Responsive Multimedia System for Virtual Storytelling*

Youngho Lee[1], Sejin Oh[1], Youngmin Park[1], Beom-Chan Lee[2], Jeung-Chul Park[3], Yoo Rhee Oh[4], Seokhee Lee[6], Han Oh[7], Jeha Ryu[2], Kwan H. Lee[3], Hong Kook Kim[4], Yong-Gu Lee[5], JongWon Kim[6], Yo-Sung Ho[7], and Woontack Woo[1]

[1] GIST U-VR Lab., Gwangju 500-712, S.Korea
{ylee, sejinoh, ypark, wwoo}@gist.ac.kr
[2] Human-Machine-Computer Interface Lab.
{bclee, ryu}@gist.ac.kr
[3] Intelligent Design & Graphics Lab.
{jucpark, lee}@kyebek.gist.ac.kr
[4] Speech, Audio, and Language Communications Lab.
{yroh, hongkook}@gist.ac.kr
[5] Nanoscale Simulation Lab.
lygu@gist.ac.kr
[6] Network Media Lab.
{shlee, jongwon}@netmedia.gist.ac.kr
[7] Visual Communications Lab.
{ohhan, hoyo}@gist.ac.kr

Abstract. In this paper, we propose Responsive Multimedia System (RMS) for a virtual storytelling. It consists of three key components; Multi-modal Tangible User Interface (MTUI), a Unified Context-aware Application Model for Virtual Environments (vr-UCAM), and Virtual Environment Manager (VEManager). MTUI allows users to interact with virtual environments (VE) through human's senses by exploiting tangible, haptic and vision-based interfaces. vr-UCAM decides reactions of VE according to multi-modal input. VEManager generates dynamic VE by applying the reactions and display it through 3D graphics and 3D sounds, etc. To demonstrate an effectiveness of the proposed system, we implemented a virtual storytelling system which unfolds a legend of Unju Temple. We believe the proposed system plays an important role in implementing various entertainment applications.

1 Introduction

With the rapid advancement of hardware and software, entertainment computing industry has been popularized during the last decade. Nowadays, it is common for users to interact with virtual environment (VE) in various kinds of application areas including simulation, training, education, and entertainment. In this regard, many VR systems have been developed in various types to show its effectiveness.

Many researchers have studied about virtual reality system for virtual storytelling. Most virtual storytelling system integrates multimedia presentation, multimodal interfaces. The representative examples are KidsRoom (Bobick et al., 1996), NICE (Roussos, M et al., 1997) and Larsen and Petersen's (1999) storytelling environment

* This work was supported in part by CTRC, and in part by MIC through RBRC at GIST.

Y.-S. Ho and H.J. Kim (Eds.): PCM 2005, Part I, LNCS 3767, pp. 361 – 372, 2005.

[1][2][3]. They combined the physical and the virtual world into interactive narrative play space. They also make a child's bedroom or a CAVE-like environment changed as an unusual world for fantasy plays by using images, lighting, sound, and vision-based action recognition. They offers interactive story through reactions to the user's actions. However, their interfaces are not natural to control general VE since they exploits specific devices (3D wand) or vision-based tracking system. Moreover, these systems reduce user's interest because they only show same responses without considering users. Moreover, they make users perceive gaps between real and virtual environments by ignoring changes in the real environments.

We present Responsive Multimedia System (RMS) for virtual storytelling with immersive multimedia contents. It consists of three key components; a multi-modal tangible user interface (MTUI), a Unified Context-aware Application Model for Virtual Environments (vr-UCAM), and virtual environment manager (VEManager). MTUI allows users to interact with virtual environments through human's senses by exploiting tangible, haptic and vision-based interfaces. vr-UCAM decides suitable reactions based on multi-modal input [4][5]. Finally, VEManager generates dynamic VE through 3D graphics and 3D sounds, etc.

The proposed RMS has the following advantages. It allows users to naturally interact with virtual environments through multimodal interfaces, such as tangible, haptic and vision-based interfaces. It also shows adaptive reactions according to the user's input through the multimodal interfaces. Moreover, it provides users with realistic virtual environments by exploiting 3D graphics and 3D sounds. Therefore, it maximizes user's interest through interactive and immersive virtual story telling system.

