Exploring Annotated 3D Environments on the World-Wide Web

Enrico Gobbetti[1,2] and Russell Turner[1]

[1] UMBC, CSEE Department, Baltimore MD 21228-5398, USA
[2] CESDIS, NASA Goddard Space Flight Center, Greenbelt MD, USA

Abstract. The long-term goal of combining virtual reality and the Internet is to create networked multi-user simulations of virtual environments. The Virtual Reality Modeling Language (VRML) represents a limited but significant step towards this goal by creating a standard data file format for representing 3D scene information, together with hyper-link information for associating it with other types of Web documents. Current proposals for extending VRML-1.0 to add behaviors will bring this goal closer, but much work remains to be done. This chapter gives a brief summary of VRML and then describes two significant projects currently under development based on i3D, a high-performance VRML browser developed by one of the authors. The first of these, currently being used at the European Laboratory for Particle Physics (CERN), uses an annotated virtual environment to visualize and walk through the physical design of the new Lepton-Hadron Collider (LHC) before it is built. The second project, "Virtual Sardinia", allows the user to tour a 3D terrain visualization of the island and access historic and tourist information through hyper-links.

1 Introduction

The World-Wide Web (WWW) has rapidly become one of the fundamental structures of the Internet. By imposing a universal organization on the variety of formats in which data resides around the world and allowing each piece to be viewed as a uniquely addressable data source, it has allowed the entire Internet to be treated as a single structured document [12, 4]. This unified distributed database, which can be universally accessed using a single software Web browser application, takes the form of an enormous hyper-media document, combining text, images, sound and video into a seamless, hyper-linked user-interface.

While the multi-media user-interface style of the Web manages to incorporate all the standard forms of electronic mass-media, it has until recently been conspicuously lacking in one of the most important new modes of human-computer interaction. Paralleling the meteoric and highly-publicized rise of the Internet in recent years has been the development of "virtual reality" (VR): the most advanced type of user-interface in which all human sensory input is synthesized by the computer to create an illusion of immersion in a virtual world [5]. While technical obstacles remain which limit the quality of the fully immersive experience, advances in 3D graphics display and input technology have now made possible a higher-quality and more practical – if rather limited – form of "desktop" VR in which the user simply views an interactive 3D image rendered on a traditional color monitor. This configuration can then be augmented to various degrees of immersion by adding stereo viewing glasses, head tracking devices, 3D mouse input, localized sound and head-mounted displays. Even the most limited forms of virtual reality open up immense possibilities for us to change the way we interact with the computer. Virtual reality is more than a buzz-word. As a long-term goal, it represents the ultimate man-machine interface, as well as the most general-purpose medium of human expression.

1.1 Adding 3D Contents to the Web

It is therefore no surprise that a union between Virtual Reality and the Internet should be proposed. Given their respective enormous potentials and their complementary nature, VR and the Internet seem to be a natural match. A marriage between them presents tantalizing possibilities: cyberspace becomes a virtual 3D information space in which remote participants may immerse themselves. The World-Wide-Web is transformed into a world-wide virtual museum where every home page becomes a home room. Inside these rooms every door or picture on the wall is a potential "portal" or 3D hyper-link to another room in the museum, or to a piece of information in some other form. Inherently three-dimensional tasks, such as architectural or engineering walkthroughs, can be enhanced by adding traditional WWW-style information in the form of annotations. Novel ways of representing complex data sets, whether it be three-dimensional scientific visualizations, multi-dimensional financial data, or non-numerical textual information, allow the Web browser to become a data visualizer. By making queries to remote sites, which return custom 3D data on-the-fly, the Web browser becomes a graphical front-end for large distributed databases.

Perhaps the most significant product of the Internet/VR union may be the development of virtual environments with distributed interactive behaviors. Most virtual environment applications to date have been static scenes in which user interaction is limited to viewpoint motion, pointing and selecting. Our own real-world environment, however, is dynamic. It contains objects with intrinsic behavior which interact with each other and with us humans when we manipulate them. Most importantly, it contains other humans, with whom we interact in a social context. All of these can be simulated in a virtual world. Behavior algorithms, based on physical laws or other rules, can be added to objects. Remote users, represented in some virtual form (sometimes known as an "avatar") can be present in the virtual space, allowing multiple users to interact with each other in a form of graphical "multi-user-dungeon" (MUD). Virtual reality has the potential to change the World-Wide-Web, currently a static and solitary universe, into a dynamic and social one.

