

# **A Case Study in the Use of VRML2.0 for Marketing a Product**

Nick D. Burton, Alistair C. Kilgour, Hamish Taylor  
Department of Computing & Electrical Engineering  
Heriot-Watt University, Edinburgh

## **Abstract**

The addition of behaviours, animation and sound to Virtual Reality Modelling Language has created interesting new opportunities to market products on the Internet. This paper investigates these opportunities and illustrates their potential through a case study in promoting electronic bagpipes.<sup>1</sup> It also reports on an evaluation into 3D navigation in VRML2.0 that exposes problems users have with current navigational interfaces in VRML2.0 browsers and suggests a way in which such problems may be overcome.

---

<sup>1</sup> <http://calligraphix.co.uk/bagpipes/index.html>  
nickb@cee.hw.ac.uk, ackb@cee.hw.ac.uk, hamish@cee.hw.ac.uk

## 1. Introduction

Recently there has been a marked increase in the use of the Internet for many business activities. One of the main reasons for this is the ease with which multi-media documents can be constructed for the World Wide Web (Web or WWW). It is now possible to buy anything over the Web from military equipment to Italian sausages. Estate agents have access to a world-wide customer base, trade associations are using the Internet to keep their members informed with up to date information and so on. The richness of Web page content has improved dramatically. It was not long ago that it was only possible to display text and monochrome pictures. Now that it is possible to include colour pictures, sound, movies, 3D animation and interactive programs, there is much greater potential for marketing products. One of the more interesting recent developments in Web technology has been version 2.0 of the Virtual Reality Modelling Language (VRML2.0) [1]. VRML2.0 is a scene description language that enables the construction of interactive, animated 3D objects and environments on a Web page.

This paper investigates the possible use of VRML2.0 to enhance the Web-based marketing of manufactured objects using interactive and animated solid models [2]. The work was carried out in conjunction with Calligraphix<sup>2</sup>, a multi-media and Internet service provider in the Scottish Borders. The intention was to develop a Web-based virtual shopping mall within which products could be displayed for purchase over the Internet. It soon became apparent that the limitations of VRML2.0 would restrict the attractiveness of such a virtual shopping mall and it was decided to investigate the marketing potential of VRML2.0 for producing Web-based 3D models of a product instead.

Rather than choosing a well known product it was decided to focus on something novel that would be amenable to 3D modelling. The promotion of an electronic bagpipe seemed suitable as a case study since its shape was regular and the requirement for sound added extra interest.

Any technology for marketing a product on the Web, must:

- attract and maintain a potential customer's attention
- be easy to use by inexperienced computer users

---

<sup>2</sup> <http://calligraphix.co.uk>

- be capable of providing useful information about the product
- be cost-effective to employ

Hypertext mark-up language (html) is the technology most commonly used to construct Web pages. Text and 2D pictures can convey a great deal of information about a product, but both are static and are limited to what the Web page author chooses to present. It would be rather more useful, when marketing a manufactured product, if a potential buyer were able to interact with an animated 3D model of the product. This would engage the buyer's attention better, and would convey information about the product that could otherwise only be gained from real experience of it. It is just such an enhancement that is expected from the use of VRML2.0.

There are a number of Web technologies that offer 3D and interaction and these are discussed briefly in section 2. Section 3 gives an overview of the aspects of VRML2.0 that were pertinent to this case study, and section 4 discusses the construction of the 3D models used. The strengths and weaknesses of VRML2.0 for marketing a product on the Web are discussed in section 5. Section 6 describes an experiment that was conducted in order to gauge users' reaction to the problems of 3D navigation. Section 7 discusses problems with the user interface controls on some of the VRML2.0 browsers and describes a prototype custom interface for object manipulation in VRML2.0 that attempts to rectify some of these problems.

## **2. Existing Technologies**

At present there are a number of technologies that promise 3D and animation on the Web. Of these the most interesting are;

- Java
- Macromedia Shockwave
- QuickTime VR
- Virtual Reality Modelling Language

### **2.1 Java**

Java is a general purpose object-oriented programming language developed by James Gosling of Sun Microsystems back in 1990. It is (almost) platform

independent due to the fact that the source code is compiled into host computer. Java can be used to produce stand-alone applications. It can also be embedded in Web pages, where it is referred to as an 'applet'. Applets provide executable content for Web pages. Since May 1995, when Java was first 'loosed on the Internet' [3], numerous 'applets' have appeared which have added animation, both 2D and 3D, to Web pages. Some of these, such as the 'ticker tape' applet, quickly become irritating. A more interesting example of 3D Java animation is the 'Virtual Rubik's Cube' [4] which can be found at the 'Applet Arcade' [5]. An example of an interactive 3D environment written in Java is the game 'Dungeon Disaster' [6] which has some similarities to the well known computer game 'Doom'. The two previous examples demonstrate that Java is quite capable of providing both 3D and interaction in a Web environment, but the amount of development required to produce such applets (as evidenced by the available source code) would not make it cost effective for marketing products. Another problem is that applets which are more than just trivial eye-catchers can take a long time to load into a Web browser, and this is not attractive for business use.

## **2.2 Shockwave**

Macromedia Shockwave is a 'plug-in' for the Netscape Web browser. A plug-in is a piece of third party software that is used in conjunction with the Netscape Web browser to add functionality. The Shockwave plug-in allows Netscape to run Macromedia Director multi-media presentations. These are interactive presentations with clickable buttons, video and sound. A disadvantage with Shockwave is that it doesn't allow the user to explore a 3D environment by navigating in and around it.

