VRML 2.0 Authoring Tool Using SGI's COSMO 3D

Nikitas V. Liogas, Dimitrios Tzovaras and Michael G. Strintzis

Electrical and Computer Engineering Department
Information Processing Laboratory
Aristotle University of Thessaloniki
Thessaloniki 54006, Greece
Phone: (+30-31) 996-359, fax: (+30-31) 996-398
e-mail: tzovaras@dion.ee.auth.gr

ABSTRACT

The paper presents the features of "VRML Author", a VRML 2.0 authoring tool based on SGI's Cosmo 3D and Optimizer public software. The tool is used for the montage of a 3D model scene using a number of already existing 3D model scenes. The developed authoring tool supports VRML 2.0 format for 3D model object insertion in the 3D scene and for interactive manipulation. The resulting 3D scene may then be stored as a VRML 2.0 object. Two possible applications of the VRML Author are presented, i.e. a military simulation tool and a general 3D model scene creation tool for developing fully 3D objects that can be distributed over the Internet for commercial purposes.

1. INTRODUCTION

The potential of recent and ongoing innovations and developments, i.e. increasing media processors and PC capabilities, explosive growth of the Internet and related object-oriented and graphics technologies, exemplified by the JAVA and VRML [1] paradigms, is bringing new perspectives for services' innovations. Indeed, part of the explosive growth of the Web are emerging browsers, which allow not just easy access to text, audio and video, but also, with appropriate plug-ins, the presentation of animated graphics contents, including virtual reality [2]. However, one important problem not yet solved, is the development of appropriate authoring tools for montage and manipulation of virtual reality objects.

Recently, research was focused on mixing any kind of multimedia components. In [3], the issue of 2D and 3D media integration was addressed, involving real-time streaming in a 3D scene, with adequate synchronization, of audiovisual information coded in any format. In [4], "Multimedia Montage" was proposed, i.e. the structural synthesis in time and space of multimedia components, such as movies and sounds, using the counterpoint theory. Finally, [5] presented the design and architecture of distributed virtual reality services, which allow collaboration-unaware VRML Animations and simulations to be integrated into cooperative applications.

The growth of the specific needs for 3D complex object creation and manipulation, led to the need for the development of complex object authoring tools supporting VRML. The paper presents the features of "VRML Author", a VRML 2.0 authoring tool based on SGI's Cosmo 3D and Optimizer public software [6]. The tool is used for the montage of a 3D model scene using a number of already existing model scenes. The developed authoring tool supports the integration of 3D model objects (in VRML 2.0 format) in the 3D scene. The resulting complex 3D scene may then be stored as a new VRML object. Two possible applications of the VRML Author are presented, i.e. a military simulation tool and a general 3D model scene creation tool for developing fully 3D objects that can be distributed over the Internet for commercial purposes.

2. SGI's COSMO 3D AND OPTIMIZER

Cosmo3D is a scene graph API that brings 3D graphics programming to desktop applications. It speeds up and facilitates the process of creating and handling complex graphics applications. Its main advantage is that it allows applications to use a higher-level interface than the lower-level OpenGL language that it is based on.

Developers just have to interact with C++ objects that are arranged in an object hierarchy. After creating a scene graph using Cosmo3D, developers can improve performance by using the OpenGL Optimizer API. This toolkit makes it easier to draw large models interactively through creative uses of culling, multi-threading, level-of-detail (LOD) rendering, and so on. It is built on top of OpenGL, so that a single Optimizer command calls several OpenGL commands.

The benefit is that the Optimizer API already "knows" the most efficient way to organize the data. It provides the highest quality image possible while minimizing the amount of work required by the graphics hardware. This is accomplished by determining what will not be seen and therefore should not be drawn (culling).

3. THE VRML 2.0 AUTHORING TOOL (*VRML AUTHOR*)

VRML Author is actually a VRML 2.0 object manipulation tool. It is built on SGI's (former Silicon Graphics Inc.) 3D API, Cosmo3D/OpenGL Optimizer 1.1.

The program itself is based on the cooperation between the MFC library and the Cosmo3D class library. All the messages received by the client window are handled by an override of the CView::WindowProc() method. In this method, after defining the corresponding device context and color palette, message-handling code, using both MFC and Cosmo3D methods, is utilized, in order to manage multiple VRML objects.

One of the most important Cosmo3D methods used, is csDrawAction:: apply(), which in fact causes the complete scene graph to be rendered, when applied to the scene graph's root node. The program calls this particular method every time a WM_PAINT message is generated, which means that the client area must be redrawn. Every action taken by the user i.e. scale, zoom in, object insertion, triggers the appendage of a WM_PAINT message to the window message queue, so that the effect of the action is realized through a "refreshed" scene graph.