To demonstrate a usefulness of the proposed system, we designed interactive story about 'Unju Temple'. We described several events on specified locations of the temple and then classified them as applicable points for vr-UCAM. Moreover, it shows narrative stories stimulated not only by human physical senses but also by their experiences, knowledge and emotion without limitations of space and time. We also implemented bi-directional interactive 3D web for enjoying contents through the web site. We expect that RMS can evoke personal experience, knowledge, feeling and senses of participants through the virtual storytelling.

This paper is organized as follows. We describe the conceptual design for proposed system in Section 2. In Section 3 and 4, we explain the detailed explanation of RMS and implementation. Finally, the conclusion and future works are presented in Section 5.

2 Conceptual Design of Responsive Multimedia System

In proposed RMS, we divide Environment into four parts as basic elements for storytelling: user, real surrounding, virtual object and virtual surrounding. We assume that these are basic elements for storytelling. All virtual objects and surroundings are connected with an input and output relationship. In this architecture, the start point of interaction can be user or a virtual object. Once users manipulate a real surrounding, user's gesture is acquired from multi-modal interface then it is delivered to virtual objects or virtual surroundings. Then the virtual object shows suitable responses according to the acquired input. Then the virtual object selects animated actions according to the context acquired [16]. Also these animated actions bring an effect to the virtual surrounding. On the other hand, the virtual surroundings influence environmental condition according to the context. As shown in Fig.1, the proposed RMS is composed of a table-type multi-modal tangible user interface (MTUI), a unified context-aware application model for virtual environments (vr-UCAM), and virtual environment manager (VEManager).

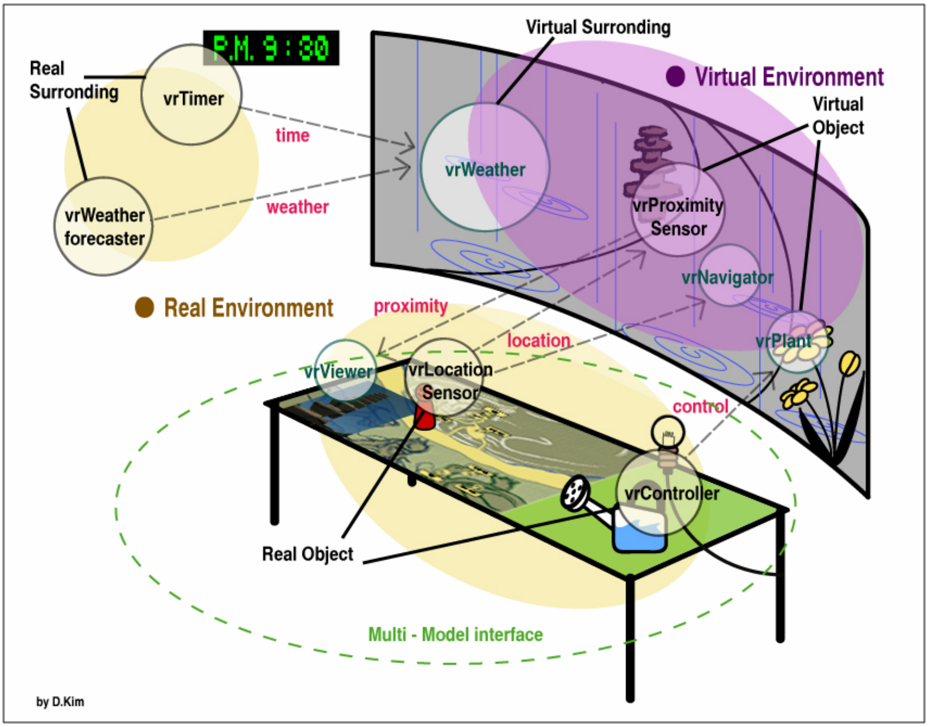


Fig. 1. Conceptual Design of RMS; It supports seamless integration of vrSensor and vrService. Left side is real environment and right side is virtual environment.