## **2.3 Quicktime VR**

Quicktime VR is software that displays an interactive panorama on the Web. The panorama is derived from photographs of real scenes and users can navigate around the panorama by clicking on the particular point that they wish to visit. Quicktime VR is good for architectural applications such as displaying the interior of buildings, and a nice example of this can be seen at the Calligrafix Web site [7]. Navigation in Quicktime VR is restricted to jumping between pre-arranged viewpoints. It does not have the flexibility required to allow potential customers to investigate a product as they choose.

## **2.4 VRML2.0**

VRML, as its name suggests, is a modelling language that allows unrestricted virtual environments to be built. In other words the geometry of a virtual world in VRML is not constrained to be orthogonal (as it is for example in the popular game 'Doom'). The viewer of a VRML virtual environment, usually described as an 'avatar', is similarly unrestricted in the direction of travel through the environment. This makes VRML rather flexible. VRML(2.0) added interaction, animation and sound to the solid modelling and navigation capabilities of the original VRML, making it more promising for marketing solid manufactured objects on the Web. The case study suggests that although VRML2.0 has considerable potential for marketing, there are several problems with its use at the moment preventing it from being acceptable in a real business environment.

## **3. VRML2.0 Concepts**

It is important to understand key concepts of VRML2.0 to appreciate how it can be used as a marketing aid. This section explains relevant VRML2.0 concepts.

### **3.1 Overview**

VRML is a 3D scene description language in ASCII file format. A Web browser requires a VRML2.0 plugin or helper application to render the 3D scene described by a VRML2.0 file. A user may navigate in and around a VRML2.0 scene and may interact with it, causing it to display certain animated behaviours. A VRML2.0 scene can contain links to other VRML2.0 scenes and to html-based Web pages.

### **3.2 Basics**

Conceptually a VRML world consists of a scene graph in which each entity is a node. Each node has a type name, fields and in some cases, events that it can receive or send. The events carry information among nodes to allow animation and interactivity within a VRML2.0 world. A scene graph consists of Group nodes and Leaf nodes in a hierarchical structure. Group nodes are containers for child nodes, which can be group or leaf nodes. Transform nodes are the most useful group nodes. They allow their children to be transformed or rotated in the co-ordinate system of their parent. If a transform node is at the top of the scene graph (i.e. it has no parent) then its children will be transformed or

rotated with respect to the world's co-ordinate system. Leaf nodes describe the components that make up a 3D world, for example: shape, lighting, colour, sound and 2D text. VRML2.0 has the cube, cone, cylinder and sphere as its elementary shapes. Complex shapes can be built up from a combination of elementary shapes, or they can be described by 'indexed face sets'. An indexed face set is defined by a number of points in 3D space. These points are joined up to form polygons (usually triangles) which combine to make up the complex surface or shape. Figure (1) shows a simple example of an indexed face set.

VRML2.0 allows authors to define 'viewpoints' or locations in 3D space from which a user may view the world in a pre-set direction. A number of viewpoints may be defined and then made available for a user to select using a browser interface control. Extensive use of viewpoints makes it easy for a user to investigate a 3D world without having to grapple with the browser's navigation controls.

### **3.3 Interaction**

VRML2.0 provides a number of different ways in which users can interact with a scene. A user can investigate a 3D world by moving among its viewpoints or by navigating into it using the VRML2.0 browser's interface controls. A user can also traverse hyperlinks that connect parts of the scene to other VRML2.0 scenes or to html documents. Finally a user can initiate behaviours or animation in a VRML2.0 scene using the mouse while the cursor is over a variety of 'sensors'. Sensors are not visible but are used to detect user input. Geometry sensors generate events based on user actions. The TouchSensor generates an event when the cursor is over an object and the user clicks the mouse button. Three of the geometry sensors; CylinderSensor, PlaneSensor and SphereSensor generate events if the cursor is clicked and dragged when over an object. The dragging movement is translated into a translation of the object (in the case of the PlaneSensor) or a rotation (in the case of the other two sensors). Time sensors generate events depending upon the relative values of their start time and stop time fields. Time sensors are used in conjunction with geometry sensors to sequence keyframe animation.

### **3.4 Animation**

VRML2.0 supports key-frame animation. When a user triggers a geometry sensor, it generates an event which causes a time sensor to start. The time sensor then generates a stream of (time) events at regular intervals, and these

are sent to an interpolator. The interpolator calculates data values for each moment in time and sends these data values to the node containing the object(s) to be animated. Depending on the type of data received, the objects will be translated, rotated, or change colour, size or shape. Animations can be programmed to run once only or to repeat themselves. Figure (2) illustrates the process.

Key-frame animation is suitable for simple linear animation. If more complex animation is required, for example opening a door if it is shut and closing it if open, then scripting is required. Scripting opens up VRML2.0 to the computational power of languages such as JavaScript and Java.

#### **4. Authoring the 3D Models**

At the time that the case study was conducted VRML2.0 authoring tools were not yet available. Models could be authored in VRML1.0 tools and then translated to VRML2.0, or could be authored from scratch by hand. The electronic bagpipe (or mini-pipe), is played by touching a number of finger pads that are set into a hollow tube known as a chanter. Authoring the model by the first approach required the use of sophisticated software such as 'Alias Animator' to craft the complex shape of the holes in the chanter, and this presumed familiarity with the use of such software. In addition the translation software was in beta release, the VRML mailing list reported that it suffered from bugs, produced huge files and still required hand crafting in order to implement VRML2.0 specific features such as animation. The second approach (authoring the models by hand) required the tedious calculation of many points in 3D space to define indexed face sets for shapes that could not be represented by VRML2.0 primitives. The second approach was judged to be the quickest and was adopted. Three objects were modelled; the mini-pipe itself, a PP3 battery and a set of earphones. Figures (3) to (8) show the 3D models.

