

Virtual Historical Heritage: Yuso Monastery

María Aróstegui^(), José Ignacio Barbero^(*),*

Carolina Cruz Neira^(#)

^(*) Labein, Mechanics Unit, Bilbao, Vizcaya. Spain.

E-mail: arostegui@labein.es

E-mail: jose@labein.es

^(#) VRAC; Virtual Reality Application Center, Iowa, USA.

E-mail: carolina@vrac.iastate.edu

ABSTRACT

Due to the maturity acquired by the new technologies, the research work developed within the project of the Virtual Reconstruction of the monastery of San Millán de la Cogolla, is focused on the use of VR techniques in very different environments, such as historical documentation or artistic purposes.

This project consists of the development of a VR application with spreading aims. The research work about cultural innovation is one of the main goals.

KEYWORDS: Virtual historical heritage, cultural innovation.

INTRODUCTION

The computer graphics application that recreates the monastery of Yuso, is born as the collaboration of two research centers: Foundation LBEIN (Bilbao, Spain), and the North American Research Center VRAC (Virtual Reality Applications Center, in ISU, Iowa State University, USA).

The aim of this research work developed within the project of the virtual reconstruction of the monastery, is the deepening in using tools of virtual reality and some photo-realistic

techniques. Other clear objective is using these technologies on historical documentation, or on industrial, and artistic environments.

On the other hand, one of the main characteristics of the project of San Millán is the easy portability of the computer graphics application. What is more, due to the conceptualization of the application like a class (using VRJuggler), or in any case like a code programmed with an oriented objects philosophy, the application can be used with all kind of input/output devices, being this way useful with most of the commercial devices. So is that, it can be used with trackers, joysticks, mice and/or keyboard (like input devices), and with DIGS / SSVR systems or Head Mounted Displays (like output devices).

In this paper several topics will be studied; in the first place, the developed tasks along the different stages of the project, will be analyzed. Besides, future implementations will be valued. And to sum up, some of the consequences and the most immediate conclusions derived of the carried out work, will be exposed.

STAGES OF THE PROJECT OF SAN MILLÁN.

First of all, it would be convenient defining what is meant by Virtual Reality. One of the best definitions comes from the “The Silicon Mirage: The art and Science of Virtual Reality”[1].

“Virtual Reality is a way for humans to visualize, manipulate and interact with computers extremely complex data”

The **visualization** part concerns the virtual world generating, so is that, visual, auditory and other sense outputs within the virtual world. The **programming** part refers to the way of manipulating objects within the virtual world, moving from the virtual world, or interacting with the virtual world.

This project has had until the moment two different stages, due to the nature and objectives of the project, that coincide with the **visualization** and **programming** parts. These ones are going to be detailed immediately after.

Modeling and Image Treatment Stage
 The first stage of the project consisted of the 3D modeling of the architectural complex, as well as textures mapping to obtain a loyal real scale reconstruction of the current monastery of Yuso in San Millán of the Cogolla.

Describing the scene within the virtual world, does not only mean describing graphical primitives and geometric objects, but lights and cameras. The Yuso computer graphics application, use a tree data structure allowing new kinds of nodes, such as attribute-definition nodes or matrix transformation nodes.

The real-time model of the monastery used in is defined in terms not only of geometry, but also of structure, or

hierarchy. The Hierarchy is used to organize the components of the database. The Hierarchy is defined by means of nodes, the database items, connected in one or more tree structures.

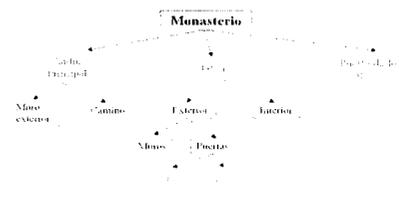


Figure 1- Example of the hierarchical structure of the database.

A scene graph holds the data that defines a virtual world. The scene graph includes low-level definitions of geometrical objects and their appearance, higher-level definitions of spatial information, such as specifying the positions, animations, and matrix transformation of objects, as well as additional application-specific data.

