

Realistic Virtual Environments Navigable Over the WWW

Ioannis Giannopoulos, Ourania Hatzi, Mara Nikolaidou, Dimosthenis Anagnostopoulos

Harokopio University of Athens, 70 El. Venizelou Str, 17671 Athens, GREECE

Keywords: Virtual reality, VRML, Virtual Environment, Virtual Tourist, Virtual Scene.

Abstract

Creation of 3D virtual environments (VE) can be performed using a variety of design tools, most of which create 3D virtual worlds using high definition graphics, 3D Game engines and smooth animation techniques. The produced result is realistic and eye-catching. Although a VR model is always less detailed than the real world artifact it represents, VR provides greater freedom in viewing a model than physical constraints would allow. It provides the ability to zoom in and examine details that could not be distinguished by the naked eye. However there is a major drawback: to ensure at least average navigation performance everything has to be accessed locally on a single workstation. In this paper, we explore the benefits of 3D web modelling languages available for creating 3D VEs and especially VRML. To effectively exploit provided capabilities a number of techniques were developed to reduce file-size and design complexity, thus producing 3D VEs holding the appropriate level of realism and navigability, making them efficiently accessible via the World Wide Web.

1. INTRODUCTION

3D web modelling languages may be used in conjunction with the World Wide Web to create 3D representations of complex scenes such as illustrations, product definitions and virtual reality presentations. Virtual Reality Modeling Language, (VRML) [1] is such a language for describing 3D interactive worlds and objects. VRML, as well as X3D [2], extend existing hypermedia nature of the WWW, providing ordinary users with the ability to use interactive 3D models as just another form of media. This idea is exciting for both the designer and the audience. However, a detailed 3D virtual world will take a long time to be loaded into the web browser, making it sometimes impossible to view over the WWW. This is a key concern for realtime VR 3D modellers, since VRML and X3D are based on computationally intensive 3D graphics and may incorporate other resource-intensive media. A Virtual World needs to be designed in such a way that its average loading time should not exceed the average loading time of a simple web page [3, 4].

The work, presented in this paper, aims to create 3D Virtual Environments (VEs) that:

1. Facilitate the user with essential information about the environment they are in.
2. Produce appropriate realism for every scene.
3. Speed up navigation.
4. Can be effectively accessed through a workstation connected to WWW.

The rest of the paper is organized as follows: In Section 2 important problems faced in the design process of a VE are discussed. In Section 3 related work is presented, while in Section 4 we propose a way of solving most of the problems identified, through a design methodology that facilitates fast execution of VR Worlds with adequate realism. The benefits of the proposed approach are discussed in Section 5 along with a case study. Conclusions reside in Section 6.

2. VIRTUAL ENVIRONMENT ASPECTS

Creating a VE for the web in 3D is not trivial. Although it is considered simple enough to create low complexity 3D objects or scenes such as simple shapes, in any other case creating a VE cannot be considered a tedious process. The main problem realtime VR 3D modellers have to cope with is the resulting file size of the environment which directly affects loading time and navigation [5]. A number of parameters have to be taken into account in every step towards the creation of a VE, giving special attention to file-size and detail. These two key parameters can determine if it is possible for a VE to be viewed over the WWW with adequate realism. VRML and X3D applications mostly target the WWW. Therefore, they have to be considered '*light*', meaning minimum possible file-size and complexity. The aim in this work is to create VEs accessible via the WWW, by exploring alternatives for the above mentioned issues, without limiting the amount of information and detail given to the end-user.

3. RELATED WORK

The work presented in this article is considered as a continuation of the work carried out by A. Triantafyllou [6], where buildings of the Harokopio University campus were modeled using VRML (Figure 1). Although the Virtual World created is quite detailed and holds a satisfactory level of realism, the implementation techniques used resulted to a VE that is almost impossible to be viewed over the WWW, because of its large file-size. The motivation behind this work was to use innovative techniques in order to reduce the file-size, but, at the same time, preserve the realism of the VE in order to make the University campus and more specifically its buildings accessible through the WWW.



Figure 1. Harokopio University of Athens VRML Building.



Figure 2. VRML representation of single objects.

