A Virtual Reality based System Environment for Intuitive Walk-Throughs and Exploration of Large-Scale Tourist Information

Kornél Szabó[1], Peter Stucki[1]
Patrick Aschwanden[2], Thomas Ohler[2],
Renato Pajarola[2], Peter Widmayer[2]

[1] MultiMedia Laboratory
Department of Computer Science
University of Zurich
Winterthurerstrasse 190, 8057 Zurich
Switzerland
ph: +41 1 257 4346; fax: +41 1 363 0035
email: szabo@ifi.unizh.ch
http: //www.unizh.ch/data/kornel.html

[2] Institute for Theoretical Computer Science
Swiss Federal Institute of Technology
Haldeneggsteig 4, 8092 Zurich
Switzerland
ph: +41 1 632 7402
email: pajarola@inf.ethz.ch
http: //www.inf.ethz.ch/

Abstract

This paper describes the concept and prototype architecture for Virtual Reality (VR) based Information Systems (ViRXIS). ViRXIS may serve as a base architecture for different kind of IS domains, such as a VR based Tourist Information System (ViRTIS) or a VR based Geographic Information System (ViRGIS). Finally, potential application scenarios of ViRTIS will be presented.

Keywords

Tourist Information Systems, Virtual Reality, virtual worlds, real-time interactive 3D simulation, information systems, man-machine-interface, object-oriented database management systems, spatial data access structures.

1. Introduction

The ever increasing computing power and storage capacity of low-cost computer systems enables the implementation of multimedia applications that integrate different media such as text, image, graphics, voice, music, computer animation, or video for the presentation and manipulation of tourist information. At present, the userinterface of such multimedia-based Tourist Information Systems (TIS) is mainly based on the 2.5D-desktop metaphor and the direct manipulation paradigm. One of their strengths compared with classical means of sales support such as color brochures, catalogues or videos lies mainly in their interactivity and dynamics concerning information presentation and manipulation. Progress in information visualization, human-computer interaction, sensing technologies, and the availability of integrated and workstation based simulation platforms allows the extension of existing computer graphics and multimedia technology to a new man-machineinterface concept, called Virtual Reality (VR). Virtual Reality, as a first approach to conceptualize and implement such new man-machine-interaction concepts, will enable the substitution of traditional 2.5D with new 3D user interface paradigms and metaphors. VR also offers potentials for todays information systems, such as Geographic Information Systems (GIS) or Tourist Information Systems (TIS), by overcoming the disadvantages of current audio-visual media spaces, such as limited space for information presentation and limited information exploration capabilities. Especially for tourism related data, characterized with its spatiality, VR may offer an essential extension and enrichment for experiencing information.

As a first walk-through application, the ViRGIS (for Virtual Reality based GIS) application prototype was designed and implemented, that allows an interactive exploration of the Digital Elevation Model (DEM) of Switzerland [1]. Based on a client-server concept, this prototype application integrates two system platforms over a local area network, one used for the visualization and navigation, the other used for the data management of the DEM. During the development of ViRGIS it became clear, that this system prototype may also serve as a base architecture for other VR based information systems, especially for TIS. In the following, the concept and underlying system architecture will be described.

2. System Concept

Interaction with large vorlumes of multimedia type data puts high affordances to the underlying hard- and software resources. Because of main memory limitations, the design and implementation of a real-time simulation environment has to consider special data handling techniques to offer efficient and effective information presentation and manipulation conditions for the user. Most simulation environments use flat files instead of a Database Management System (DBMS) for storing and loading virtual scenes. Unfortunately, such techniques limit the possibilities for launching data queries to objects within the virtual environment. The underlying idea of ViRXIS, a prototype architecture for VR based information systems, was to develop a VR based information system architecture, that offers the potential for complex data queries. For the management of the existing entities in a virtual interactive 3D environment, an object-oriented DBMS (ooDBMS) was chosen. The ooDBMS´s role is to serve the simulation environment with any kind of requested information entities and store changes of entity attributes. The virtual scene is sub-divided in logical 3D rectangly organized patches that form the basic building blocks for a scene in the main memory on the simulation platform. Depending on the viewing frustum, defined by the user´s actual viewpoint, field-of-view, viewing direction and clipping planes, only object data is loaded and kept in main memory that is in the user´s primary focus of attention.