All of these ideas were very much in the minds of the participants at the first birds-of-a-feather meeting at the 1994 Geneva World-Wide-Web conference. It was this meeting which launched the project to create a Virtual Reality Markup Language (VRML, which later became "Virtual Reality Modeling Language") that was intended to become a standard data format for writing networked virtual environment simulations on the Web. During the intense e-mail discussions in the following months, it became obvious that a truly general-purpose language for representing arbitrary environments was not a practical short-term goal. In fact no such language truly existed, even in an experimental form. Since the ultimate goal of VR is to create a virtual world with the same richness and complexity as the real world, such a general-purpose language must attempt to model many of the complexities of real-world objects, including not only their form and visual properties, but also their intrinsic behavior and dynamic relationships to other objects. Most importantly, such a language must be able to encode the manner in which both remote and local users interact with the virtual objects, and provide a protocol by which multiple distributed users may synchronize their local environments so as to give the impression of coexisting in the same virtual environment.

1.2 VRML-1.0

Given the extremely ambitious nature of the task, and keeping with the technical spirit of the World-Wide-Web, it therefore seemed more practical to take an incremental approach to creating a VRML standard. While no general-purpose virtual environment data format existed, 3D graphical file formats such as the GKS-3D and PHIGS metafile formats did exist and were well-understood. A first version of a VRML standard file format was therefore proposed which can be thought of as something analogous to an image file format. Rather than 2D pixel information, however, it instead contains the actual 3D geometric data of the objects comprising the virtual world. In addition, surface material information specifies the colors and other reflective properties of the various objects, lighting information describes the position and orientation of lights in the scene, and viewing information determines from what vantage point and angle it is initially viewed by the user.

Selecting a hyper-link to such a VRML file loads the 3D data into the local machine and allows it to be viewed in three dimensions using whatever 3D viewing techniques the local browser software and workstation hardware allow. Like an HTML document, the VRML document can be structured so that portions of it may be associated with hyper-links to other Web URLs, while other portions may be "inline" references to other VRML files. VRML is also hierarchical, so that complex 3D scenes can be built up from combinations of object components and assemblies. This allows a single virtual space to contain a combination of objects located at distributed sites, as well as "hot" objects which are analogous to text or multi-media hyper-links in an HTTP document. Interactively selecting a hot object hyper-linked to another VRML file will cause the user to be "teleported" to the associated 3D space. Selecting a hot link to a more traditional HTML document would invoke a standard document browser to view it.

Most importantly, rather than the ultimate design goal of a true "networked interactive simulation" standard, the first version of the proposed VRML standard describes an essentially static world. There are no behaviors, no interaction techniques and there is no support for multiple users. Given the fast-paced development of the Web, it seemed that the highest priority should be to make a minimal level of VR functionality available to everyone to allow a basic standard of interoperability. This would prevent a potentially debilitating plethora of competing standards and buy time for people to explore the less well-understood problem of adding behavior extensions to the initial standard.

The desire for quick results also dictated that the initially somewhat idealistic plan to build up a VRML standard from scratch be abandoned in favor of modifying an existing standard. While several standards were proposed, including ones based on OOGL [13], Labyrinth [15], CDF [21], and MSDL [16], a consensus rapidly developed around using a simplified version of the OpenInventor file format. OpenInventor [19], a general-purpose object-oriented 3D graphics toolkit developed by Silicon Graphics, is well-poised to become the industry standard 3D graphics toolkit, and SGI proposed a VRML-1.0 standard based on a simplified version of its ASCII file format.

In fact, this VRML-1.0 standard, which was announced at the 1995 WWW conference in Darmstadt, added only two new constructs to the OpenInventor standard: a WWWAnchor statement, which allows a piece of geometry to be associated with a URL as a hyper-link, and the WWWInline statement, which indicates the URL of a VRML document to be loaded into the currently-viewed scene. This relationship between VRML-1.0 and the commercial OpenInventor product has made it very easy to develop VRML-based software by using the OpenInventor toolkit and has helped SGI to position itself as a leading supplier of VRML-based products. It has also raised some important questions about who controls the future course of the VRML standardization effort, and whether or not the original dream of a true virtual reality modeling language has resulted in little more than just another industry-standard 3D graphics file format.