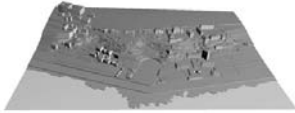


Figure 3. AUB VRML campus representation.

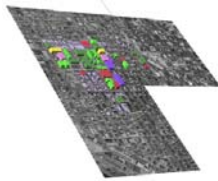


Figure 4. De Paul University campus map.

Although there are numerous examples of VEs enabling VRML and X3D on the web, in most cases, VRML and X3D are used to model simple objects that can hold enough detail to represent a real-life model in 3D. Models like these are used for example by online shops in order to enable customers to have a 3D view of their products (Figure 2).

However, approaches that attempt to represent entire Virtual Worlds which include more than one object are of more interest. Examples of VRML Worlds for University campuses are discussed in the following. A virtual world such as the ones mentioned above is a 3D campus tour created by the American University of Beirut, AUB [7]. Their implementation is a VRML 3D map of the campus providing information through objects linked to HTML pages (Figure 3).

Figure 3 reveals that this attempt is a simple 3D representation with no realism. However, its file-size is 770KB and over a simple PSTN connection it takes about 155 seconds to actually load.

Another similar example is that of De Paul University [8] (Figure 4). In this case, the file-size is about 1.2 MB, but no significant increase of realism is observed. Its loading time over a simple PSTN connection, is estimated over 250 seconds.

An interesting work is that carried out by the Nanyang Technological University in Singapore (Figure 5). In this case,



Figure 5. Nanyang Technological University VRML Campus.



Figure 6. Harokopio University VRML Campus. Realism is easily observed from every angle.

more realistic features [9] are presented but due to the complexity of the implementation, loading time exceeds 60 secs, measured over a (DSL 768) connection.

Our goal was to provide a Virtual World (Figure 6) with a higher level of realism for every scene, which is far more elaborate than existing examples. Moreover, despite the realism and detail, the file-size should be preserved small, reducing the loading time of the Virtual World to that of simple web pages.

4. PROPOSED METHODOLOGY

The loading time of the VE must not be significantly larger than that of an average website. Since web users are accustomed to wait until the website or web page fully loads before they actually start browsing it, and loading time of an average website, is about 25 – 30 seconds, the loading time for a VE should not be over 60-80 seconds.

In addition to file size, ease of navigation also depends on the complexity of 3D objects, that is, the amount of polygons a 3D object holds, or the lighting technique used to illuminate it. Combinations of complex 3D objects constitute complex 3D scenes in a virtual environment. Viewing such complex scenes on workstations with limited hardware capabilities, such as slow processors and conventional inexpensive graphics cards, might result in slow rendering of the scenes, and therefore hinder smooth navigation. The same facts apply to other types of viewing devices with limited hardware capabilities, such as PDAs.

A practical development methodology is proposed, which incorporates discrete steps that can help a VR 3D modeller minimize both overall complexity and file-size of a VE. It consists of 8 steps that are analytically described in table 1. The specified order of the proposed steps optimizes the modelling procedure and avoids unnecessary effort. In the last steps of the methodology realism is progressively added. Therefore

Table 1. Proposed methodology steps.

Proposed Methodology		
	TITLE	DESCRIPTION
STEP 1	Studying the real World	Examine Real Life Model
STEP 2	Single Component creation	Produce single components designed purposely with appropriate detail
STEP 3	Joining components and Inlining	Add VR components in predefined order
STEP 4	Textual Representation	Use of Textured text vs. VRML text
STEP 5	Lighting the Virtual World	Use less light sources, select light location purposely
STEP 6	<i>Adding Realism</i> User Interaction	Add user interaction without increasing complexity
STEP 7	<i>Adding Realism</i> Use of Readymade 3D objects	Reducing complexity in readymade objects
STEP 8	<i>Adding Realism</i> Interactivity through animation	Adding animation

the modeller can determine which realism elements to include, taking into account file size and complexity.

4.1. Studying the real world & trade-off between realism and complexity

First of all, the actual world to be modeled should be examined thoroughly, in order to identify all those features considered important enough to be represented into virtual world. In simple words “what should be included in the virtual world and what should be left out” [3] should be decided. The benefits of this analysis are: First, design time is minimized since fewer objects have to be created. Secondly, by excluding objects of minimum importance from our world, overall complexity is decreased without losing useful information.