##### **4.1 Complex Shapes**

More than a thousand points were used to model the chanter section of the mini-pipe. Figure (8) shows a close up of one of the finger pads set onto a hole in the chanter tube. These cut-out holes were the most difficult part of the mini-pipe to model. The difficulty was caused by having to represent the material thickness of the chanter tube. Solid models in computer graphics are represented by closed surfaces and are not actually 'solid'. This is true whether the objects are modelled using polygon mesh surfaces [8] as in VRML, or

constructive solid geometry (CSG) [9] (which is used in some ray tracing systems). It is possible to represent a hole in a non-black plane surface of an object, by positioning a black object on that surface. This technique was used to model the earphone socket hole and the hollow parts of the battery terminals, but it is not very realistic as the holes do not appear to have any depth. It was not possible to use this technique to model the finger-pad holes in the chanter as the finger pads had to be recessed below the outer surface of the chanter tube as can be seen in Figure (8).

## **4.2 Animation**

Two components of the mini-pipe were animated; the lid of the battery compartment and the volume control knob. The battery and earphones were also animated. In all cases it was necessary to control the animation using scripts. The battery compartment lid opens and closes and the battery will go into the battery compartment or come out of it only if the lid is open. The volume control knob turns up or down and the earphones plug themselves into the ear-phone socket. The models were developed for the Cosmo Player browser which meant that the scripting language was VRMLScript, (a sub-set of JavaScript). The biggest problem with the animation was working out 3D co-ordinates for objects that were simultaneously rotating and translating. The local co-ordinate system rotates with each incremental rotation of the object, which means that the direction of translation along a particular axis also changes each time. The problem is compounded if rotation (with respect to the world co-ordinates) is required about more than one axis. A compromise solution is to complete any rotation before starting a translation, and this was the approach used for the battery compartment lid and earphone animations. The battery animation used simultaneous rotation and translation and required a great deal of trial and error to determine the path of the battery. Hopefully VRML 2.0 authoring tools that simplify this process will become available soon. Animation is acknowledged to be one of the most difficult aspects of VR authoring.

## **5. Strengths and Weaknesses of VRML2.0 for Product Marketing**

### **5.1 Strengths**

### **5.1.1 Web-based**

Nowadays, Web browsers are available on virtually all platforms and are used by many people with a broad range of computer expertise. Obviously information about a product which can be displayed on Web browsers will reach a wide audience with the minimum of distribution effort. There is no real alternative to VRML on the Web for the widespread dissemination of interactive 3D information.

### **5.1.2 3D**

With so much information available on the Web, one of the main concerns of Web authors is to attract users to their page. People find 3D fascinating, and if there is sufficient user interaction built in to a 3D environment, this will not only attract interest but maintain it as well.

### **5.1.3 Relatively easy to create**

VRML2.0 is a scripting language. It is not necessary to know anything about computer graphics in order to produce interesting 3D scenes. This makes VRML2.0 attractive to content producers who will be coming more from an artistic than a computer science background.

### **5.1.4 Demonstration of product behaviour**

It is possible to demonstrate the functionality of a manufactured product using video embedded in a Web page. As far as the viewer is concerned this is a passive activity. Engaging a user's attention through interactive involvement with the 3D environment makes VRML2.0 more attractive. The user and not the computer is in control!

## **5.2 Weaknesses**

### **5.2.1 More than one browser required**

Additional software in the form of a plug-in or helper application is required in order to view VRML2.0 scenes. VRML2.0 was supposed to be a standard but unfortunately during the negotiation of the scripting part of the VRML2.0 specification there was a language war involving Javascript and Java. The result is that some browsers only support Javascript and other browsers only support Java. VRML2.0 content authors tend to develop for a particular browser and

their files will not work in all of the VRML2.0 browsers. It is to be expected that browsers of the future will support both Javascript and Java, but at the moment it is a real problem that most do not.

A Web browser will display a VRML2.0 scene in its own window. In order to allow VRML2.0 scenes to be embedded within, and controlled from an html file, the Java External Authoring Interface (EAI) [10], was developed. Unfortunately not all the browsers support the EAI, so it is quite common to download a VRML2.0 file and find that it doesn't work in a particular browser. The lack of portability across VRML2.0 browsers dramatically reduces the potential audience that authors can expect to reach and makes VRML2.0 much less attractive for the marketing of products.

### **5.2.2 Unacceptable file download time**

A number of factors contribute to the size of VRML2.0 files. First, it is an ASCII file format. Second, complicated shapes may have a high polygon count. Third, textures tend to require a large amount of specification. Fourth, sound files tend to be large. The result is that if you have packaged your complicated product in a virtual environment with realistic textures, enhanced with sound, the user may have to wait 20 minutes for the scene to load. If your VRML2.0 scene contains many links to Web pages or VRML2.0 scenes that also have to be downloaded, your potential customer is going to lose interest quickly.

### **5.2.3 Slow rendering of complex scenes**

VRML2.0 scenes are rendered in real time by the VRML plug-in or helper application. This can result in unsatisfactory animation due to the current limitations in the processing power of typical end-user computing vehicles. One of the worst effects is the loss of textures that can occur if a scene is changing rapidly. If a scene relies heavily on texture-mapping to convey 'photo-realism' it is unacceptable for these textures to be replaced by minimal wire-frame outlines whilst the scene is moving. It is hardly surprising that the rendering capability of VRML2.0 is limited when you consider that each frame in the computer-generated film 'Toy Story' took up to 15 hours to render. Nevertheless people's expectations are high, largely due to the excellent graphics quality in modern computer games. For users who do not have an appreciation of the technical difficulty of rendering 3D scenes in real time, the results seem to be crude and not 'state of the art'.