Moreover, the context menu, used for quick access to the program's features, is associated with the CMainFrame class, the project's CFrameWnd-derived class. As a result, every context menu command is handled by a CMainFrame method, which uses the data provided by the MFC framework, namely the position of the clicked point, to implement the supported features, i.e. the insertion of the new object at the place indicated by the user. When the first object is inserted, a new scene graph is created, containing various kinds of nodes including transform, light and sensor nodes. In addition, a viewing camera is defined. The window's device context is bound to Cosmo3D's context using the constructor of csContext and the method csContext::makeCurrent().

Finally, the object is positioned in front of the camera, so that it is visible by the user, and its coordinates are recorded. From that moment on, every new object inserted into the scene uses the coordinates of the first object to position itself correctly in front of the camera. Moreover, the application uses the coordinates, recorded for every object, to decide whether there is indeed an object at the point where the user clicked, and thus make the corresponding choices available. For example, one can zoom in at any point of the scene, but can only scale one of the existing objects.

The features implemented in the VRML Author authoring tool reflect the power embedded in Cosmo3D and OpenGL Optimizer APIs. First of all, the user can load a VRML 2.0 object and insert it anywhere in the scene, just by pressing the right mouse button and selecting "Insert…" from the context menu. After selecting the desired *.wrl

file, the object appears at the indicated point ready for manipulation. Objects can easily be removed from the 3D scene by selecting "Remove" from the context menu. The tool only supports VRML 2.0, since VRML 1.0 and 2.0 formats are incompatible.

After inserting an object, the user is able to perform various actions on it such as 3D translation, rotation and scaling. All operations may also be performed by dragging the object to the desired location. Alternatively, the context menu, that pops up whenever the user right-clicks on an object, may be used to translate, rotate or scale it, by entering the desired values in the appearing dialog boxes. One can enter the new location as x,y,z coordinates (translation), or the desired rotation as x,y,z components of a rotation axis, a rotation center and an angle. Moreover, one can scale the object along the three axes, by different factors along each axis if so desired.

Another important feature is zooming. The user can zoom into any part of the scene by a factor of his choice, entering the desired value in the appropriate dialog box. The point where the user clicks automatically becomes the new center of the scene, so that one is able to focus his attention on a specific place by constantly zooming into it. "Zoom-out" works in a similar way. The user enters the zoom-out factor (in %) in the appearing dialog box and the clicked point becomes the new center, similar to the implementation of the "zoom-in" feature.

Finally, a first version of a "Save" feature has been developed, so that one can save the scene graph into a VRML 2.0 file, which could be re-inserted in the VRML Author and manipulated anew. However, there still exist problems in this feature that will be faced in the next version of *VRML Author*.

Apart from these basic operations, which are already available, many other are under construction. An object toolbar will be implemented allowing for drag-and-drop insertion and removal of objects. "Undo-Redo" features will be added allowing for more flexibility in the manipulation. Group manipulation will be made possible, meaning that whole groups of objects could be translated, scaled etc. Of course, printing and print preview will be available in the next version of the tool.

4. APPLICATIONS

4.1 Military Simulation Tool

This project application was developed under the auspices of the Greek Army. It is intended to become a tool for simulation of the Field of Operations, used by generals and other officials in the decision-making, during a wartime situation. Nevertheless, its structure allows it to be a VRML authoring tool as well as a simulation tool, with pioneer capabilities and functions.

Figures 1, 2 and 3 show the stages of inserting existing VRML 2.0 objects (the helicopter and the bridge) into the scene graph and the creation of a more complex VRML object.

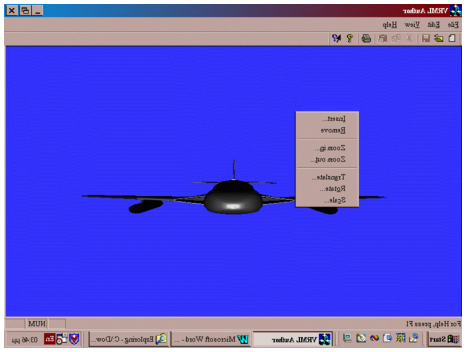

Figure 1. Military Simulation Tool: Only one VRML object is inserted in the scene.