4.2. Single component creation

It is important for the implementation of every VE to find ways so that each component, from a simple chair up to a building floor, will be small in file size but at the same time carry adequate detail information. To achieve that, every component has to be designed with a specific purpose.

For example, if a whole building structure has to be designed, and each building has more than one floor, an attempt to divide the building into floors would be very useful.

In order to achieve an adequate level of detail [10], instead of using basic VRML building blocks (4 basic VRML Shapes) the designer can use, if possible, the architectural data and implement each building component Floor with the help of a CAD tool. As a result a very detailed representation of each Floor is achieved, (Figure 7). Note that at this point no single-part objects have to be created. Objects such as windows, doors, furniture, can be implemented at a later time. Designing these objects using CAD would result to enormous object complexity.

Although using a CAD tool can generate somewhat complex 3D objects, sometimes this approach is preferable since it provides us with more accurate objects (see Figure 8) .

4.3. Joining components & Inlining

Having created each floor as a separate object, one should join each 3D component in such a way that every part of the VE will load in a predefined order. For example, consider a virtual tourist standing in front of a multi-store building at the main entrance (Figure 9). In order to maintain the impression

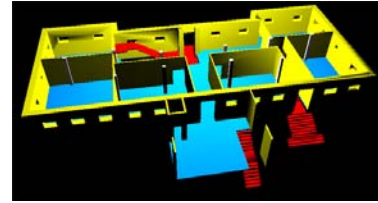


Figure 7. Harokopio University Building floor implementation. Designed in AutoCAD and converted to VRML after polygon reduction.

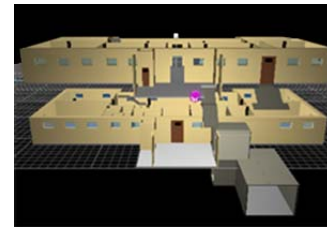


Figure 8. Composing Harokopio University Building floors: Joining two VRML Components. Due to AutoCAD design detail, high accuracy is achieved.

that the whole building is there in front of him/her, each floor is loaded in a sequence, starting from the bottom and ascending. The virtual tourist can start navigating through the first floor until the rest of the building components are fully loaded into the VE above them, giving the false impression that the Virtual World is pre-loaded, while in fact only a small visible part is loaded.

This can be achieved by *inlining* each object into the VE. However, the designers should not attempt to inline the so called *big chunks* of a virtual world. Instead, every component should be analysed and an attempt should be made to determine which parts are important to be visible initially, in order for them to be loaded into the world in the appropriate order.

As an example, consider a Building floor consisting of: the building itself (the actual structure), doors, windows, furniture and office information (door signs, billboards etc.). The best way to define inline order is to determine what the virtual tourist should see first, and what would help increase his sense of realism for every scene. Therefore, considering

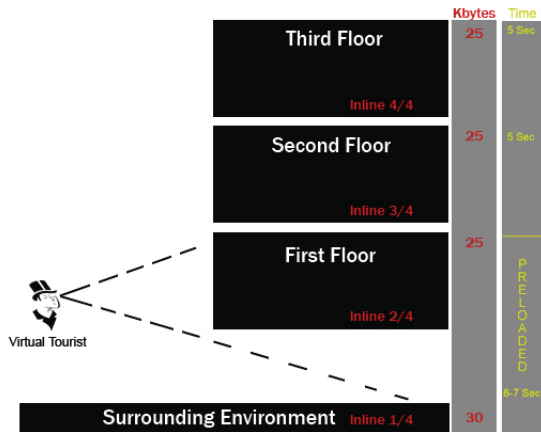


Figure 9. Inlining Objects.

the above example, the correct order should be the following: a)The building itself, b)Office information (door signs, billboards etc.), c)Windows, d)Doors, e)Furniture.

Inlining objects into a VE is a positive step towards making the VE more realistic, since this action helps the designer present users parts of their world in a predefined order.