1.3 VRML-1.0 Applications

Even if it is currently far from truly being a “virtual-reality modeling language”, VRML-1.0 is enjoying a big success. The possibility of making available annotated 3D environments on the World-wide Web opens the road to the creation of a new class of World-Wide Web applications. The relatively low cost of modern 3D graphics platforms makes it possible for a large community of users to benefit from this evolution. Although the static nature of VRML-1.0 currently limits user’s interaction with 3D environments to real-time navigation and hyper-link selections, these operations are still sufficient for a large number of applications. In scientific visualization, CAD, and virtual environment applications, virtual camera motion specification is often the most important form of three-dimensional interaction[23, 6, 9]. The visual cues provided by interactive 3D viewing with continuous viewpoint control can offer invaluable help in understanding the represented data. If images are rendered smoothly and quickly enough, an illusion of real-time exploration of a virtual environment can be achieved as the simulated observer moves through the model[6]. For example, in architectural CAD applications, this gives both the architect and the client the ability to naturally walk through virtual buildings and inspect them from any angle, allowing the architect to explore complex spaces and the client to provide feedback early in the design process[6, 1]; for scientific visualization, large multi-dimensional data sets can be inspected and better understood by walking through their 3D projections[22].

The effectiveness of interactive 3D viewers for communicating information about 3D environments can be dramatically enhanced by attaching multi-media annotations to the environment's models. This is one of the major advantage that VRML browsers have over simple 3D viewers. By allowing users to interactively recall and view the attached information by selecting objects of interest during navigation, a VRML browser becomes a natural front-end for querying information about 3D models. Since the latest graphics workstations have become true digital media platforms that combine sound generation, interactive 3D graphics and movie playback capabilities, annotations can refer to text, still images, animations or even other 3D models. In the architectural CAD example, the virtual building representation could be augmented in this way by linking to its various components the original drawings showing engineering details of the structure, photographs of the real site, and so on. The interactive 3D model can therefore be used for data management purposes during the design phase, and information about the building can be presented to the client with maximum efficiency.

2 VRML-1.0 Browsers

A number of VRML 1.0 browsers that offer interactive viewing and hyper-link selection capabilities have been developed so far. These include Webspace [25], WebOOGL [26], and WebView [27].

Most of these systems are limited to mouse-based interaction and are not able to ensure constant high frame rates when dealing with large datasets, thus limiting their appropriateness for large scale projects. To provide an overview of current VRML browsers' advanced capabilities, we describe in the following sections a Web browser developed by one of the authors called i3D. i3D is [3], the only VRML browser to date that incorporates the 3D input and high-performance rendering capabilities of high end VR system with the data retrieval abilities of standard network browsers. It is implemented on top of X11/OpenGL and runs on Silicon Graphics and DEC Alpha workstations. Using a 3D device, the user can explore its three-dimensional data and request access to other documents. When retrieving and displaying media documents, i3D handles the three-dimensional data directly and collaborates with NCSA Mosaic or Netscape for other types of media. Stereo-glasses are used to provide binocular perception of the 3D world.

Executable versions of i3d for supported platforms are made publicly available at the address "http://www-venus.cern.ch/i3d/".