The scene graphs couples perfectly with the object-oriented paradigm supported by languages such as C++. This way we can define classes to manipulate encapsulation to build up scenes of great complexity. We can add animation nodes in the scene graph that provide the needed parameters to change the scene, and even to redisplay it.

Depending on the attributes and the geometry created for each node, they can be classified in two different ways:

- (A) Paying attention to the animation properties inherent in the nodes, they can be divided into:

*SCS, Static Coordinate System,

*DCS, Dynamic Coordinate System

(We defined DCS, when animation properties are attached to that node, and SCS in the contrary case.)

(B) Because of the geometry used to defined them, each node might be:

Geometric nodes; Vertexes, Edges, Facets, Objects, Groups, LOD(Level Of Detail),etc.

Attribute nodes; Transformation nodes, Sound beads, Color nodes, etc.

On the other hand, the Yuso monastery database is built using approx.: 67.000 triangles, 513 group nodes, 816 object nodes, and more than 100 level of detail nodes.

Level of Detail are sets of models that represent the same object with varying degrees of complexity. The real-time system selects one of the LODs to display, depending on the distance from the eyepoint to the LOD and the number of polygons the real-time system can process.

If the eyepoint is far away from an object, the object is displayed as a low LOD, containing relatively few polygons. As well the eyepoint moves towards the object, the real-time system replaces it with increasingly complex LODs. LODs play an important part in optimizing a visual scene for drawing, because there is an effective limit on the number of polygons each real-time system can display.

Another point to be taken into account during this stage, is the texture mapping. Textures patterns are bitmapped images that are mapped onto polygons to give

the Yuso database a photo-realistic appearance, without increasing the polygon count.

To get a good texture mapping within the Yuso database, the measure of the polygons, the texture size, the wrap method, the image resolution, the texture format, and the texture filters, have been studied. For example mapping a 2.5m x10m wall with an 32x32 pixel rgb file, is not the same as mapping the same wall with a 512x512 pixel rgb file. What is more, we have map the wall pondering the bricks size and the user's size in order to get a more real sense in our virtual world.

A very important point in this stage of the Yuso project was the lighting. The correct study of the lights and shades determined what kind of lights were the most accurate to recreate the monastery ambient. For example, instead of using local lights (Omnidirectional finite distance) or spot lights(Lobular finite distance) in the exterior, we preferred to use infinite lights (Omnidirectional infinite distance), whereas in the interior we used diffuse local lights.

Besides, the choice of the light source color and positioning light sources, were points studied too to get better results.

Finally, a good definition of the database hierarchy is needed when the real-time system is rendering, because the better "inheritance concept" is defined, the quicker the rendering can result. Besides, a not correct dependence relationship between nodes "father" and "children" can give place to wrong animations or movements.

Navigation Code

As it has been advanced in the introduction, the project of the virtual reconstruction of the Monastery of Yuso doesn't only have as objective obtaining a geometric modeling of the architectural complex, but programming and designing some algorithms of collisions detection, and navigation codes to fit our necessities. The study of the possibilities that VR offers to the engineering world, or education among others, is one of our priorities.

Several VR software tools were used to develop the Yuso application. It were the VRJuggler libraries (software library for VR application the Performer Libraries development, wholly developed in VRAC), the VEGA software, and obviously the Iris Performer Libraries.

Therefore, the most important details, obtained during the second stage of the Yuso Project, will be explained depending on the software used in each case.

VRJuggler libraries

With the purpose of using the model of the monastery of Yuso in the C2, the VRJuggler libraries were used.

The C2, or CAVE is a VR wholly immersing system where the virtual world is projected in a 4 sides room. The user has the opportunity to visit virtual environments that would not be possible in other case. By way of example, students of art at the Iowa State University have the opportunity to visit a sample of the Spanish Baroque art: the walls of San Millán. They can visit the chapels, or observe all the little details of the main altar, go for a walk for their cloister, or fly to watch the monastery from angles that in the real world would be completely impossible to reach.