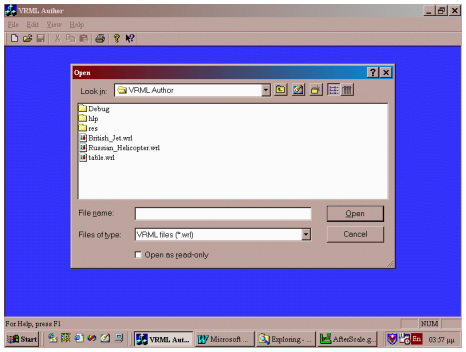

Figure 2. Military Simulation Tool: Dialog box for inserting a second VRML object.

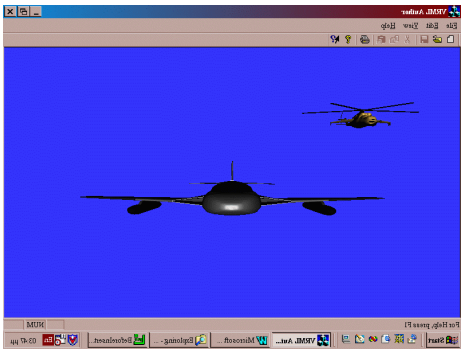

Figure 3. Military Simulation Tool: The 3D scene after the insertion of the VRML 2.0 object "helicopter".

More objects may be added and manipulated interactively, producing thus complex objects, simulating an air fight, as seen in Figure 4.

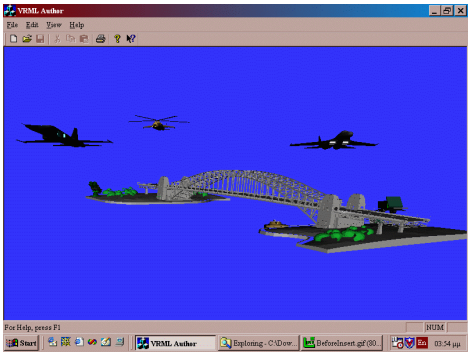

Figure 4. Military Simulation Tool: A complex 3D scene synthesized using *VRML Author* and the VRML objects "Plane", "Helicopter" and "Bridge".

4.2 Virtual Shopping Application

VRML Author may be used for commercial applications also, such as virtual shopping and advertisement over the Internet.

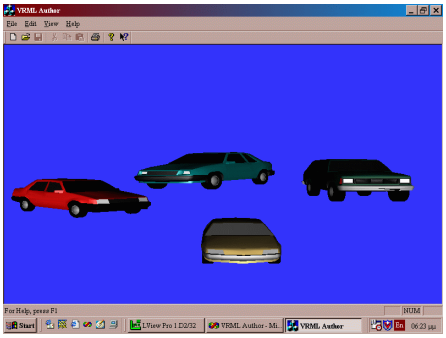

Figure 5. Virtual Shopping: Virtual car shop.

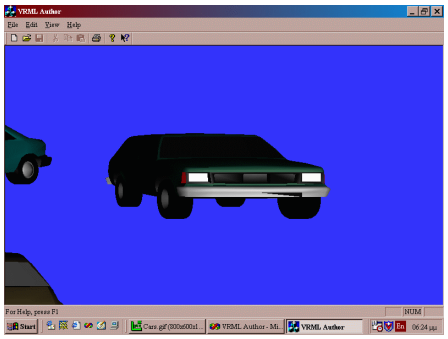

Figure 6. Virtual Shopping : Customers zoom-into a specific product (car).

A complex scene may be created using VRML Author illustrating the products for sale in 3D. The 3D scene may then be stored in VRML 2.0 format and offered to possible customers over the Internet. Figures 5 and 6 show such an application for a virtual car shop.

5. REFERENCES

[1] Virtual Reality Modeling Language, *ISO/IEC WD 14772*.

[2] J. D. Folley, A. van Dam, S. K. Feiner, J. F. Hughes, ''Computer Graphics Principles and Practice'', *Addison Wesley Ed.,* 835-842, 1995.

[3] C. Bouville, C. Guillemot and R. Mallart, "A MPEG4 - MSDL Viewer Implementation", *IWSNHC3DI'97*, Santorini, Greece, Sep. 1997, pp. 16-19.

[4] R. Suzuki and Y. Iwadate, "Multimedia Montage – Counterpart Synthesis of Movies", *IEEE Multimedia Systems '99, Florence*, Italy, Jun. 1999, pp. 433-438.

[5] W. Geyer and M. Mauve, "Integrating Support for Collaboration - Unaware VRML Models into Cooperative Applications," *IEEE Multimedia Systems '99, Florence*, Italy, Jun. 1999, pp. 655-660.

[6] Cosmo3D Toolkit, *http://vrml.sgi.com/Products/cosmo/cosmo3D*.