To meet the real-time requirement for interactive 3D simulation, the level-of-detail (LOD) concept was applyed, that is also often used in flight simulator systems. The LOD mechanism stores virtual objects in different geometric resolutions. The LODs of the patches to load into main memory are selected as a function of speed and object´s distance to the viewpoint while exploring the virtual scene. Only one LOD per patch is held in main memory to keep memory load at a minimum and have the opportunity to hold more objects in the main memory.

3. System Architecture

The ViRXIS system architecture consist of two main components, the ViRXIS-InterActor and the ViRXIS-ooDBMS. These two components are linked together by a local area network (LAN) through a low-level networking interface. The objective of this client-server based approach is to distribute the work load over several system platforms and guarantee real-time interaction in a virtual environment. The ViRXIS architecture is shown in Figure 1.

3.1 ViRXIS-InterActor

The interactive 3D simulation component ViRXIS-InterActor runs on a Silicon Graphics Onyx RealityEngine2 with two raster managers using the IRIS Performer visual simulation toolkit. The main task of the ViRXIS-InterActor, the component for interactive 3D visualization and navigation, consists in the processing of the user´s input, loading the required scene patches into main memory and visualizing them on the graphics display. The core element builds the IRIS Performer (Performer) visual simulation toolkit [2] [3]. Performer is a sophisticated toolkit that offers special mechanisms like multi-processing, auto-synchronization of application and rendering processes or shared memory support for interactive real-time 3D simulation. Many of the features of high-end flight simulators such as LOD management are incorporated. Performer builds up a visual non-persistent database at run-time to store and manage object and attribute data of the virtual simulation environment.

Responsible for the dynamic management of the actual scene is the scene manager. It traces the actual scene, inititates data load queries and deallocates no longer needed data. The data loader processes the data load query coming from the scene manager and sends eigther a query message to the ViRXIS-ooDBMS or loads the requested data from the local file system. After the reception of the requested data, the internal visual database of Performer is updated with the new data and automatically visualized. The user interaction manager handles the interactions, mainly mouse inputs and control panel manipulations, between the user and ViRXIS.

3.2 ViRTIS-DBMS

The ViRXIS-ooDBMS runs on a Sun SPARC Server 690 and uses ObejctStore from ObjectDesign Inc. as the underlying ooDBMS. The objective in chosing ObejctStore was to gain the advanteges and features of a commercially available DBMS, such as data distribution, concurrency control, data recovery, support of storing multimedia type data structures, persistency, and a virtual memory mapping architecture to guarantee referential integrity. ObejctStore uses C++ as the host language and offers several tools like a graphical schema designer to ease the design and development process.

The communication with the ViRTIS-InterActor is done over a LAN by BSD sockets. The communication server builds the link between the simulation and ooDBMS environment. It receives data load queries and sends them further to the query manager, where a corresponding data request is initiated to the ooDBMS. To filter the requested objects from the database as fast as possible, a sophisticated data access structure in the form of an R-Tree [4] was implemented on top of the ooDBMS. To minimize the number of disc accesses and network transfers, the persistent neighbour objects are stored physically in clusters. This is well supported by the ooDBMS including corresponding data access mechnisms. This offers the advantage, that only neighbour relationsships need to be considered and spatial selectability can be applyed during information retrieval. The existing application prototype only deals with DEM related data and triangles do not overlap in z-dimension. Therefore only 2D range queries have to be processed.

4. Terrain Database

In our first application prototype, only topological data are managed by the ooDBMS. This data consists of the Digital Elevation Model (DEM) of Switzerland in a resolution of 250 meters between single vertices. The original data had first to be converted into the Silicon Graphics Object (SGO) data format. On the simulation platform dedicated routines allow an easy loading of SGO type objects. The DEM is stored in three LODs: 1.) 250 meters, 500 meters and 1000 meters. To achieve better performance while transfering data from the ooDBMS to the simulation platform, the SGO data format was enhanced to our new SGF format. This enhancement reduced the amount of polygons to be stored and transmitted by 55% percent and offered a better data parsing and read technique for the simulation component.