2.1 Application Overview

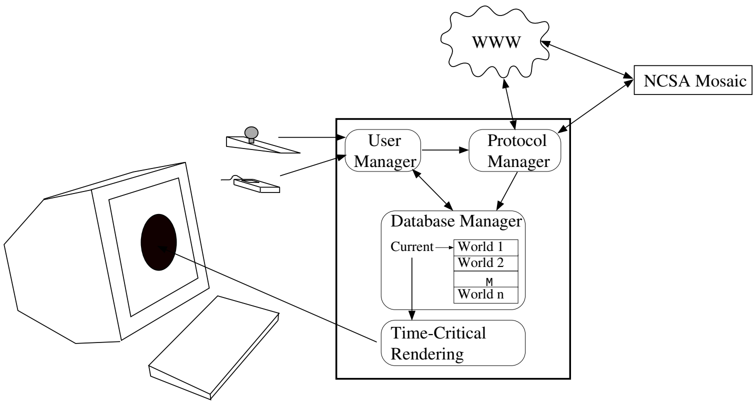

Fig. 1. Application overview

As shown in figure 1, I3D is composed of the following units: the user manager, the protocol manager, the database manager, and the rendering manager. The user manager is responsible for sensing and analyzing the user's movements and actions in order to recompute the new viewpoint position and orientation, to trigger retrieval of media documents by the protocol manager, and to navigate among the stack of worlds that is maintained by the database manager. The protocol manager is responsible for the retrieval of media documents from the World Wide Web. Three-dimensional scenes are loaded locally and transmitted to the database manager, while requests for other types of media documents are delegated to a WWW browser (NCSA Mosaic or Netscape) for retrieval and display by the most adequate viewer application. The database manager maintains the state of the 3D scenes in order to provide the necessary geometrical information and visual attributes for the user and rendering managers to perform their tasks. It also maintains a stack of all the scenes that have been visited to reach the current world and provides fast switching between worlds upon the user manager's request. The rendering manager is responsible for the generation of the visual representation of the current scene at a high and constant frame rate (e.g. 10 frames per second).

2.2 User Interaction

I3D's device configuration uses a Spaceball and a mouse as input devices. The Spaceball is used for the continuous specification of the camera's position and orientation using an eye-in-hand metaphor[24], while the mouse is used to select objects and access multi-media documents through hyper-links. Both user's hands can therefore be employed simultaneously to input information. Keyboard commands are used to control various visibility flags and rendering modes. The ability to continuously specify complex camera motions in an intuitive way together with high visual feedback rates provides an accurate simulation of motion parallax, one of the most important depth cues when dealing with large environments[1].

Fig. 2. I3D's device configuration.

While navigating inside a three-dimensional scene, the user can request additional information by accessing Web documents associated with the geometrical data. Since these hyper-linked geometrical figures are drawn with a blue silhouette, they can be visually identified easily. Selecting an annotated geometry by clicking on its visual representation with the mouse triggers the associated Web document's retrieval and display. Object selection is implemented using ray-tracing. For three-dimensional scenes, i3D maintains a stack of active worlds. Using keyboard commands, the previous or next world in the stack can be made current, thus providing a means to quickly navigate among active worlds.

2.3 World Model

I3D's database manager stores the representation of three-dimensional scenes as a collection of 3D objects, including light sources and cameras. This collection of objects is obtained by flattening the hierarchical input VRML file. From the resulting set of objects representing the virtual world, the database manager then builds an octree spatial subdivision to enable rapid spatial queries. For example to select objects using the mouse, a ray is traced from the viewpoint through the current mouse position until it intersects an object in the scene.

2.4 Rendering

The task of the I3D's rendering manager is to display a visual representation of the current world at high and constant frame rates. During navigation, the rendering manager is activated at regular intervals by the main i3D event loop and is requested to refresh the screen while adhering to the user-specified timing constraint. At each activation, the rendering manager renders a single frame by executing a sequence of operations. These operations are described in the following paragraphs:

Visibility determination First, the database is traversed and the objects visible from the observer's viewpoint are identified. This task is accelerated by first hierarchically determining the visibility of portions of the scene through a traversal of the spatial subdivision maintained by the database manager;

Display list construction Each of the objects identified in the previous step is then compiled into a graphical description by stripping off its appearance attributes and compiling them into a device-dependent sequence of commands (in the current version of i3D, OpenGL display lists are used for storing the compiled versions of graphical objects). During this conversion, geometries are optimized to reduce their rendering time (in particular, structured triangular meshes are generated from the triangle lists stored in the database). To avoid recreating compiled versions at each frame, as it is done in systems like Performer[17], i3D caches the graphical descriptions generated for each database object and reuses them until they become invalid. The validity of a description is checked by using a time-stamping scheme: the database manager maintains for each of the attributes of the model a time-stamp that is updated every time the attribute is modified, and graphical descriptions are time-stamped by the rendering manager as the most recently modified attribute they requested.

Level of detail selection To reduce the number of polygons rendered in each frame, so as to be able to meet the constant-time rendering requirements, the rendering manager traverses the generated display list and selects the level of detail at which each of object will be represented. Level of detail selection is based on the importance of each object for the current frame (which is determined by computing an approximation of its size projected on the screen) and on feedback regarding the time required to render previous frames. The feedback algorithm is similar to the one presented in[17];

Display list optimization Once the levels of details are selected, the system has all the information required to render the frame. To exploit coherence, the display list is sorted to optimize the rendering speed. In particular, objects sharing the same texture and/or material are grouped together. In other visual simulation systems, this task is left to the scene designer, who must encode this information together with the scene description[17, 7, 19];

Display list rendering The sorted display list is finally traversed and rendered by executing each of the compiled command sequences. Rendering statistics for the current frame are updated and stored so as to be used when determining the level-of-detail selection for the next frame.

