sound nodes that point to sound files named in the **mapfile**; Creates routes from Proximity nodes to the Sound nodes.

6.3 speakWorldInfo

SYNOPSIS: speakWorldInfo input.wrl

DESCRIPTION: This utility reads the input VRML file and looks for WorldInfo nodes. WorldInfo nodes may be associated with individual object, not simply the entire file. The utility creates a new VRML file, sent to STDOUT, in which the user may select the objects with associated WorldInfo nodes, and the browser sends the title field of the WorldInfo node to a speech synthesizer, via a URL mechanism.

7. EXAMPLES

We have created two examples of accessible VRML worlds. One, the *Audible Assembly Line*, is representative of an environment intended for “walk” mode. The other environment, *The Talking Miter Saw*, is intended for “examiner” mode. Both words are available at the OVRT web site [11]. In the case of the miter saw, object

descriptions appear on the browser's window because of the description field of the Anchor node. The name of the part being selected is spoken by passing the string to a speech synthesizer.

The Audible Assembly Line demonstrates the use of spoken descriptions. The user is immediately greeted with an introductory description. Each viewpoint, accessible via the PageUp and PageDown keys, has associated with it a spoken description of the workstation.

8. CONCLUSIONS AND FUTURE WORK

While we have discussed and illustrated how to create accessible worlds through the addition of audio content, VRML browsers should also be capable of accepting additional audio information. For example, the Anchor node in the VRML2 specification, contains a "parameter" field, intended for use by the VRML or HTML browser. The parameter field, an MFString, contains keyword=value pairs. One could easily define, a spokenText=text, pair which would instruct the browser, upon selection of the Anchor node, to speak the text. A more thorough discussion of browser issues is in the Serflek and Treviranus paper cited previously, and points out issues such as keyboard equivalences, and alternative input devices.

A future research agenda for VRML accessibility should address, at a minimum, the following issues:

- **Object selection by people with visual impairments**
- **Virtual environment navigated by the blind**
- **Sounds as navigational aids**
- **Utility of alternate input devices such as position trackers and gloves**
- **Authoring utilities to automate accessible virtual world production**

It is clear there are challenges and a variety of research issues to address. Starting with the guidelines outlined in this paper there is much that can be done today towards making VRML accessible.

9. ACKNOWLEDGMENTS

We would like to thank Mike Paciello of the Yuri Rubinsky

Insight Foundation (www.yuri.org) for his encouragement and support for these concepts. Thanks to Sharon Laskowski for her ruthless editing which improved this paper ten-fold. The authors would also like to acknowledge the continued support of the NIST Systems Integration for Manufacturing Applications (SIMA) Program for making this research possible.

10. REFERENCES

- [1] Gilnert, E. and York, B. Introduction to Special Section on Computers and People with Disabilities in *Communications of the ACM* Vol. 35, No. 5, 1992.
- [2] VRML. *VRML 2.0 Specification ISO/IEC CD 14772*, 1996.
- [3] Carl Brown "Assistive Technology Computers and Persons with Disabilities" *CACM* May 1992, Vol. 35, No. 5.
- [4] Sandy Ressler. Approaches using virtual environments with mosaic. In *The Second International WWW Conference '94 Mosaic and the Web*, volume 2, pages 853-860, 1994.
- [5] Mars Pathfinder VRML models
URL: <http://www.ksc.nasa.gov/mars/vrml/path-vrml.html>.
- [6] Gaver, W.W. (1986) Auditory Icons: Using sound in computer interfaces. *Human-Computer Interaction*, 2,167-177.
- [7] WebABLE!
URL: <http://www.yuri.org/webable/>
- [8] Chris Serflek, Jutta Treviranus "VRML: Shouldn't Virtual Ramps be Easier to Build."
URL: <http://www.utoronto.ca/atrc/rd/library/papers/vrml.html>
- [9] Tanenblatt, Bell Labs Text-to-Speech System web site,
URL: <http://www.bell-labs.com/project/tts/>.
- [10] Dragon Systems, Dragon Dictate Personal Edition,
URL: <http://www.dragonsys.com/>.
- [11] Ressler, The Open Virtual Reality Testbed.
URL: <http://ovrt.nist.gov>.