To design multi-modal user interface, we consider which modality have to be integrated in RMS. Human perceive the environment in which they live through their senses (e.g. vision, hearing, touch, smell, and taste)[6][7]. Since vision, hearing, and touch are the main factors of human sense, we select these three modalities. On the other hand, human express their own intention with voice, hand/body movement, facial express, and so on. Considering mapping of different human-action modalities to computer-sensing modalities, position/motion and video sensing devices can sense multiple human actions. For instance, facial expression and hand or eye movement can be sensed through the vision sensor. Therefore, we integrate position/motion and video sensing devices in prototype of RMS

We design multi-modal tangible user interface as a form of table since we want to use various types of objects on the table. It has table screen, LCD projector, table glasses, sound devices and one workstation. We include vision-based and haptic interface for the multi-modality in it. So users can manipulate the tangible object and watch information from table screen. Users also sense touch from direct manipulation of tangible object. From the vision-based tracking system, user's gesture and motion are detected by tracking position/motion of tangible object. Clearly, force feedback from haptic device stimulates human's touch sense.

We can improve realism of user's experience by exploiting vr-UCAM2.0 (a Unified Context-aware Application Model for Virtual Environments) [371]. It supports seamless interaction between real and virtual environments. Moreover, it makes virtual object to show adaptive reactions according to user's context. Thus, we can apply vr-UCAM2.0 for integration of the whole system and for unfolding the story. It is also essential to decide pertinent responses suitable for user's explicit and implicit inputs.

VE should provide realistic immersion. It needs to be modeled realistically and objects in VE have to show animated sequences realistically. It is important to show realistic 3D terrain and environment factors such as sky, lightning and weather because users can feel the mood of environment. In addition, animated sequences of virtual objects should be considered. These animations should be used to show appropriate responses to unfold a scenario.

3 Responsive Multimedia System

3.1 Multi-modal Tangible User Interface

MTUI allows exploration and interactive manipulation through haptic and vision-based interfaces. Since we designed MTUI by exploiting tangible interfaces, it supports common daily interaction [8][9]. It assists participants to have deeper recognition of the environments, and enhances the sense of immersion in environments. When a user interacts with virtual cultural assets, it provides a chance for interesting interactions [11]. Our system reflects this point by allowing users to make a Buddhist statue and then locate it in 'Virtual Unju Temple'.

3.1.1 Vision-Based Interface

ARTable is a table-based tangible user interface system which uses a projector and two cameras to display information and to track tangible objects [10]. One of the cameras is placed over the table to capture its surface and augment virtual objects on it. By showing what is happening in the VE within real space, the augmentation helps the user to know how the table and the VE are related. Another camera and the projector are placed under the table. By using half-transparent material, the table could show projection images on the table and the camera could see objects on the table surface. The objects are attached with ARToolKit markers on their bottom so that marker detection is not interfered with users' hand occlusion [13]. And if the objects are needed to be tracked in 3D space, markers are also attached on them.

In RMS, ARTable displays a map with indications of interaction cues through the projector and users manipulate tangible objects to interact with the VE. Although most virtual reality systems are equipped with a 3D joystick for the navigation, it is easy to lose users' current location and direction in VE. Thus we give a location and meaning of it to the user to help user's perception of the VE. Tangible objects are more familiar form than 3D joystick or mechanical devices especially to the novice

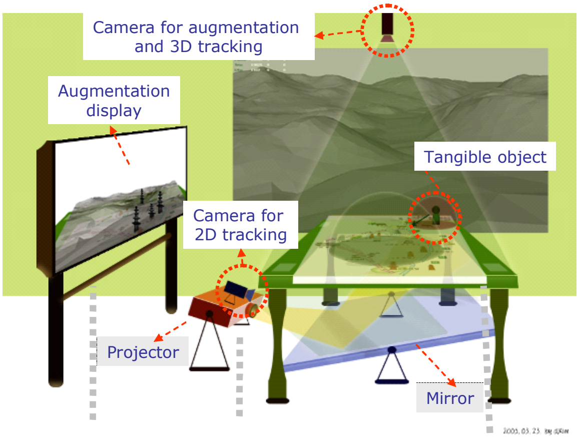


Fig. 2. ARTable and Display

users. Thus they are sensible to manipulate and reflect where the user wants to navigate or what the user want to do.