#### **5.2.4 Time to author complex shapes and behaviours**

Simple shapes that can be built up from the elemental shapes can be authored quickly and efficiently. More complex shapes, requiring indexed-face sets, are time consuming to author by hand. Authoring packages are available now that make the task of modelling complex shapes easier, but there are still areas where a certain amount of hand tweaking is required. One of the most difficult shapes to model is a hole punched through a solid object. Simple key-frame animation is likewise relatively straight forward to achieve and is available on at least one of the VRML2.0 authoring packages. Complex animation on the other hand, requires a lot of hand crafting. For inexpensive products the cost of authoring a marketing aid using VRML2.0 is going to be prohibitive for some time to come.

#### **5.2.5 Wide colour variation across platforms**

The rendering of colour in VRML2.0 scenes varies so much from platform to platform that a working group has been set up to try and address the problem. In some cases, for example 3D games, obtaining the correct absolute colours may not be critical, although the aesthetic appeal of a scene may be adversely affected. For marketing products such as pictures, textiles or clothing, it is essential that the colours created by the VRML2.0 author are reproduced faithfully in the browsers of potential customers.

#### **5.2.6 Awkward 3D navigation**

The main reason why VRML2.0 should be such an effective marketing aid is that it promises potential customers the ability to explore a virtual product. The interface controls on VRML2.0 browsers are based on the concept of a viewer, (or avatar), who navigates in and around a virtual environment, but the controls are so awkward to operate that navigating a VRML2.0 environment is frustrating. Not at all the situation that will encourage users to stay at a site. In addition there is no built-in interface that will allow users to manipulate individual objects within the 3D scene. This functionality has to be programmed into the VRML2.0 file. The next section reports on a study that was carried out to gauge users' reactions to VRML2.0 navigation [2].

### **6. 3D Navigation Experiment**

The purpose of this study was to ascertain the attitude of users towards 3D navigation in VRML2.0. An experiment was conducted in which subjects were

timed as they navigated a 3D road circuit, (Figure (9)). The main objective of the experiment was to ensure that each subject had a similar experience of 3D navigation in VRML2.0 prior to forming a judgement about its suitability for navigating virtual shopping malls. Also of interest was the degree of improvement (if any) between a subject's first and second lap of the course. Following the experiment the subjects were asked a few questions relating to the system they had just used. They were also asked to choose a preferred environment for navigating a virtual shopping mall, from: VRML2.0, a 2D map and a list of shops or products.

## **6.1 Subjects**

Ten volunteers took part in the study, seven male and three female, ranging in age from 15 years to 47 years old. Four of the subjects were of school age (15 to 17 years old), three were between 20 and 40 years old and three were over forty years old. Of the ten subjects, only two used a computer less than once a week, three used a computer on average once a week and five used a computer every day. Two of the subjects were not British but spoke fluent English.

## **6.2 The Experiment**

Every subject was given the same task. They were asked to navigate around a VRML2.0 road circuit and then after a pause of about one minute they were asked to navigate around it again. The time taken for each complete navigation of the road circuit was recorded. No subject had experience of the 3D course or VRML2.0 prior to the experiment.

## **6.3 Materials**

Elonex 486/66 PC with 16Mb RAM, running Windows 95

Cyber Passage V2 beta 2 (Sony's VRML2.0 Browser)

Netscape 2.02

VRML2.0 file 'drive/main.wrl' from Sony Pictureworks

## **6.4 Results**

Figures (10) and (11) summarise the preferred navigation method. Results for the navigation experiment are summarised in Figure (12).

## 6.5 Assessment of Results, 3D Navigation

Figure (10) shows that a majority of subjects rejected 3D navigation. This was not a statistically significant result as the Chi-square values in Figure (11) show. The observed Chi-square was 2.6 which is much less than the calculated value of 4.61 (for an alpha of 0.1). It is necessary to go to an alpha of 0.5 to get a calculated Chi-square less than the observed value. This can be explained by the small number of people in the sample, (for example if the ratio of preferences was kept at 1:4:5 but 100 people had been sampled instead of 10, the observed Chi-square would have been about 26.

The object of the study was to get a feel for people's attitudes rather than to prove a theory, so the statistical significance of the result was not a prime consideration. Initially two subjects chose 3D as their preferred navigation method, both were in the youngest age group and did so because they thought that it was fun. One changed after reflecting on the question (they hadn't initially related it to shopping). It is interesting that the only subject who stayed with 3D as a preferred navigation "because it was fun", declined the invitation to do a second lap. May be it wasn't that much fun!

The results for improvement of lap time from the first to the second lap were also not significant. Figure (12) shows that the measured 't' value was 1.71 compared to a calculated value of 1.75 (with an alpha of 0.1). A possible reason for this is that some subjects tried to use the 'in-screen' controls on the first lap and went wildly off the course, (one started going the wrong way round the track). These subjects reverted to the safer 'arrow' controls for the second lap. The standard deviation for the first lap was 1.36 which is high compared to a mean lap time of 3.98 minutes. The second lap had a much more acceptable standard deviation of 0.73 with a mean lap time of 3.1 minutes.

Conceptually the navigation controls were straightforward which is why Cyber Passage was used in preference to Cosmo Player. In practice the controls proved to be awkward and frustrating to use as is illustrated by the following selection of comments;

- Subject 2: "I certainly couldn't use the controls on the screen, I would get lost. I was glad there were two types of control. It's too slow for shopping malls"
- Subject 3: "This is difficult" (twice). "I wouldn't have the patience, I would find it very irritating".

