

Application Scenarios for Scientific Visualization and Virtual Reality Using a CAVE Infrastructure

Ch. Stern¹, H. Noser¹, J. Weissmann¹ and P. Stucki¹

¹ MultiMedia Laboratory (MML), Department of Computer Science, University of Zurich, Zurich, Switzerland

Abstract

Today's Virtual Reality applications require a high degree of immersion. This may be achieved by using a CAVE incorporating powerful 3D computer graphics workstations, panoramic multiple screen displays and projection walls, stereovision and interaction devices such as trackers and data gloves. This article describes Virtual Reality applications from the fields of scientific visualization in medicine and education embedded within the Blue-C environment, a state-of-the-art CAVE infrastructure.

Keywords:

Scientific Visualization, Virtual Reality, CAVE, Virtual Endoscopy, L-Systems

Categories and Subject Descriptors (according to ACM CSS): H.5.2 [User Interfaces]: Input devices and strategies; I.3.7 [Computer Graphics]: Virtual Reality; J.3 [Computer Applications]: Life and Medical Sciences

1. Introduction

It is an objective of our research activities to design, specify and develop state-of-the-art software components and applications enhancing the scientific impact of the Blue-C project [1]. Due to our cooperation with the University Hospital of Zurich, deep knowledge and corresponding software frameworks in the fields of medical imaging, interactive 3D visualization and Virtual Reality (VR) are already available. The application frameworks presented within this article make use of standard hard- and software components. For rendering, the Silicon Graphics Onyx graphics workstation family was employed. The *OpenGL* Application Programming Interface (API) was used to access the *Infinite Reality* graphics subsystem.

2. Immersive Virtual Endoscopic Viewing Using the Blue-C Infrastructure

Virtual Endoscopy (VE) is a widely used method for performing screening and diagnostic activities in radiology. It involves reconstructing 3D virtual medical objects from Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) images using iso-surface generation or volume rendering methods. In the present work state-of-the-art VR techniques are used for conducting non-invasive

diagnostic tasks, such as detecting small tumours contained within the lumen of the colon, visualizing brain tumours or placing and measuring virtual stent grafts within aortic aneurysms [2]. The software framework specified within this project is also intended to be used as a positioning evaluation system for patients being scanned via MRI or CT devices. In order to achieve the project's goal efficiently, modern software technologies and methods, such as distributed computing and object oriented programming are incorporated.

Our software environment for the 3D visualization of anatomical objects reconstructed from CT and MR image datasets has been successfully ported to the Blue-C infrastructure at ETH Zurich. It allows to perform Virtual Endoscopy sessions using a panoramic projection with an enhanced degree of immersion. To determine the appropriate viewing frustums for splitting the computer generated scenes among multiple viewports and screens, the formulas adopted from [3] were used.

The spacemouse offers the ability to navigate within these virtual worlds intuitively, i.e. with six degrees of freedom. Furthermore, we connected a tracking device to our software prototype for Virtual Endoscopy. The tracking device allows the user to control the position and orientation of virtual 3D objects such as Multiplanar

Reformatted (MPR) images in an interactive and intuitive manner.

3. The LWorld Software Environment

In order to fully exploit the hard- and software features of the Blue-C infrastructure, our Virtual Reality framework for timed and parametric L-Systems [4] called *LWorld* [5] supports client-server application architectures and thus represents a distributed system. *LWorld* is a versatile real-time animation system. Designers can define mutable virtual environments, animations, and user interaction by specifying appropriate L-Systems.

The symbols generated by repeatedly applying the production rules are visualized in 3D by using turtle-based graphics. The axioms and rules defining the L-Systems to be evaluated have to be written into a textfile, which is parsed by *LWorld*. L-Systems define the complete virtual environment including user interaction. In *LWorld*, a rule has the general form

```
"REPLACE" LeftSideSymbol "IF" Condition "BY"
"<" Probability RightSideSymbols ">"*
```

It is specified by a left side, a condition and a right side. In stochastic L-Systems, the right side is a list of “probability – right side” pairs, where the sum of all probabilities must yield 1. Here, *LeftSideSymbol* is the symbol to be replaced, *RightSideSymbols* are the symbols to replace it with. This replacement will be made with a given probability if the condition evaluates to true.

3.1. Integrating the Data Glove

In an environment such as the Blue-C, navigation or object manipulation by means of keyboard or mouse is not adequate. We have therefore focused on other means and concepts of human-computer-interaction, particularly gesture recognition. Using the client-server architecture of the *LWorld* system, a Gesture Recognition Module (*GeRM*) based on a data glove, the *CyberGlove*, was integrated to examine the usability of VR navigation and manipulation by means of simple gestures. The gesture recognition system uses a neural network to process the measurements conducted by the *CyberGlove* to determine the particular gesture made, one out of an “alphabet” of 20 different “hand-postures”. In order to endow the gestures recognized by *GeRM* with a semantic, only the rules defining camera navigation had to be adapted. Thus, no recompilation of *LWorld* was required. This fact emphasizes the generic character of the *LWorld* software framework. For instance, the rule

```
REPLACE SI IF condition =
(netData(2) eq 1) and (t>scanTime); BY
< 1 SI {turtx=turtx+fStep*turtGet(3);
turtz=turtz+fStep*turtGet(5);} >
```

states that the viewpoint shall move forward by a certain amount *fStep* if the gesture “Point” is made. The expression *netData(2)* retrieves the second number of the last data package received from the client (the number 1 stands for the “Point” gesture), *turtx* and *turtz* are the global coordinates of the viewpoint's position in the plane and the function *turtGet()* retrieves the x- and z-component of the turtle's heading vector. The expression *t>scantime* in the condition ensures that the time between two executions of this rule is greater than *scantime*. The variable *t* is the local age of the symbol in which it appears, i.e. the time that has passed since the symbol has been created as the result of the application of a rule. Here, *t* refers to the age of the symbol *SI* appearing on the left hand side of the rule.

3.2. The Virtual Park Application

The Virtual Park is a 3D world containing several labeled pedestals with initially unexpanded objects such as virtual plants, trees, butterflies and mathematical objects (e.g. Koch Surface), each of them defined by an L-System. As a visitor navigates through the park by means of the data glove, his vicinity to a pedestal is detected by rules that model a kind of proximity sensor which in turn trigger a set of rules of the L-system associated with the pedestal. Then, the visitor can observe the object's generation and animation. As soon as the visitor moves away from this object it is eliminated. This level-of-detail concept allows the placement of several different L-systems in one virtual world without having to endure the penalty of low frame rates. Using *LWorld*, these autonomous objects and their growth processes may be inspected and tracked interactively. Our preliminary results indicate that the data glove represents an intuitive and user-friendly device to navigate within virtual worlds in a CAVE environment.

References

1. Blue-C – Immersive Environment, Virtual Reality, Collaboration. <http://blue-c.ethz.ch>
2. E. Stoll, Ch. Stern, S. Wildermuth and P. Stucki. A New Filtering Algorithm for Medical MR and CT Images. *J. of Digital Imaging*, 12: 23-28, 1999.
3. A. Dornie. <http://oss.sgi.com/projects/performer/mail/info-performer/perf-99-09/>
4. H. Noser, Ch. Stern and P. Stucki. Distributed Virtual Reality Environments Based on Rewriting Systems. Appears in *IEEE Transactions on Visualization and Computer Graphics*, 2003.
5. <http://www.ifi.unizh.ch/~noser/Lworld/lworld.html>