3 VRML-1.0 Application Examples

During the last year, a number of projects that exploit VRML-1.0 capabilities have started to be developed. CERN VENUS and Virtual Sardinia are two of the most large scale projects. These two projects, described in the following sections, use i3D as a VRML browser because of its suitability for dealing with complex scenes.

3.1 The CERN VENUS Project

The European Laboratory for Particle Physics (CERN) is currently involved in designing its next generation particle accelerator, namely the Large Hadron Collider (LHC). In any project of this scale, the design phase is probably the most delicate one, as this is when some critical choices are to be taken which might dramatically affect the final results, timing and costs. The ability to visualize the model in depth is essential to a good understanding of the interrelationships between the parts. An iterative design optimization process can remarkably improve solutions to space management and ergonomic issues. However, with the visual capability of the present CAD tools, it takes a fair amount of time and imagination to isolate eventual design faults. A pilot project was started at CERN in January '94, named VENUS (Virtual Environment Navigation in the Underground Sites). Its mandate is to produce a detailed virtual prototype of the LHC premises and allow navigation and access to engineering data in the form of flythroughs by natural interaction. The i3D system is being actively used for the exploration of the virtual prototype. Figure 3 shows a snapshot of a typical i3D session.

The VENUS virtual prototype is extracted entirely from the original EUCLID CAD database. As soon as engineers add new drawings, these are extracted and converted to an internal i3D format in two steps: first, EUCLID data is converted to Wavefront OBJ format; then, the resulting Wavefront objects are converted to the i3D format and assembled into a scene. Some minor manual treatment is necessary at this point, in order to compensate for the lack of some features (e.g. color and textures) in the EUCLID -to-Wavefront converter utility supplied by Matra-Datavision. Hypertext annotations that refer to various sources of information are then added by associating URLs with relevant 3D objects. A further step is necessary to optimize the geometry for interactive navigation. The entire process is fairly automatic and does not require a major effort, since the data conversion and optimization are handled by software utilities.

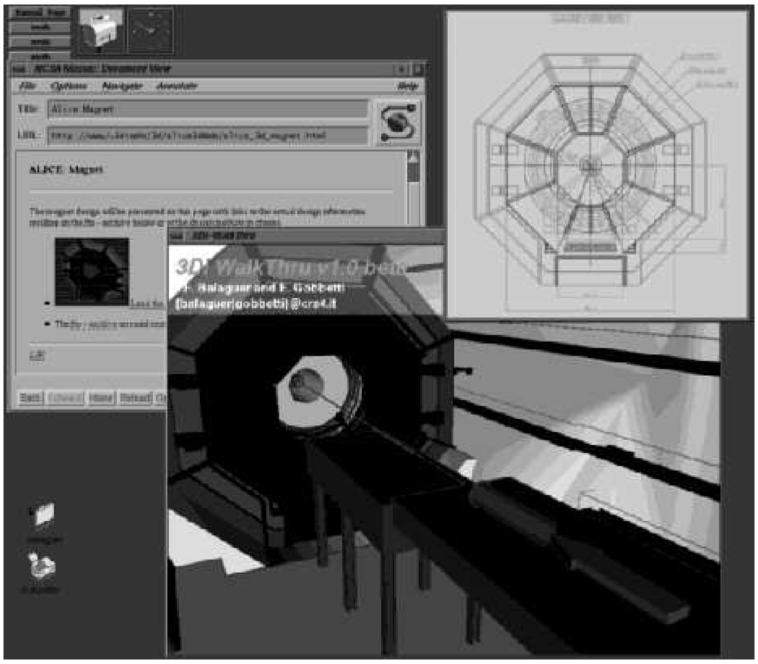

Fig. 3. Exploring CERN Alice.

When dealing with geometries, the way objects are constructed or exported from the modeling tool or CAD system may influence the visual representation of the objects or the rendering performance. A typical problem is the direction of surface normals not being specified or defined only per face (rather than per vertex), resulting in a flat shading rendering of the object. Another typical problem is that often multiple topologically separated models are merged inside a single geometry file, hence preventing efficient culling of non-visible objects when rendering. Filtering tools are provided to help world designers to work around these problems by smoothing normal vectors and splitting geometries. Additionally, another filter can be used to define levels of detail by creating oriented bounding-box geometries around small objects.