Subject 5: "It's not doing what I wanted it to do, I don't know where I am, Oh here we are". (This subject went completely off the road a couple of times).

Subject 9: "Your concentration is on the navigation and not where you are".

The comments and results discussed above reinforce the belief that building a virtual shopping mall in VRML2.0 would not be beneficial, at least without a major improvement in the navigation interface(s) offered by VRML2.0 browsers. The situation will improve when the appropriate rendering software is implemented in hardware. A binary file format for VRML2.0 [11] is being developed, and this should improve download times. It is not just speed though which needs to be addressed, the basic usability of the controls has to be improved. For marketing products it is desirable that potential customers can manipulate objects directly rather than navigating around them. The next section describes a prototype interface for this purpose.

## **7. VRML2.0 Browser Interfaces**

There is a need for a VRML2.0 interface aimed at object manipulation rather than 3D navigation. A prototype object manipulation interface is described.

### **7.1 Design of Custom Interface**

The interface consists of four icons which represent four modes of object manipulation; translation in the plane of the computer screen, translation in a vertical plane going into the computer screen, rolling about an arbitrary point and spinning about a single axis. In addition to the iconic representation of each mode, a textual description appears when the cursor is passed over the icon and disappears when the cursor is moved off again. An icon is highlighted to show that it has been selected. Once an icon has been selected, objects may be manipulated by clicking the mouse when the cursor is over the object and then dragging the mouse. The dragging movement of the mouse is converted into movement of the object according to whichever mode is current. The standard browser interface is still available if required or it can be switched off. Figures (13) and (14) depict the custom interface within a sample VRML2.0 file.

Ideally the architecture of a VRML2.0 system should allow an interface implementation to be separated from the file that is being viewed. At the time of the project not enough information was available to allow a separate

interface to be developed. The interface was therefore programmed into a VRML2.0 world. It is possible to define an interface as a VRML2.0 Prototype that can be re-used and adapted. This would be an improvement over the present situation but it would still be an interface in a file.

The icons used in the interface are themselves VRML2.0 primitive objects and in order to get them to stay stationary with respect to the viewer it is necessary to position them within the front half of the volume occupied by the viewer's imaginary avatar, i.e. at the current viewpoint. The icons must be in the front half of the avatar to ensure that they are not clipped when the browser renders the scene. In order to keep the icons tracking the viewpoint (or avatar position) a proximity sensor must be attached to the icons. The proximity sensor should be large enough to cover the entire virtual world so that the viewpoint will always be inside it. As the viewpoint moves around inside the proximity sensor, the sensor sends position information to the icons' Transform node to enable it to move the icons in synchronisation with the viewpoint. Figure (15) illustrates this concept.

The icon design attempts to convey the type of movement associated with each mode. The translation icons depict the plane in which movement occurs and the rotation icons depict an object whose rotational behaviour is similar to the rotation of the objects, for example a ball and a cylinder.

A small red bar appears at the top of an icon to show that it has been selected. Selecting an icon is achieved by clicking on it with the mouse.

In order for objects to respond in the appropriate manner they must have a range of geometry sensors attached to them. The translation behaviours of the first two modes are achieved with a PlaneSensor. Sideways movement of the mouse is converted into sideways object movement for the first mode and object movement in or out of the screen for the second mode. A script node detects which mode is current and routes the translation information to the X or Z co-ordinate of the objects Transform node accordingly. Up or down movement of the mouse is converted into up or down movement of the object in both cases. For the rotational modes, movement of the mouse is converted into rotation of the object. In the case of the ROLL mode the rotation may be about an arbitrary axis. The axis of the SPIN mode is fixed.

A Switch node activates the appropriate icon and de-activates all others. A further switch node causes an appropriate text message to be displayed when the cursor is over a particular icon. When the cursor moves off the icon the text

message is removed.

## **8. Conclusions**

A case study has shown that VRML2.0 has the potential to enhance the marketing of products on the Web, however at the present time there are too many problems for its use to be of real benefit. The main advantage that VRML2.0 has over rival technologies is that it allows the exploration of a 3D environment with movement in all degrees of freedom, This gives a potential customer the sense of being in control. In addition the interaction and animation capabilities of VRML2.0 can convey much information about a product's functionality. What limits VRML2.0's usefulness as a marketing aid is that it is not cost-effective for inexpensive products, it is non-standard across VRML2.0 browsers and platforms, the rendering of colour varies unacceptably, it is slow and the browser interfaces are not user-friendly.

## References

- [1] *The VRML2.0 Specification*, <http://vag.vrml.org/VRML2.0/FINAL/>.
- [2] Burton N, *Interactive Solid Models for Enhancing Business on the Internet* Master's thesis, Heriot-Watt University, 1995-1996.
- [3] Semich W. and Fisco D, *Java: Internet Toy or Enterprise Tool* Datamation, March 1996.
- [4] *Virtual Rubik's Cube*, <http://www.cs.umbc.edu/~sli2/cube/cube.html>.
- [5] *Applet Arcade*, <http://members.aol.com/shadows125/arcade.htm>.
- [6] *Dungeon Disaster*, <http://www.igd.fhg.de/~wiener/java/dungeon/doc/index.html>.
- [7] *Calligrafix*, <http://www.calligrafix.co.uk/horizons/tour.html>.
- [8] Foley J D et al, *Computer Graphics, Principles and Practice*, chapter 11.1, Addison-Wesley Publishing Company, second edition, 1991.
- [9] Foley J D et al, *Computer Graphics, Principles and Practice*, chapter 12.7, Addison-Wesley Publishing Company, second edition, 1991.
- [10] *Java External Authoring Interface for VRML2.0*, <http://vrml.sgi.com/moving-worlds/spec/ExternalInterface.html>.
- [11] *VRML2.0 Binary File Format*, [http://quickdraw3d.apple.com/vrml20/vrml\\_2.0\\_binary.html](http://quickdraw3d.apple.com/vrml20/vrml_2.0_binary.html).