ARTable is composed of table-type frame, projector, two cameras, table screen, and various types of tangible objects as shown in Fig. 2. Calibration is an off-line process for a transformation between two cameras. After calibration, we can calculate the poses of AR Markers attached on the tangible object. The table and AR display show information to help users' interaction. Tangible objects support various interactions for VEs. And we also display a map of VE on the table screen and we calibrate position of tangible object and the map.

3.1.2 Haptic Interface

A system configuration for haptic interaction is shown in Fig. 3. This system is mainly composed of three parts: database, graphic, and haptic process. The database has rough Buddhist image generated by a modeling tool such as Maya or 3D Max. Each 3D model is represented by geometric vertex, face, color, and texture as obj file format.

The geometric data are used for performing graphic and haptic deformation in each process. Since the graphic and haptic process has their own thread, two processes run at 30Hz and 1 KHz, respectively. In the graphic process, an immersive virtual environment which has a virtual sculpting background is created and a rough Buddhist model is loaded for achieving deformation. Each virtual object is loaded with realistic texture mapping to provide a user with immersive interaction. In order to enhance performance of graphic rendering, each object is optimized by display list and indexed geometry methods which are commonly used for computational optimization of graphic rendering.

In the haptic process, 3 DOF haptic interaction and haptic deformation are performed in 1 KHz rate. 3 DOF haptic interaction is implemented by graphic hardware based haptic rendering algorithm that can deal with any type of object data such as surface-based or volume-based 3D representation as well as primitive models since only the graphic rendering contexts are referred for the collision detection and force

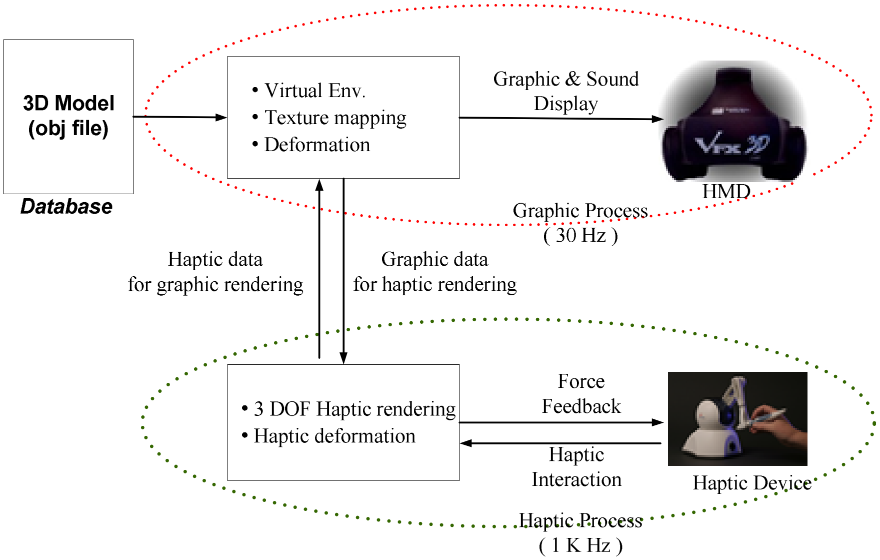


Fig. 3. Haptic Interaction Configuration

generation. In addition, it does not use any pre-computed hierarchy of object data such as bounding boxes or voxmap because the data structure of the on-line the LOMI is small enough to be updated in real-time [12].

After collision detection is performed by graphic hardware based haptic rendering algorithm, then 3 DOF haptic deformation is performed according to 3 DOF force computation between HIP (Haptic Interaction Point) and IHIP (Ideal HIP) [12]. At this time, contact point (IHIP) between haptic interaction point and a surface of 3D model is determined, and then triangles nearby contact point are remeshed. Deformed region are specifically decided by reaction force, and weight factors of moving vertices which determine deformed shape is calculated in the deformed region. Then geometry information of deformation is transferred to graphic process for rendering graphic context. In calculating deformation forces, weight factors of each vertex in deformed region are considered to generate reaction force.