The entire conversion process should be completely automated in the near future, and triggered from the EUCLID side upon any significant changes to the geometry. In this way the virtual prototype will always reflect the latest state of the design. I3D is made available to all CERN users, allowing any CERN user with a Silicon Graphics workstation to connect to the CERN server, fly through the latest models of the detectors, and inspect all hyper-linked annotation information attached to the three-dimensional model. A subset of all the information available to CERN designers is made publicly available on the internet at the address "http://www-venus.cern.ch/".

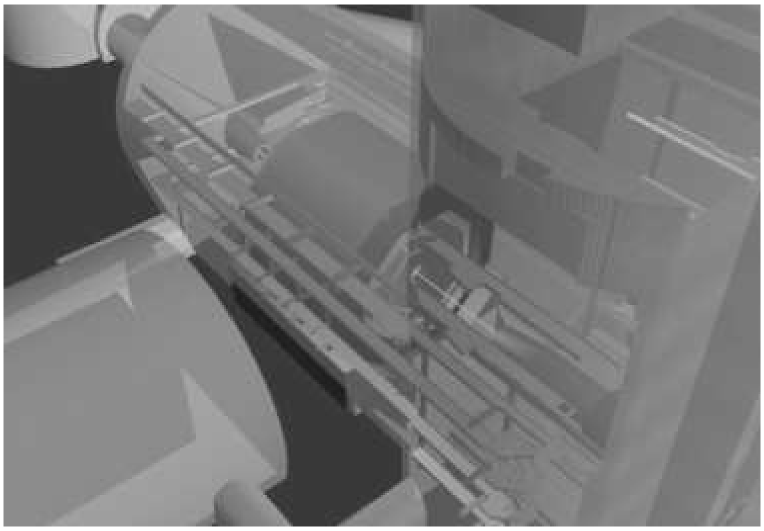

Fig. 4. CERN Alice pit viewed using i3d

3.2 The Virtual Sardinia Project

The Virtual Sardinia project [8] under development at the Center for Advanced Studies, Research, and Development in Sardinia (CRS4) aims at collecting a large amount of heterogeneous data concerning the island of Sardinia and representing it in such a way that a casual user can easily navigate through it in a virtual trip. All these data are interconnected in a Web-browsable hypermedia style, both in 2D and 3D, ranging from geographic and archaeological data to historical and tourist information.

Using i3D, users can explore a 3D model of the island, built from digital terrain model data textured with satellite images. 3D markers representing hyperlinks are positioned on the surface of the terrain to indicate sites of interest. Using the mouse to interactively select of one of these markers during navigation triggers the loading of a descriptive Web document, while the selection of any other location on the terrain triggers the request to view a high-resolution version of the area surrounding the selected point.

The possibilities of i3D are exploited to allow the exploration of detailed terrain models on a range of machines. To produce a VRML description of the terrain, the original terrain data (a regular grid) is subdivided into subregions that can be drawn and culled independently, and each subregion is described at various levels of detail by transforming the original regular grid into simpler irregular triangular meshes through a decimation process that iteratively groups nearly coplanar polygons and simplifies them. Different tolerances for planarity checks are used to produce the various levels of detail. To avoid cracking problems, small tolerances are used at the borders of the subregions. In addition to the optimization of the geometric model, images that are to be used as textures