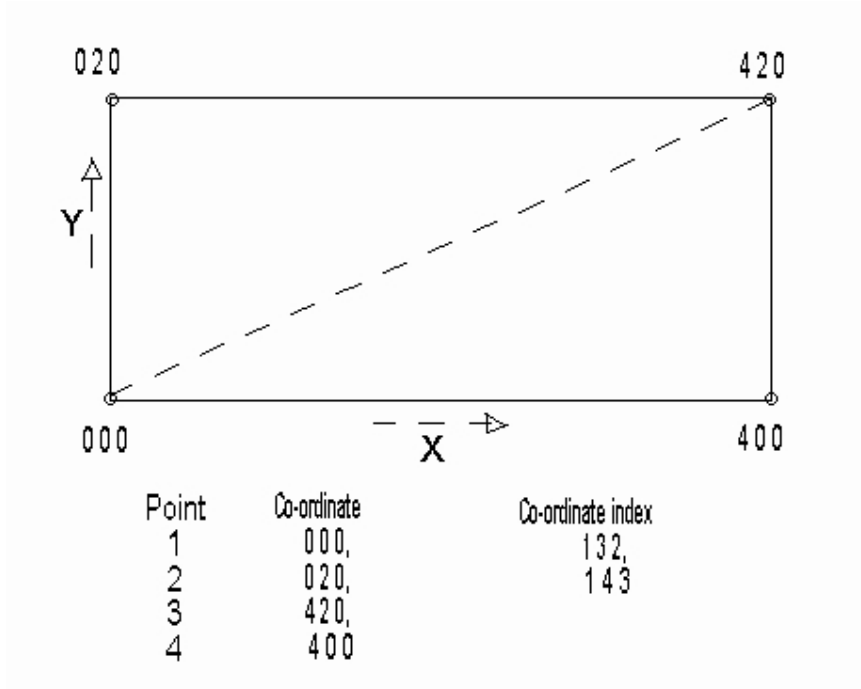


Figure 1: A simple indexed face set

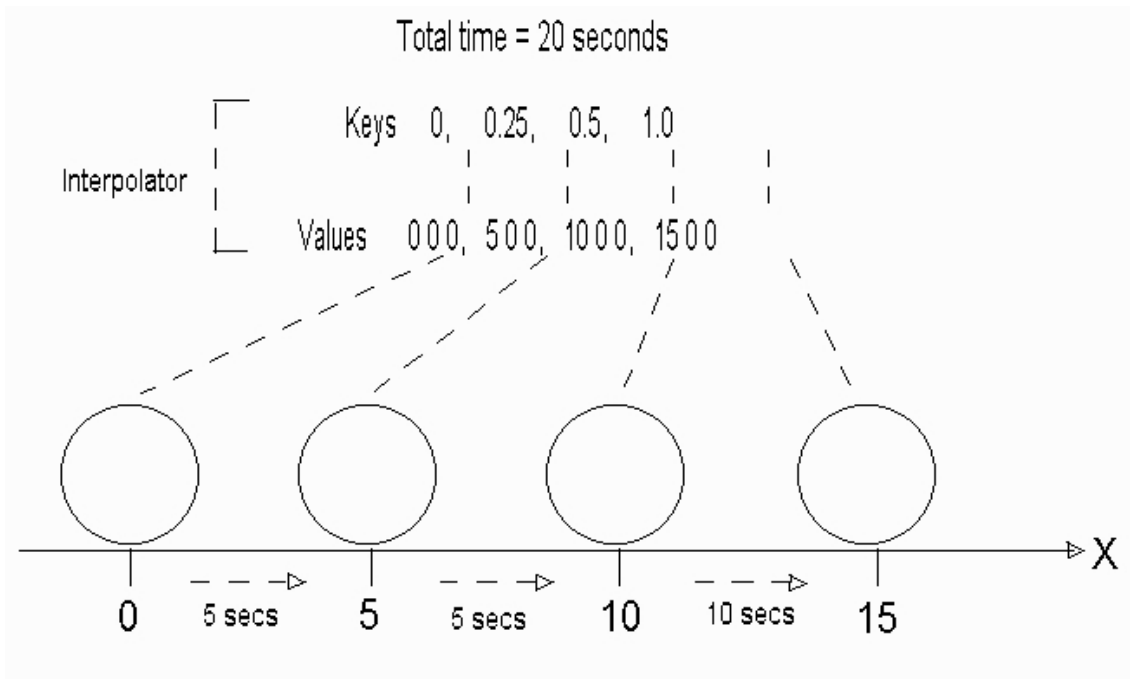


Figure 2: Key-Frame animation



Figure 3: The Mini-Pipe in use

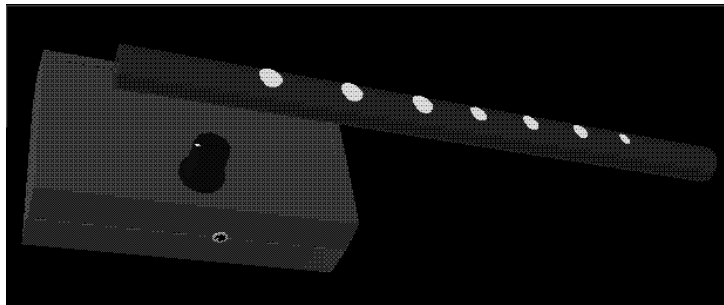


Figure 4: VRML2.0 model of Mini-Pipe

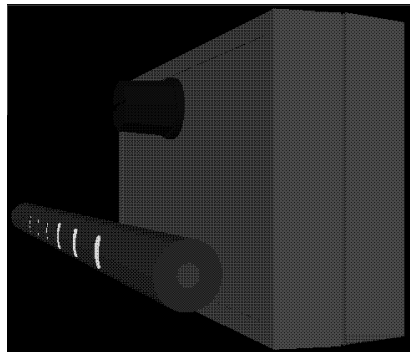


Figure 5: VRML2.0 model of Mini-Pipe

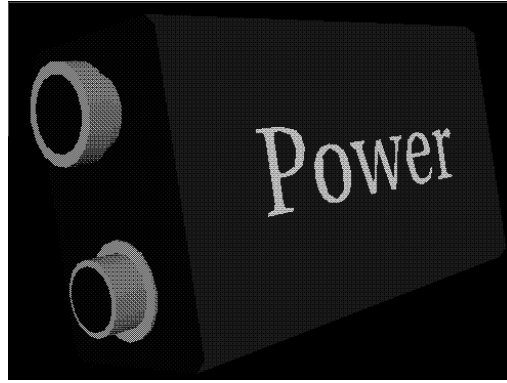


Figure 6: VRML2.0 model of pp3 9V battery

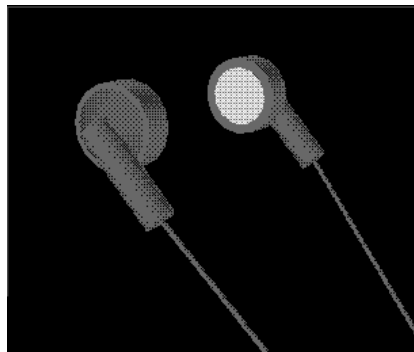


Figure 7: VRML2.0 model of ear-phones



Figure 8: Close-up of finger pad on chanter