3.1.3 Networked Synchronization for Collaborative Haptic Interaction

We applied server/client architecture. The client sends his or her haptic cursor position to server. The server manages and updates the virtual object states by using the data and sends the update information of virtual objects to all clients. The client architecture supporting the proposed scheme consists of three layers: application, synchronization, and network layer, as shown in Fig. 4. This layered architecture intends to assist the application developer by isolating the synchronization and network issues. Also, it supports easy expansion of a haptic application into a network version with the proposed scheme.

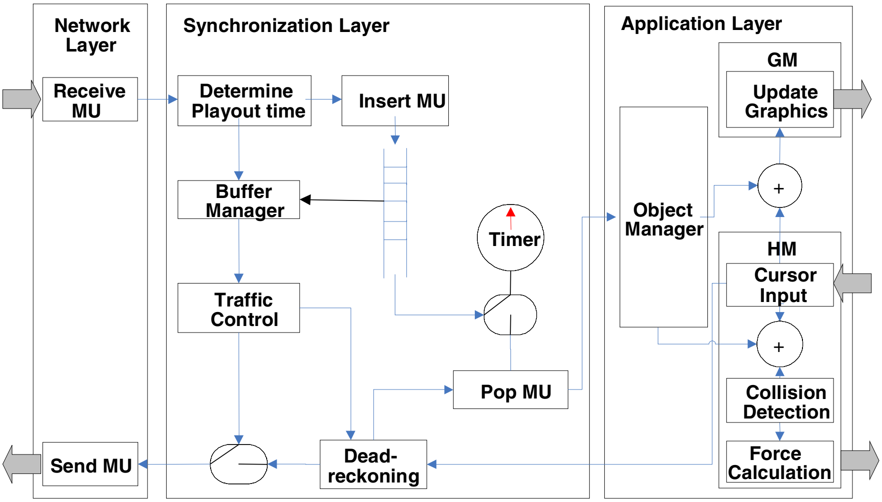


Fig. 4. Architecture for the delay jitter compensation

The synchronization layer relieves the deterioration of haptic interaction quality caused by network delay (jitter) and loss by using the adaptive playout and transmission rate controls. The adaptive playout determination algorithm adaptively controls the playout time according to current network delay distribution to filter out short-term fluctuations and to control appropriate buffering (i.e., waiting) time for smooth haptic interactions. The transmission rate determination algorithm control the transmission rate based on the number of MUs in current buffer. The rate control is achieved by filtering out MUs by exploiting the dead-reckoning scheme.

Finally, the network layer communicates with the other nodes (server or clients) by the MUs. We implement the transmission protocol that refers to the control method of RTP (real-time transport protocol). Additional features such as time stamp and sequence number are expected for the control in consideration of the case to send the state of the VEs. Extra headers to assist the adaptive QoS control of haptic interaction are placed on top of RTP.

3.2 vrSensor and vrService of vr-UCAM

vr-UCAM2.0 (A Unified Context-aware Application Model for Virtual Environments) is a framework for designing reactive virtual environments [12]. It makes virtual objects to show personalized reactions according to user's context. That is, it is aware of user's situation through the user's explicit interactions. Furthermore, it infers user's implicit context (e.g. preference, intention, emotion, etc) from the explicit situation. Finally, it offers adaptive responses according to the context.

The vr-UCAM2.0 consists of vrSensor, vrSCM and vrService. vrSensor generates the context from detected changes. vrSCM supports seamless context sharing between real and virtual environments. vrService decides specific services by analyzing

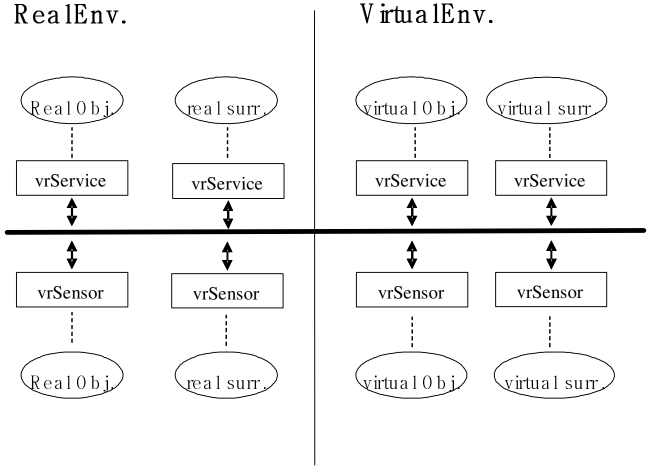


Fig. 5. Circles represent object or surrounding in Environment. Each element is attached to vrSensor or vrService of vr-UCAM. All elements communicate each other over the network.