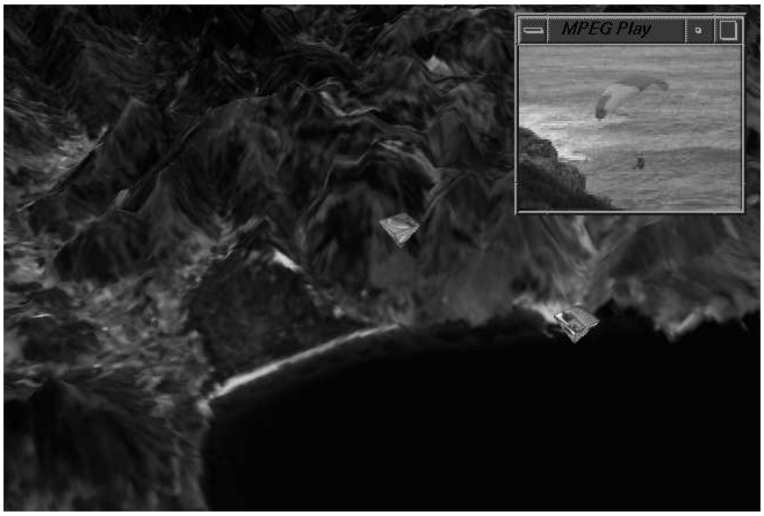

Fig. 5. Sight-seeing in Virtual Sardinia

are clipped and rescaled so as to have them fit into texture memory. All these optimizations are done automatically by a tool that takes as input the original digital terrain model, the satellite images and a list of descriptions of geographical location that have to be marked and associated to an hyper-link. Thanks to hierarchical culling and on-the-fly level of detail selection, the resulting model can be explored at interactive speed (more than 10 frames per second) on a Silicon Graphics Onyx RE2. The Virtual Sardinia project shows that, with appropriate preprocessing tools, it is possible to use VRML even for applications that usually require real-time visualizations of detailed terrain representations.

World-wide-web users may access the Virtual Sardinia project at the address "http://www.crs4.it/PRJ/VIRTSARD/".

4 Conclusions

The marriage of virtual reality and the Internet can be viewed from both an evolutionary and a revolutionary perspective. As a revolution, it presents us with no less than the possibility of giving cyberspace a tangible 3D form. The Internet could becomes a distributed virtual space in which we can move around, explore information, manipulate dynamic objects and tools and interact with other users in a social context. As an evolution, it is little more than a 3D data format standard for the Web. Just as a hyper-link may represent a piece of text or a 2D image, it may also point to a piece of 3D geometric information. To understand the vision behind extending World-Wide-Web browsers to support virtual reality, and its current state of development, it is important to examine the Internet/virtual reality union from both these perspectives.

Fig. 6. Collaboration between i3d and Netscape during a Virtual Sardinia Session

While virtual reality promises us limitless possibilities for the man-machine interface, the Internet is delivering on the promise of limitless access to information. Most VR applications are information poor, requiring tremendous amounts of data to flesh out their virtual environments with enough detail to satisfy the user's desire for visual complexity. The Internet, on the contrary, is information rich, usually presenting the user with too much information in too disorganized a way to be useful. While textual (or verbal) communication is certainly the most general-purpose means of inputting data to the human brain, visual communication has the highest-bandwidth. Adding three-dimensional information, with highly interactive multi-modal user interfaces that make use of sound, tactile and haptic display, has the potential to make the greatest use of the human brain's sensory and motor processing capabilities to create the highest possible bandwidth of man-machine interface.

Despite its name, VRML-1.0 is not a truly "Virtual Reality Modeling Language", at least no more than GKS-3D and PHIGS are. It is currently limited to providing a standard way for describing static 3D scenes composed of objects that can be annotated with hyper-links. The challenge of virtual reality, creating synthetic worlds that "look real, act real, sound real, feel real" [20] is only partially addressed. While VRML offers some support for describing worlds that "look real", these worlds have little possibility to "act real", as all VRML-1.0 objects have only one possible behavior: asking the browser to fetch associated documents when selected.

Recent VRML extensions proposals [18] [10] have started to focus on this problem, providing ways to specify animated and interactive behaviors in VRML worlds. Describing animated and interactive behaviors is, however, a long term research topic, which people in computer graphics, computer simulation, knowledge engineering and other disciplines have been studying for years, and it is very unlikely that a standard for appropriately describe them could emerge anytime soon.

Despite all its limitations from the "revolutionary" perspective, VRML-1.0 is very successful in allowing the use of 3D graphics as an effective medium to share information on the Web. Much as the availability of graphics library standards such as PHIGS, GKS-3D and OpenGL allow the creation of portable 3D graphics applications, the availability of a standard file format for describing annotated 3D scenes allow a larger distribution of 3D graphics documents. An entire new class of hypermedia applications has been enabled by the standardization of this technology,

5 Acknowledgements

i3D has been designed and developed by Enrico Gobbetti at the Center for Advanced Studies, Research, and Development in Sardinia (CRS4), Cagliari, Italy, with help from Jean-Francis Balaguer. The work is now continued by Jean-Francis Balaguer at CERN, Geneva, Switzerland. The Virtual Sardinia processing tools have been developed by Enrico Gobbetti and Andrea Leone at CRS4. Research and development for i3D and Virtual Sardinia have been carried out at CRS4 with the financial contribution of Sardinia Regional Authorities.