4.4. Textual representation

Presenting text in VR is most commonly done through 3D models of letters. As part of this methodology, such 3D produced text is avoided since it adds complex objects in the VE. While testing possible solutions to this problem it was decided that using images as textures instead of actual 3D shapes is most efficient. Consider using 3D shapes to produce text for each door of every room for every floor of every building. The file-size would be quite large because of the complexity of each shape, thus loading time will be increased. Furthermore it would make it really hard for a non-VR modeller to make any type of changes.

For example in the case that the occupant of office A changes office and now occupies office B. This means that part of the VRML code for every office door sign should change in order to hold information about the new occupant. The proposed method facilitates:

- Simpler Design (Non – VR, but easily embedded)
- Significantly smaller file-size
- Less loading time
- Less rendering complexity

Firstly, a two layer image template containing (a) a background colour layer and (b) a text layer, is created. This can be done, using any image editor such as Adobe Photoshop. Secondly, the image template is stored a number of times as a JPEG image containing different textual representation for every nametag in the VE. Every JPEG file is then compressed to reduce its file-size even more and is assigned an appropriate

Table 2. Template codenames.

Building A – Office 1	BA – O1.jpg
Building B – Office 1	BB – O1.jpg
Building A – Notice Board 1	BA – NB1.jpg

Table 3. Size of text in VRML vs. JPEG.

VRML FILE	LECTURE ROOM 1	128 KB
JPEG FILE	LECTURE ROOM 1	9,1 KB

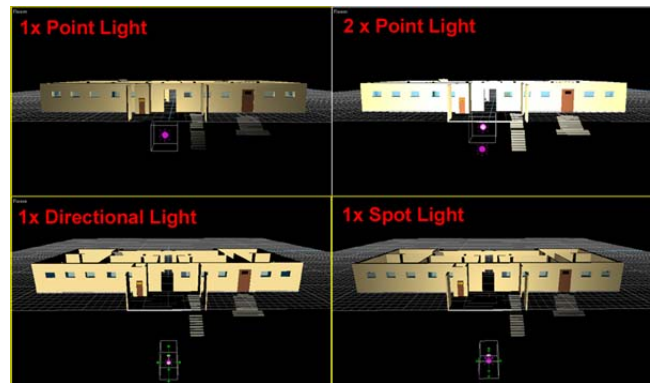


Figure 10. Harokopio University Building ground floor: Possible lighting techniques.

filename (Table 2). For example, at table 2 we get an idea on how filenames could be used.

Based on this model a moderate computer user holding simple knowledge on image editing can easily alter the image template file in case of a possible name change, re-edit, and just save it with the appropriate filename representing the designated sign (nametag). This has as an effect the immediate change of the texture on the VE. This method decreases the file-size because no complicated shapes are being introduced into the VE, and the size of a compressed JPEG image file is significantly smaller (Table 3).

4.5. Lighting the virtual world

Lighting can accomplish a number of things, as establish a line of viewing and focus the users' attention on what is coming up next, or on particular objects. VRML and X3D provide three lighting possibilities: Point, Directional and Spot Lighting. An example is given in Figure 10.

Any of those can be used more than once in a VE, but as seen from the work done in this project, there is a very important aspect needed to be taken into account, which relates to hardware performance on every workstation viewing or rendering the VE.

Multiple light sources increase scene complexity, making it

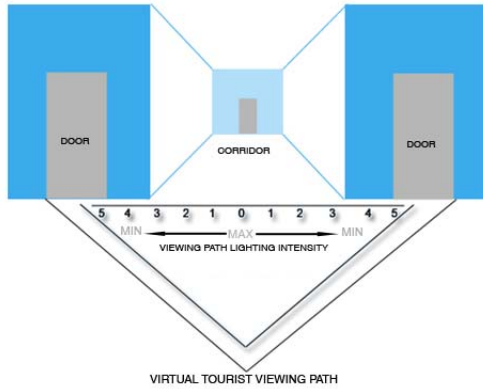


Figure 11. Directional Light on top of Avatar.

harder for inexpensive video cards to display/render complex scenes. This has as a result an increase in the loading time of every scene, regardless of the VE being accessed via the WWW or locally through the user's workstation, thus making it almost impossible to navigate with ease in either case.