contexts generated from other vrSensors and vrServices. Thus, it detects user's preliminary context through user's input with tangible objects and haptic interface. Then, it comprehends the user's explicit and implicit contexts, and then it makes virtual environments to show personalized reactions according to extracted contexts. The Fig. 5 exhibits the relationship between four elements which is based on the vr-UCAM2.0.

3.3 Virtual Environment Manager (VEManager)

VEManager is composed of physical engine and environment engine. Physical engine supports interaction between virtual object and virtual surrounding based on collision detection. Environment engine has a role to control virtual surroundings such as weather, temperature, humidity and so on. In this approach, LOD for a massive terrain model and 3D sound generation are important.

3.3.1 Levels of Detail

The RMS contains a number of fine 3-D objects and a massive terrain model. In order to display these models at a specified frame rate, we implement the Level-of-Detail (LOD) technique. The LOD technique is to construct a number of progressively simpler versions of an object and to select one of them for display as a function of range. Three-dimensional objects which locate a great distance from the eyepoint are rendered in less detail, while relatively close objects are represented in great detail. The LOD technique saves rendering time significantly and improves overall display performances. In this system, two types of LOD schemes are used. One is for 3-D objects placed on the terrain, and the other is for the terrain model.

For 3-D objects placed on the terrain, we prepare multiple models of an object with varying levels of detail and associate them with one node called pfLOD provided by OpenGL Performer. A pfLOD node contains an *x, y, z* location of the center of LOD processing which defines the point used in conjunction with the eyepoint for

LOD range-switching calculations. During the culling phase of frame processing, we compute the distance from the eyepoint to the object and select which LOD model to display.

The LOD scheme for the terrain model is slightly different from the above LOD scheme. The terrain model is so large that the entire terrain cannot fit into a single LOD range. In order to address this problem we use Active Surface Definition (ASD). The ASD is a real-time surface meshing and morphing library and an ASD surface contains several hierarchical LOD meshes. When Performer renders an ASD surface, a pfASD node which defines necessary information such as the structure or the evaluation function selects triangles, not model, from many different LODs and combines them into final surface. This approach enables to render the large terrain model that is too large to select single LOD without requiring a lot of system memory.

3.3.2 3D Sound Generation for RMS

The 3D sound generating system consists of two modules: one module is a converter to generate an appropriate sound format and the other one is an interface with the speaker system. Fig. 6 shows a functional procedure of the 3D sound generating system for RMS. The sound generation system first searches a file from a sound file server whenever RMS requests a sound to the sound generating system. After that, the file format of the sound file is modified into the format that can be dealt with the interface with the sound system that supports 5.1-channel speaker system for RMS to give a realistic sound.

For example, consider a sound file stored in the sound file server with the MPEG-1 stereo format. In this case, the sound generating system applies a technique to convert this stereo sound file into a sound file with a 5.1 channel format. Moreover, the sound generating system can play out the sound with the spatial sound effect if RMS gives the direction and distance between the user's position and an object that generates some sound. That is, the sound generating system can relatively control the volumes of five speakers by using the direction information and adjust the loudness for each speaker according to the distance in VE.