Figure 9: The course for the navigation experiment

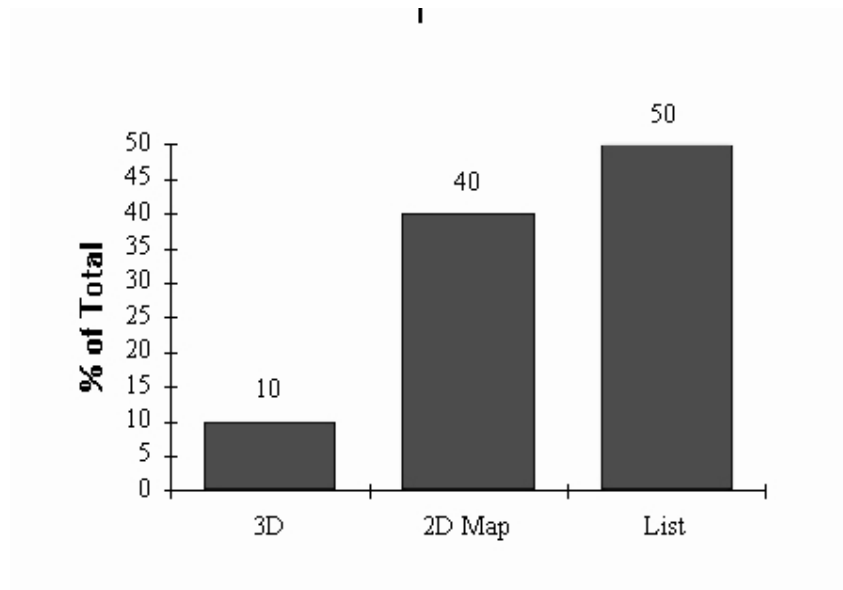


Figure 10: Summary of navigation preference for shopping mall

Sample Size	10
Degrees of freedom	2
Chi-square (observed)	2.6
Chi-square (tabulated, alpha = 0.1)	4.61
Chi-square (tabulated, alpha = 0.5)	1.39

Figure 11: Chi-square for navigation preference

	Lap 1	Lap 2	Combined
Sample Size	9	9	18
Mean Lap Time (mins)	3.98	3.1	-
Sum of Squares	14.86	4.22	-
Variance	1.86	0.53	1.19
Standard Deviation	1.36	0.73	-
S.e.d.	-	-	0.52
t (measured)	-	-	1.71
t (tabulated, alpha = 0.1)	-	-	1.75