An alternative way of lighting up a Virtual world without diminishing the realism of each scene is a point light source as a headlight on the virtual tourist (that is, a directional light placed on the head of the virtual tourist, illuminating towards the viewing direction). Therefore, the walking path of the virtual tourist is correctly illuminated (Figure 11) without overloading the computer hardware responsible for rendering the scene with excessive rendering commands. This facilitates accessibility by a number of systems with the minimum requirements possible.

4.6. Adding Realism – Texture Mapping

A first step to increase realism of 3D VEs is the use of *textures*. Texture mapping is the addition of images, called *textures*, to surfaces in the 3D model, so that the surface takes on the appearance of the image. It is possible to map texture to any surface of a 3D object. Consider adding vegetation to a VE without the use of textures. The only solution could be a 3D object painted green. Using textures we can achieve much more, since an image texture can be duplicated a number of times in order to cover a whole 3D object. We can use a large number of textures to increase realism but we have to ensure a small texture size and a final compression to images. Texture can also be used to substitute 3D objects of minimum importance.

4.7. Adding Realism – User Interaction

It is very important to allow users to interact with the VE, making it easier for them to locate points of interest. One way of achieving this is to provide predefined navigation paths along which the virtual tourist is transported to the desired location, after a selection process through a graphical interface providing all possible routes. An alternative is to provide the

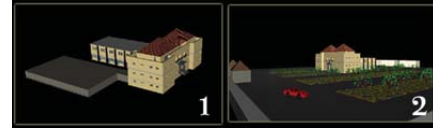


Figure 12. In order to ensure realism, the surrounding environment of Harokopio University Campus is added.

user with all necessary directions to the point of interest without in fact transporting them to the desired location. In this case the user will get acquainted with the VE but at the same time will be able to visit multiple locations. Directions can be provided to the user through 3D objects linking to HTML pages.

4.8. Adding Realism – Use of readymade 3D Objects

In every environment, there are objects that help the user increase his sense of realism. In the case of a building, a factor that increases realism is its surroundings, for example a garden consisting of vegetation, parking spaces, fences etc.

All of these objects are hard to design using primitive VRML shapes, but can be easily found, or even designed, using high definition 3D design tools, for example AutoCAD or 3D Studio. Therefore it is proposed to convert them to VRML or X3D, and compress them in such a way that the minimum number of polygons will be displayed making the object simply recognizable (Figure 12). Thus more objects can be in the VE without significantly affecting file-size.

4.9. Adding Realism – Interactivity through Animation

A simple way to achieve interactivity is animation. For example, in a building, instead of leaving all doors open so that a virtual tourist can move through them, animation can be added to each door. The Virtual Tourist interacts with each door by clicking on it to open every time they need to enter a room, or even when having to move from corridor to corridor inside a building. Another alternative is to use elevators in a building enabling the user to enter an elevator and press the appropriate destination button to move to the desired floor.

Interactivity can also be achieved by adding moving avatars [11], Simple animations can be achieved in most cases with little effort, and they can help increase realism and interactivity to an adequate level.

5. HAROKOPIO UNIVERSITY CASE STUDY

Based on the proposed approach the 3D Environment representation of Harokopio University campus was implemented (Figure 13). The VE is designed in such a way that a virtual tourist can visit the University through the WWW, navigate around the University grounds and find their way to any office, lecture theater, laboratory etc.

The entire VE implementation can be viewed remotely from any workstation regardless of its processing and graphics rendering capabilities, connected to the WWW even at the



Figure 13. Harokopio University Campus: Snapshot of the VE after applying all methodology steps.

lowest speed of ≈ 56 Kb/sec (PSTN Internet Connection). In this case the loading time does not exceed 45 seconds while it is reduced to 15 seconds over a 1Mbps DSL connection. The realism of the implemented VE is relatively high, as can be seen in the corresponding figures included in section 4. This was achieved by using basic textures, 3D CAD objects, animation, appropriate lighting and accurately designed objects. Applying the first step of the proposed methodology the following elements of the real world were excluded from Harokopio University VE: buildings that do not belong to the campus, the sky, lighting poles and other non-defining objects. Next step was to create each related component separately, using AutoCAD. Furthermore, the components created in AutoCAD underwent polygon reduction and were joined easily due to the aforementioned accuracy (third step). The implementation of the textual representation step included processing nametags and billboards in image editing software (fourth step). In order to illuminate the virtual environment a headlight was used as the main source of light (fifth step). Adding realism required introducing animation and adding 3D ready made objects, especially for the surrounding environment (Figure 13).