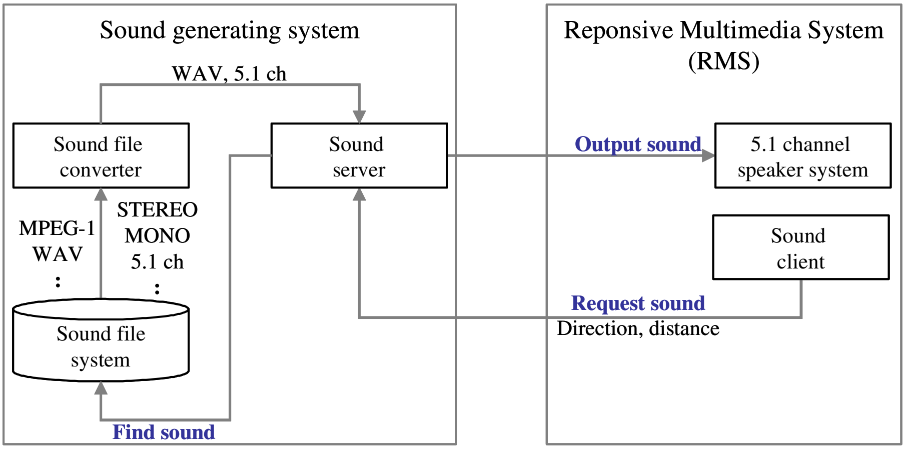


Fig. 6. 3D sound generating system for RMS

3.4 Bi-directional Interactive 3D Web

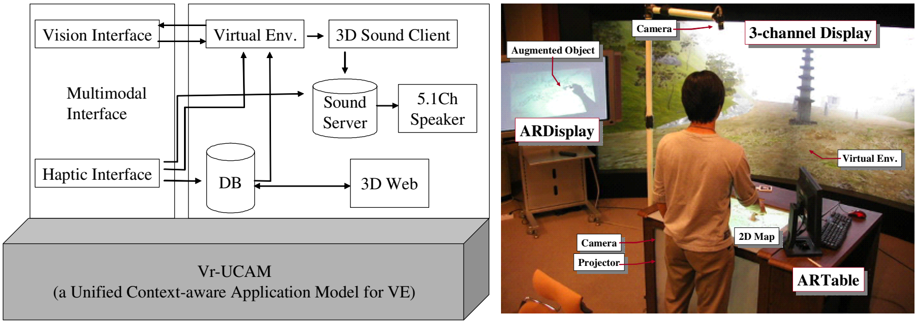
G3 (GIST 3DWeb) started as an effort to enrich the interactiveness in the virual Unju temple. G3 allows one to revisit the statues and monuments seen in the virual Unju temple on his/her personal computers. The direction of interactiveness is bi-directional because one not only can see but can apply paintings to these objects. And surprisingly, this personal artistic works can be seen on the next visit to virual Unju temple. G3 is implemented as an Active-X control and thus can be run as a stand alone MS-Windows application or as an embedded object in a web page (restricted to MS-Explorer). G3 is digitally signed with a certificate from thawte USA (487 East Middlefield Road Mountain View, CA 94043, http://www.thawte.com/) for secure delivery over the Internet.

4 System Integration and Demonstration

We integrated hapic interface, ARTable, 3D Sound system, VRManager, and vr-UCAM into the RMS. Fig. 7 (a) shows structure and information flow of RMS. We built Database and Sound server to save 3D Model and to play 5.1 channel sound. With the haptic interface, we deform 3D model and deliver it into the DB over the network. We display this deformed content in VE and we download it from the 3D web site. We connect ARTable, Haptic interface and VE with sound server. When trigger signal is delivered to the sound server, it selects proper sound file to play through 5.1 channel speaker installed in the room. We designed 3D web server to allow users to interact our contents over the network.

Various equipments were deployed into RMS as shown in Fig. 7 (b). There was three-channel stereoscopic display system. The clustered 3 workstations for cylindrical 3D stereoscopic display, 2 workstations for vision-based and haptic interface, and 1 workstation for sound and database server. In addition to workstation, we installed two cameras, haptic device, and 5.1 channel sound system.

All objects in real and virtual environment were connected with each other using vr-UCAM. We applied vrSenor class to development of vrTimer, vrWeatherForcaster,



(a) (b)

Fig. 7. (a) System Architecture and (b) Demonstration of RMS vrLocationSensor, vrPot and vrShovel. The role of vrTimer was acquisition of time. vrWeatherForecast detected the changes of weather of real environment. vrLocationSensor found the location of object. vrShovel recognized the user's gesture of purring water and the gesture of transferring of object respectively. We applied vrService class to implement vrWeather, vrPicture, vrNavigator and vrPlant. vrWeather changed the weather and time of VE according to the messages generated by vrSensor. For the navigation, vrNavigatior changed the coordinate of VE according to message from vrLocator. When a user approached specific location, vrPicture showed photos of that location. vrPlant grew up a tree in virtual environment. It used message from vrTime, vrPot, vrWeatherForcaster. Virtual plant grew up by taking water and nourishment which were acquired from multi-modal interface and virtual surrounding.