Figure 12: Summary of navigation experiment results

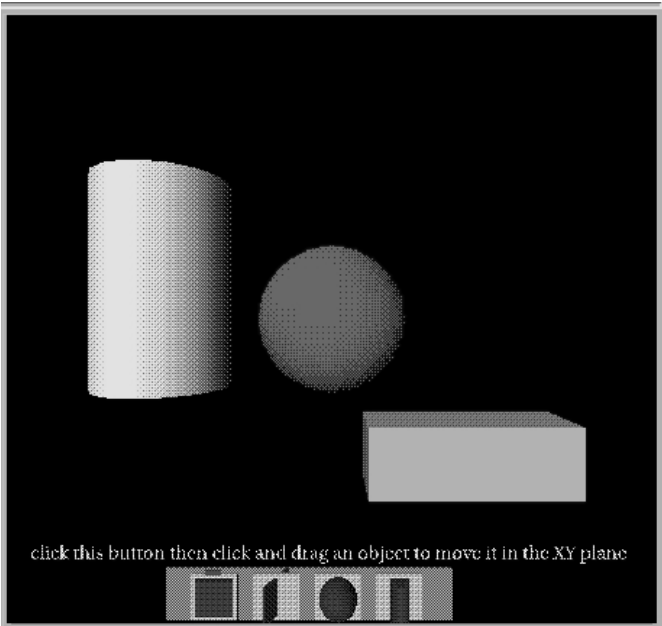


Figure 13: Object manipulation interface, mouse over first icon

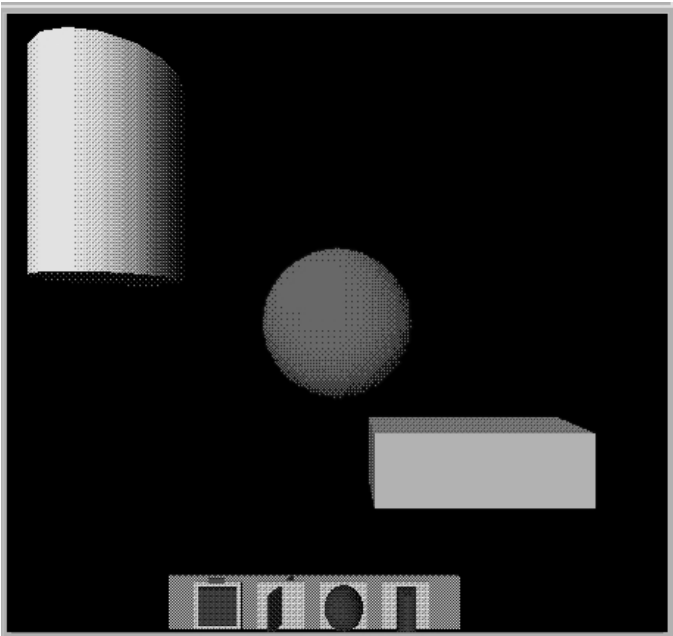


Figure 14: Object manipulation interface, cylinder moved

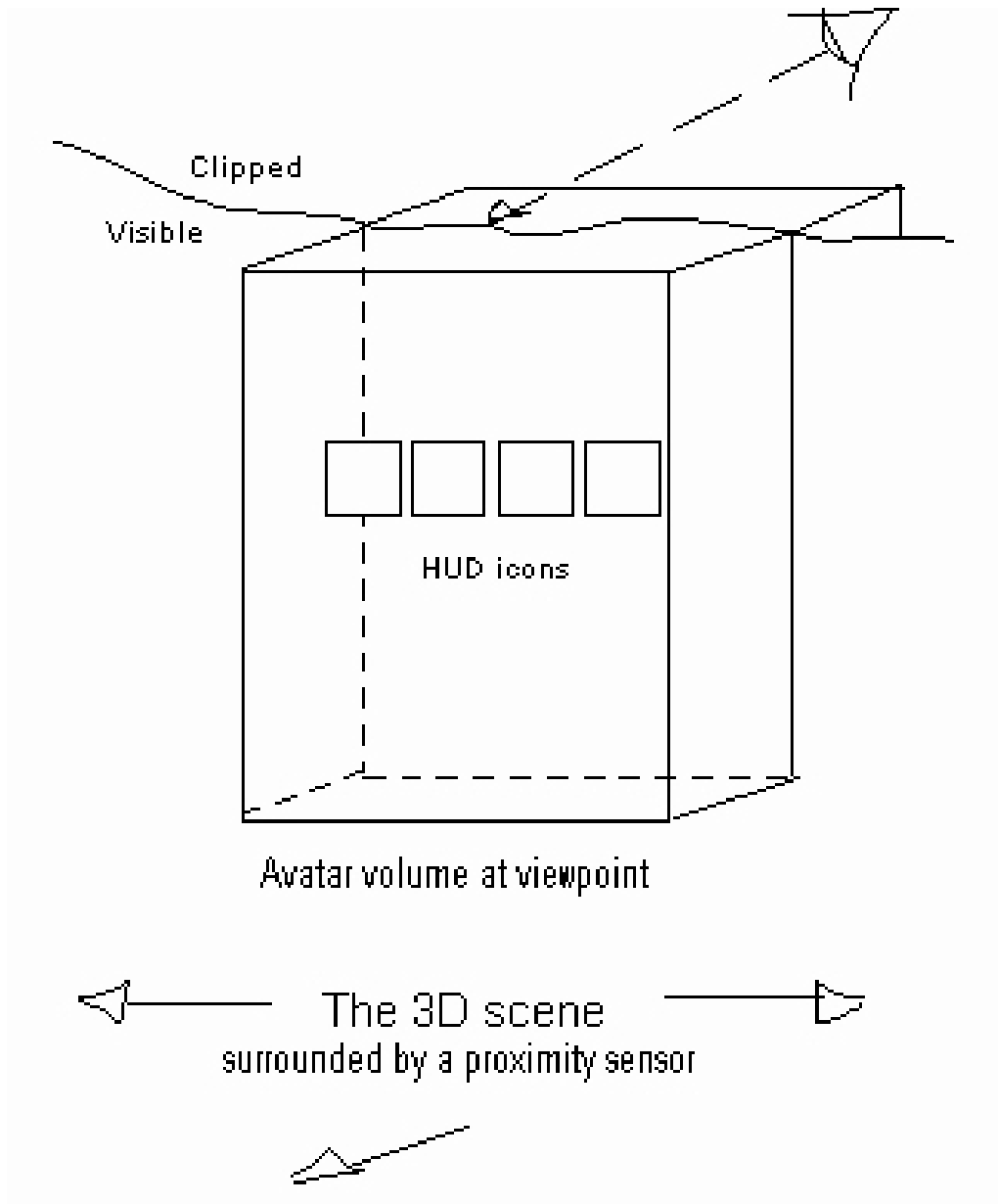


Figure 15: Object manipulation controls inside imaginary avatar volume