References

1. F. P. Brooks Jr and P. Frederick: Walkthrough - A Dynamic Graphics System for Simulating Virtual Buildings, Proc. ACM SIGGRAPH Workshop on Interactive 3D Graphics (1986): 9–22.
2. J.-F. Balaguer and E. Gobbetti: A New Dimension to HyperMedia. URL: http://www.crs4.it/ 3diadm/ (1995).
3. J.-F. Balaguer and E. Gobbetti: i3D: A High-Speed 3D Web Browser. Proc. ACM Symposium on VRML (1995).
4. T. J. Berners-Lee, R. Cailliau, J.-F. Groff, and B. Pollermann: World-Wide Web: The Information Universe. Electronic Networking: Research, Applications, and Policy **2(1) (1992): 52–58.**
5. **S. Bryson, R. Pausch, W. Robinett W, and A. van Dam: Implementing Virtual Reality. ACM SIGGRAPH Course Notes 43 (1993).**
6. **T. A. Funkhouser and C. H. Séquin, Adaptive Display Algorithms for Interactive Frame Rates During Visualization of Complex Virtual Environments. Proc. ACM SIGGRAPH (1994): 247–254.**
7. **S. Ghee and J. Naughton-Green, Programming Virtual Worlds. ACM SIGGRAPH Tutorial Notes on Programming Virtual Worlds (1994): 6.1–6.58.**

8. E. Gobbetti, A. Leone, A. Marini: Virtual Sardinia: a hypermedia fly-through with real data. Proc. International Workshop on Soft Computing in Remote Sensing Data Analysis, Milan (1995).
9. J. D. Mackinlay, S. Card, and G. Robertson: Rapid Controlled Movement Through a Virtual 3D Workspace. Proc. ACM SIGGRAPH (1990): 171-176.
10. Microsoft Corporation: ActiveVRML White Paper. http://www.microsoft.com/intdev/avr/avwhite.htm (1995).
11. T. Munzner: An Experiment in Three-Dimensional Distributed Hypermedia. The Geometry Center, University of Minnesota (1995).
12. M. Pesce, P. Kennard, A. Parisi: Cyberspace. Proc. WWW1 (1994).
13. M. Phillips, S. Levy, and T. Munzner: Geomview User Manual. The Geometry Center, University of Minnesota (1994).
14. A. Parisi, M. Pesce: Virtual Reality Markup Language (VRML), URL: http://www.wired.com/vrml/ (1994).
15. A. Parisi, M. Pesce: Virtual Reality Markup Language (VRML) Working Specification, URL: http://vrml.wired.com/vrml/ (1994).
16. M. Preston, N. Gatenby, W. T. Hewitt: The Manchester Scene Description Language (MSDL) Version 1.01. http://info.mcc.ac.uk/CGU/MSDL/MSDL-intro.html (1994).
17. J. Rohlf and J. Helman: Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics. Proc. SIGGRAPH (1994): 381–395.
18. Silicon Graphics Inc.: The Moving Worlds Proposal for VRML 2.0. http://webspace.sgi.com/moving-worlds/ (1996).
19. P. S. Strauss and R. Carey: An Object-Oriented 3D Graphics Toolkit. Proc. SIGGRAPH (1992): 341–347.
20. I. Sutherland: The Ultimate Display. Proc. IFIP 2 (1965): 506-508..
21. C. Tollander: Cyberspace Description Format (CDF). http://vrml.wired.com/proposals/cdf/cdf.html (1994).
22. C. Upson, T. Fulhauber, D. Kamins, D. Laidlaw, D. Schlegel, J. Vroom, R. Gurwitz, and A. van Dam A: The Application Visualization System: A Computational Environment for Scientific Visualization, IEEE Computer Graphics and Applications 9(4) (1989): 30–42.
23. V. Watson: A Breakthrough for Experiencing and Understanding Simulated Physics, ACM SIGGRAPH Course Notes on State of the Art in Data Visualization (1989):. IV.26–IV.32.
24. C. Ware, and B. Osborne, Exploration and Virtual Camera Control in Virtual Three Dimensional Environments, Proc. ACM SIGGRAPH Workshop on Interactive 3D Graphics (1990): 175-183.
25. Webspace... because the World is not Flat! http://www.webspace.com/
26. WebOOGL: Integrating 3D graphics and the Web. http://www.geom.umn.edu/software/weboogl/
27. SDSC WebView - A VRML Internet Browser. http://www.sdsc.edu/EnablingTech/Visualization/vrml/webview.html

This article was processed using the LaTeX macro package with LLNCS style