The main characteristics of this VR library, and the Yuso application will be explain bellow.

VRJuggler is a library of classes of C++, entirely developed in VRAC, used as a frame for the development of Virtual Reality application. Its object-oriented system is designed to take advantage of the power and flexibility of C++. Most of the functionality of Juggler has been divided into a handful of distinct and well-defined components called Managers. Each manager encapsulates a specific set of system details. For example, there are several Managers to deal with different windowing systems, and others to handle aspects of drawing with different graphics systems. A single Manager oversees all input devices, while another control output devices. Other Managers control networking and system configuration. The Managers are united and controlled by a small kernel, which brokers all communication in the system.

Defining the Yuso application using VRJuggler, meant defining it like a class, allowing to the developer a bigger encapsulation degree, and control on the same one. The class included functions to draw the frames, to proceed on the necessary calculation for the PRE and the one POST drawing, and to manage the synchronization of the system. Using VRJuggler, each application is programmed like a class independently of the hardware devices that will be used. That is to say, VRJuggler provides common interfaces to different types of input devices, and the applications are not tied to any type of hardware device defined by default. What is more, it provides a dynamic reconfiguration without compiling and linking the code, in order to obtain different versions of an application.

Finally, the 19 of June 2000, the center VRAC inaugurated the C6, a three-dimensional virtual environment wholly immersing that consists of a room with four walls, floor and roof, where stereoscopic images are projected providing a total immersion to the participants. One of the DEMOs shows in the inauguration ceremony was that the Monastery of Yuso application.

Vega libraries

VEGA is a high-performance software environment and toolkit for real-time simulations and virtual reality applications. Among their modules, the APIs and the libraries have been used to get the objective that was sought. VEGA is constructed as a layer on top of Iris Performer.

VEGA is a library of C classes, and the Developer's Application is not a class but an executable.

Developing our own VEGA application meant defining the input file and programming the Yuso code, that is to say:

(A) On one hand, the input of a VEGA application is an ASCII file. It contains information about the geometrical model, texture mapping, the type of windows that are displayed, the number of observers and specifications of the illumination that is being used. But the main purpose of the input file is to initialize, define and configure the system.

(B) Once defined the input file, the properly VR application needs to be programmed. Here, the navigation code, the collision detection algorithms, and the definition of all the events such as sound reproductions or lighting are implemented.

The Yuso application, like many other VR applications, handles lots of information and needs an accurate memory resources management. Only the monastery model is an 8.2 megabyte database, accompanied by 63 megabytes of textures. So the Yuso application uses data structures to store the necessary information to define the user movement, or the impact volumes data.

Once the scene graph is loaded from the input file, and the objects declared, these are rendered into the windows. Each time the user uses the tracker (input device to define the user location into the virtual world), the Yuso application compute that information and calculate the following user's position according to the tracker movement. All those location information has to be stored and compute to get the newest positions.

In order to do that the Yuso application uses some functions to set and get the shared memory arena site.

At the same time the information about the objects the user can collide with, or see in each moment, need to be stored in order to be rendered or not, and to check that the user can not go through the walls, floor or ceilings.

A vital part in the computer graphics applications is the programming of the CALLBACKs, since they allow that the user interact with the virtual objects. In the application of the monastery they determine the kind of movement, the type of collision detection, as well as the actions and events that have to do with the keyboard, trackers, etc.

For example, to define a function that opens doors, and that causes the sound of the door hiss: this door is defined like DCS, a dynamic local coordinate system is fixed within this object, and the limits

consists on a location-mechanism that marks the user's position in the architectural complex. There are defined several channels; one of those shows us the perspective view, in which the user navigates. Another channel has associated a window using the orthographic view, that locates the user in the monastery.

2) One of the utilities that can make easier the user's navigation, is the possibility to implement a programmed navigation. For it, several tools can be used, but the most effective could be introducing a default path to be used during the simulation. The user is transported into the virtual world, but he can stop and interact with the objects during the trip. This way, the user can forget to handle with the joystick, tracker, or whatever input device that is being used, and enjoy the virtual trip.