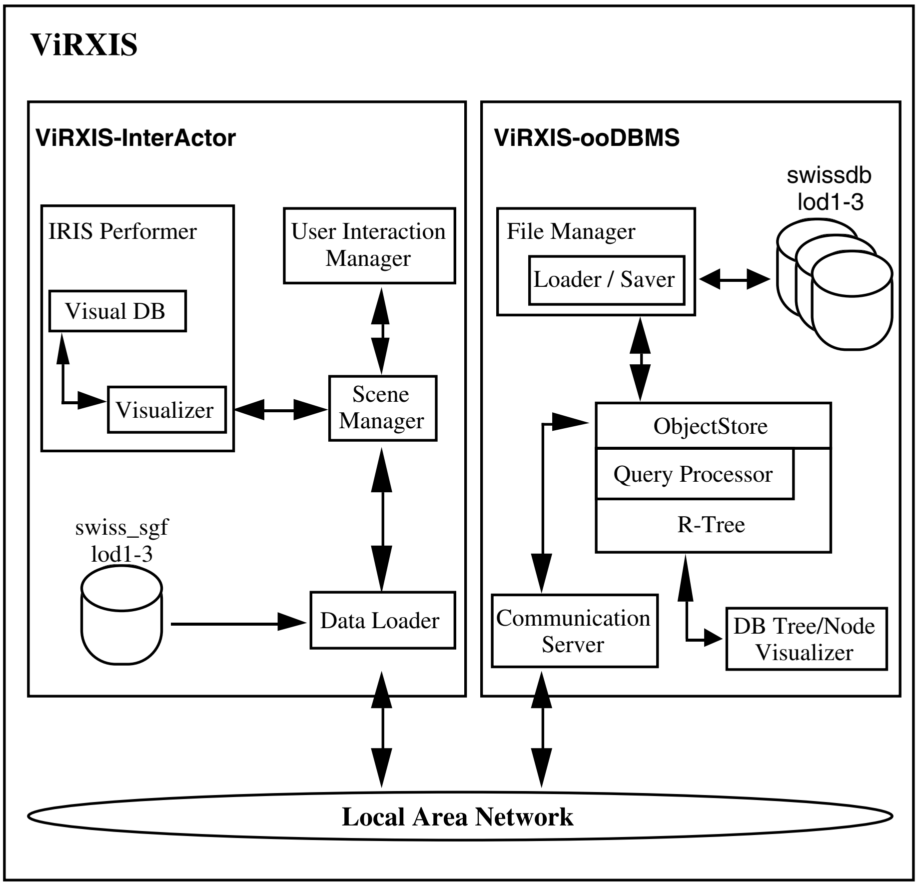

Figure 1: ViRXIS System Architecture

5. Human-Computer Interface

The ViRXIS prototype uses mainly a 2D mouse as input device. The mouse serves on one hand as a navigation tool for exploration of the DEM, on the other hand parameters on the Graphicsl User Interface (GUI) based control panel can be manipulated. Keyboard input is applyed for numerical parameter input, such as setting the viewpoint within the DEM, visualization of rendering statistics, and modification of the rendering mode.

The left mouse button is used for accelleration, the right mouse button for de-accelleration. A stop or stand still is performed by pressing the middle mouse button. The navigation direction can be set by moving the mouse pointer to the corresponding screen partition. Figure 2 shows the computer screen, including visualization window, GUI based control panel and ooDBMS activity window when exploring the DEM of Switzerland interactively.

(Color Screen Dump of ViRGIS showing Scene around Pilatus)

Figure 2: Screendump of the Running ViRGIS System based on the ViRXIS architecture

6. Application Scenarios for VR based Tourist Information Systems

As already mentioned above, ViRXIS may also serve as a software architecture for a VR based Tourist Information System (ViRTIS). By the help of sophisticated 3D data representation and interaction techniques, the user of such a ViRTIS may explore and experience tourism related data in a more intuitive and natural way. This new paradigm for man-machine-interfaces offers potentials that go far beyond conventional 2.5D GUI based information systems.

Enabled by VR research results, telepresence and teleoperation platforms, based on high-speed and high-bandwith digital networks, offer the potentials to revolutionize the way communicative and collaborative work is done. Applying tetepresence and distributed interactive simulation concepts to ViRTIS gives another very interesting and challenging scenario for Computer Supported Cooperative Work (CSCW) like application domains. A user may log-in into a remote host to prototype his next holiday trip. On demand, he can ask for a local tourist guide´s help to show him within the virtual world the interesting sights to visit while planning his stay.