For designing such an environment a variety of implementation tools which help the modeler visually design the VE are required. Some of the tools used for the particular project were *Internet Space Builder Version 3.01* by Parallel Graphics, which visually designs VRML components, without requiring the modeler to write any VRML code, and *Cosmoworlds version 2.0* by Silicon Graphics Inc., which is used for the same purpose as the previous tool, as well as to add textures and animation. Furthermore, various graphical tools that have the ability to export their files to VRML were used. Examples of such tools are *AutoCAD*, where building floors were initially designed, and *3D Studio*, which provided a number of ready made objects used in the virtual world to enhance realism. Moreover, image editing tools were used, like *Adobe Photoshop CS2* and *Corel Draw*, which performed all image editing. Using a variety of tools is part of the methodology since each of the above holds an important role towards the final result.

6. CONCLUSION AND FUTURE WORK

In this paper, a methodology accommodating the creation of 3D VEs navigable over the WWW is proposed. A case study of Harokopio University Virtual Campus demonstrating the potential of the proposed methodology using VRML is also presented. The VEs produced are small in file-size but detailed enough to present all necessary information to the virtual tourist. Furthermore, due to their short loading time and minimized complexity such VEs can be easily viewed through any device, portable or not (PDA, Mobile Phone etc.), with minimum processor and hardware needs, just using a 3D VR viewer/browser. The refinement of the VE can be performed in such a way that interactivity will be elevated to a higher level, allowing information such as audio and video to be introduced into it, and make the environment accessible through portable devices, such as PDAs and mobile phones, is currently explored.

Regarding the Harokopio University Virtual Campus, the conversion of the virtual environment to the X3D standard is considered as a future step, since X3D is a considerably more mature, refined standard than VRML. A benefit of the conversion is that developers will be able to achieve more sophisticated functionalities.

ACKNOWLEDGMENTS

This research was supported in part by Pythagoras program (MIS 89198) co-funded by the Greek Government (25%) and the European Union (75%).

REFERENCES

- [1] The VRML Consortium Incorporated, "The virtual reality modeling language international standard ISO/IEC 14772-1," 1997, <http://tecfa.unige.ch/guides/vrml/vrml197/spec/>.
- [2] Web 3D Consortium, "X3d international specification standards," <http://www.web3d.org/x3d/specifications/x3d-specification.html>.
- [3] M. Pendlebury and W. Mitchell, "3d virtual reality reconstruction on the internet using vrml," *Masters Thesis*, 1988.
- [4] N. Holmberg, B. Wünsche, and E. Tempero, "A framework for interactive web-based visualization," in *AUIC '06: Proceedings of the 7th Australasian User interface conference*. Darlinghurst, Australia: Australian Computer Society, Inc., 2006, pp. 137–144.
- [5] B. Crispin, "Vrml works the secret of the best vrml worlds on the web," 1995, <http://vrmlworks.crispen.org/essays/tweaking.html>.
- [6] A. Triantafyllou, Postgraduate Thesis, National and Kapodistrian University of Athens, 1999.
- [7] American University Of Beirut, "Vrml campus," tour the AUB campus on a 3D VRML model.
- [8] De Paul University, "Vrml campus," <http://gis.depaul.edu/geography/animap.htm>.
- [9] Nanyang Technological University, "Vrml campus," <http://blaxxun.sc.ntu.edu.sg/csbin/community/print.exe?TPL=common/login>.
- [10] D. L. Baird, "Vrml fractals," <http://www.leemon.com/vrml/>.
- [11] Humanoid animation (H-Anim), "Information technology – computer graphics and image processing – humanoid animation (h-anim)," http://www.h-anim.org/Specifications/H-Anim200x/ISO_IEC_FCD_19774/.