3) Interactive device with informative goals. This device consists of several buttons that fit with some important topics related with San Millán. When anyone of the virtual buttons is pressed, a panel with the requested information pops up.



Figure 3: Interactive device to facilitate information.

4) Creation of a virtual repository: Because of the importance of the library of the monastery of Yuso, and with the

purpose of letting the user interact with these virtual codices, this implementation is one of the most functional applications from the informative point of view.

This possibility would not only consist on the mere animation of the codices, but in creating a virtual repository and providing the user the possibility to handle virtually those codices. What is more, depending on the "professional degree" of the user, this one has access to some documents or has control to perform determined actions. This way all the researchers of the world would have access to that valuable information without moving to Spain, and asking for so many permissions.

5) Another important utility that can have the project of Yuso is not only the recreation of the monastery, but in general of some different historical buildings, with the purpose of showing the evolution that those ones have suffered along the time. This way, the different stadiums would show how the building has changed in a visual and interactive way.

6) Finally, and more focused on the construction world, the model can be used for evaluating the impact that possible restoration works could cause on the building. This way, the effect of a restoration work would be simulated, and some possible works could be evaluated.

CONCLUSIONS

The new technologies have acquired the maturity enough to be used in very different environments as for example history, medicine, education and of course the industry. Exploring and innovating the resources that these technologies offer us, in these and in other kind of fields, is in last term the

purpose of our project.

The importance of the part of programming the codes, or the easy portability of the application, is the possibility to personalize the application as much as it is desired. The project is not limited to the projection of an elaborated geometric model, but the user has the possibility to choose one kind of navigation, and interact with the objects. For example, the user opens doors, he listens at any kind of sounds due to his presence, finds information about the historical importance of the monastery, or flies to the tower.

Depending on the expected results to obtain, different methods should be used. Even among the applications of Virtual Reality which goals are mainly informative, depending on the public those applications are meant for, is convenient to use one type of interfaces or another.

The study on new design of interfaces facilitates to personalize a "made-to-measure" application. This way, focusing the design on the functionality, and working with heterogeneous groups, many of the existent deficiencies in prototypes or previous versions, can improve.

The potent tools that the new technologies offer us, are in a position to let us innovate the world of the cultural diffusion. It is our work to use them.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the provided information given by the "Fundación San Millán de la Cogolla" for this research work.

REFERENCES

1. Steve Aukstakalnis & David Blatner *"The Silicon Mirage: The art and Science of Virtual Reality"* PechPit Press 1992, ISBN0-938151-82-7.

ABOUT THE AUTHORS

María Aróstegui, studied the Licenciatura in Mathematics in the Basque Country and got an scholarship to finish her degree in Milan. She got two internships focused on VR technology to work in, LABEIN(Spain) and VRAC(USA), respectively. Among her research areas highlight Virtual Heritage, and computer graphics and visualization techniques for simulation purposes.

She has recently written some papers about Virtual Heritage for building magazines and the international meeting Culturtec2000.

E-mai: arostegui@labein.es

José Ignacio Barbero, was born in Legazpia (Spain) in 1965. He is Mechanical Engineer from the University of Navarra. He joined LABEIN in 1989 and currently is Manager of the Virtual Prototypes Technological Program. His main research interests are focused on Virtual Prototyping, Virtual Heritage, simulation and visualization techniques for engineering analysis and HPCN (High Performance Computing and Networking).

He has participated in multiple national and European research projects. E-mail: jose@labein.es

Carolina Cruz Neira is a Litton Assistant Professor at the Electrical and Computer Engineering Department and the Associate Director of the Virtual Reality Applications Center at Iowa State University.

In 2000 Dr. Cruz received the Iowa State Foundation Award for Early

Achievement in Research.

Dr. Cruz completed her doctoral studies in Electrical Engineering and Computer Science (EECS) at the Electronic Visualization Laboratory (EVL) at the University of Illinois at Chicago in 1995.

E-mail: carolina@vrac.iastate.edu