RMS supported users' natural interactions with VE [15]. Users can easily see how to use the interface and feel comfortable with the interface that looks just like an everyday objects in real world. For example, users can move real object that is connected to vrNavigator in VE. Users can carve stone when users beat stone with a hammer shaped device. If users want to pour water on flower, all they need to do is tilt bottle-shape tangible object.

5 Conclusion and Future Work

In this paper, we proposed Responsive Multimedia System for a new approach to virtual storytelling. We made a multimodal tangible user interface for natural interaction by exploiting vision based and haptic interfaces. VEManager provided intelligent responses by showing dynamic scene and animations. Most important feature of our system is that we combined context-aware application model, called vr-UCAM, with VR System. However, there are several possible enhancements to improve the proposed system. We have plans to find an evaluation scheme of our storytelling. We will also apply the concept of artificial life to generate life-like virtual objects. In the future, RMS will be used for providing personalized virtual story according to user' context.

References

1. Bobick, Aaron & Intille, Stephen et al. (1996) "The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment,". Technical report, MIT Media Laboratory.
2. Roussos, M., Johnson, A., Leigh, J., Barnes, C., Vasilakis, C., and Moher, T. (1997). "The NICE project: Narrative, Immersive, Constructionist/Collaborative Environments for Learning in Virtual Reality." In *Proceedings of ED-MEDIA/ED-TELECOM 97*, Calgary, Canada, June 1997, pp. 917-922.
3. Larsen, C.B. & Petersen, B.C. (1999)" Interactive Storytelling in a Multimodal Environment." Aalborg University, Institute of Electronic Systems
4. Seokhee Lee, Youngho Lee, Seiie Jang, Woontack Woo, "vr-UCAM : Unified Context-aware Application Module for Virtual Reality," *ICAT*, pp. 495-498, 2004.
5. Sejin.Oh, Youngho Lee, Woontack Woo, "vr-UCAM2.0: A Unified Context-aware Application Model for Virtual Environments," *ubiCNS*, 2005.

6. D.L. Hall, “An Introduction to Multisensor Data Fusion,” *Proc. of the IEEE*, pp.6-23, vol.85, no.1, Jan. 1997
7. B. V. Dasarathy, "Sensor fusion potential exploitation: Innovative architectures and illustrative approaches", *Proc. IEEE*, vol.85, pp.24- 38, Jan. 1997
8. H. Ishii, B. Ullmer, “Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms,” *in Proc. of ACM CHI*, pp. 234-241, Mar. 1997
9. Sejin Oh, Woontack Woo, “Manipulating multimedia contents with Tangible Media Control System,” *ICEC*, Vol.3166, pp. 57-67, 2004.
10. Y.Park, W.Woo, “Context based AR Interaction Table using Tangible Objects,” *ubiCNS*, 2005
11. K. Salisbury, F. Barbagli, and F. Conti, “Haptic Rendering: Introductory Concepts”, IEEE *Computer Graphics and Applications*, vol. 24, No. 2, pp. 24-32, 2004.
12. Jong-Phil Kim, Beom-Chan Lee, and Jeha Ryu, “Haptic Rendering with Six Virtual Cameras”, *HCI international* 2005. (accepted)
13. ARToolKit (http://www.hitl.washington.edu/research/shared_space/download)
14. Yochen Hiltmann, “Miruk: Die Heiligen Steine Koreas,” *Edition Qumran im Campus Verlag*, 1993
15. Youngho Lee, Sejin Oh, Woontack Woo,”A Responsive Multimedia System (RMS): VR Platform for Immersive Multimedia with Stories,” *ICVS* 2005 (accepted)
16. Seiie Jang, Woontack Woo, “Unified Context Representing User-Centric Context: Who, Where, When, What, How and Why,” *1st International Workshop on Personalized Context Modeling and Management for UbiComp Applications* (accepted)