Another interesting application may lie in the field of entertaining gastronomy by the concept of the "The Flying Panorama Restaurant". CAVE [5] or VET [6] like system, that are already available today, allow the setup of an immersive dining room, where guests of a restaurant may have a dinner while for example flying over the mountains of Switzerland.

Finally, ViRTIS could be applied at a Leisure Based Entertainement (LBE) park, at a World Expo or in the classroom of the future to experience exotic places on the Earth or Space in a game like way. CAVE or VET like installations may be linked together over wide area networks to allow users, staying physically at different places, interact together in a shared virtual environment.

7. Conclusions

The ViRGIS project and the evolved ViRXIS architecture gave first insights in the development of real-time interactive walk-through applications. It became clear, that there has to be a focus on efficient data management and access techniques to fullfill the requirement for real-time performance within virtual worlds. Special techniques, such as R-Tree based data access structures, clustering of neighbouring objects, Level-of-Detail, parallel loading of scene data were implemented to maintain real-time performance and interactions within the virtual environment. The loosly coupling of ViRXIS-ooDBMS with ViRXIS-InterActor allows a flexible modification and extension of the database with new entities, that can be used by any other multimedia or VR based applications. The application of an ooDBMS offers the possibility to manage large amounts of data and launching data queries that go far beyond the capability of traditional file system based interactive simulation systems.

Adding VR technology to information systems, such as TIS or GIS, offers potentials to experience large volumes of complex data and allows a faster and more intuitive problem understanding. Benefits gained from such new intuitive and natural interaction paradigms have also implications to the reduction of information retrieval and manipulation costs.

8. Future Works

The development of general purpose 3D tools for advanced information systems, the implementation of 3D user interfaces applying 3D data visualization and manipulation techniques will also gain an increasing focus in TIS and GIS research. The increasing interest for information highway environments require more sophisticated models and techniques for data query, access and retrieval strategies. Therefore, future work will focus on the extension of the spatial model with texture mapping for more realistic scenery, the integration of new information entities, such as tourism related data and architectural objects, and interfacing ViRXIS with 3D peripherial devices like a space-ball, 3D mouse and a dataglove, that are already existing and in use in the MultiMedia Laboratory. This requires also a simultaneous extension of the ViRXIS-ooDBMS classes and query methodes. Additionally, new interaction mechanisms and 3D gadgets will be applyed to enhance the existing interaction techniques. Finally, to benefit from object-oriented design and programming paradigms, we plan the integration of other object-oriented components into the ViRXIS architecture. One of these new components will be the ELECTRA toolkit [7], an object-oriented distributed programming environment based on the CORBA standard [8], that may ease the communication between ViRXIS-InterActor and ViRXIS-ooDBMS and allow the handling of virtual environment objects on a high-level base.

9. References

[1] RIMINI, Das digitale Höhenmodell der Schweiz mit einer Maschenweite von 250 Metern, Bundesamt für Landestopographie, Wabern, 1992.

[2] Silicon Graphics Inc., “IRIS Performer™ Programming Guide”, Silicon Graphics, Mountain View (CA),1994.

[3] J. Rohlf, J. Helman, “IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics”, Proceedings of ACM SIGGRAPH´94, in: Computer Graphics, Annual Conference Series, ACM SIGGRAPH, Orlando (FL), 1994, pp.381-394.

[4] Guttmann A., “R-Trees: A Dynamic Index Structure for Spatial Searching”, Proceedings of the ACM SIGMOD, International Conference on the Management of Data, Boston, 1984, pp.47-57.

[5] CAVE

[6] VET

[7] Maffeis S., “A Flexible System Design to Support Object-Groups and Object-Oriented Distributed Programming”, in: Guerraoui R., Nierstrasz O., Riveill M. (ed.), Proceedings of the ECOOP´93 Workshop on Object-Based Distributed Programming, Lecture Notes in Computer Science 791, Springer, 1994.

[8] Digital Equipment Corp., Hewlett-Packard Corp., HyperDesk Corp., NCR Corp., Object Design Inc., Sun Soft Inc., “The Common Object Request Broker: Architecture and Specification”, Rev. 1.1, OMG Document Number 91.12.1, December 1991.