Client System for Realistic Broadcasting: A First Prototype

Jongeun Cha[1], Seung-Man Kim[2], Sung-Yeol Kim[3], Sehwan Kim[4],
Seung-Uk Yoon[3], Ian Oakley[1] Jeha Ryu[1], Kwan H. Lee[2],
Woontack Woo[4], and Yo-Sung Ho[3]

[1] Human-Machine-Computer Interface Lab.,
Gwangju Institute of Science and Technology,
1 Oryong-dong, Buk-gu, Gwangju 500-712 Republic of Korea
{gaecha, ian, ryu}@gist.ac.kr
http://dyconlab.gist.ac.kr
[2] Intelligent Design & Graphics Lab.,
{sman, lee}@kyebek.gist.ac.kr
http://kyebek9.gist.ac.kr
[3] Visual Communication Lab.,
{sykim75, suyoon, hoyo}@gist.ac.kr
http://vclab.gist.ac.kr
[4] U-VR Lab.,
{skim, wwoo}@gist.ac.kr
http://uvr.gist.ac.kr

Abstract. This paper presents a prototype of a client system for Realistic Broadcasting that can receive and process immersive media. It provides a viewer which supports stereoscopic video display and haptic interaction with the displayed media. The structure of the system is introduced and each component is described. In order to show the feasibility of Realistic Broadcasting, a home shopping channel scenario is applied to the system and its demonstration is performed in an exhibition. We also discuss users' comments and directions for improving of Realistic Broadcasting.

1 Introduction

Realistic Broadcasting is a broadcasting service system using multi-modal immersive media in order to provide users with realism, i. e., photorealistic and 3D display, 3D sound, multi-view interaction and haptic interaction. The concept and overview of Realistic Broadcasting is well introduced in [1].

However, new broadcasting and display technologies succeed or fail depending on how they are received by their audience. Now ubiquitous advances to basic broadcast services such as colour display or stereo sound have been well received. Indeed, it is now hard to imagine watching TV in black and white, or listening to music without stereo sound. On the other hand, many technologies that originally appeared promising have experienced a less illustrious history.

Y.-S. Ho and H.J. Kim (Eds.): PCM 2005, Part II, LNCS 3768, pp. 176–186, 2005.

The wrap-around displays found in IMAX cinemas have been relegated to a niche domain. Many other advances, such as the primitive smell, touch and stereoscopic displays pioneered by the cinema industry in the 1950's (when it was concerned its popularity would fade as televisions became commonplace) have disappeared completely.

What is clear from this is that it is insufficient to simply develop new broadcasting technologies; the opinions of viewers must be considered from the outset. To achieve this we have developed a prototype system which represents the client side of our proposed Realistic Broadcasting system. It implements only the interface elements and not the technological architecture of the final system and enables us to perform user evaluations of our system early on, potentially feeding into a cycle of iterative development. It is this client system that we describe in this paper.

In order to create a practical example of Realistic Broadcasting, we focused on a home shopping scenario. Home shopping channel is a widespread broadcasting format, and one that, due to its focus on showcasing products for consumers to buy, seems likely to benefit from the additional realism of the media that we aim to create. Imagine being able to not only see, but also feel or interact with the products being described in the program. In the system we describe in this paper, a shopping host introduces a number of items and guides viewers through their features. The audience has a stereoscopic view of the scene and is able to touch the products with a haptic interface.

This paper is organized as follows; Section 2 introduces a client system for Realistic Broadcasting and describes detailed part of the system and its example application, a home shopping. Section 3 explains processes of immersive media acquisition and its edition based on the home shopping application. Section 4 describes a prototype implementation of the client system. Section 5 introduces a demonstration of this prototype in ITRC forum and discusses users' comments and directions for improvement of the client system for Realistic Broadcasting.

2 Client System for Realistic Broadcasting

A typical broadcasting client system receives, processes and displays content in the form of video and audio media. In the case of digital television broadcasting, the client system also serves as an interactive user interface, typically allowing Internet browsing or e-commerce applications. For Realistic Broadcasting, we propose a client system that can receive and process immersive media and provide a viewer which supports stereoscopic video display and haptic interaction with the displayed media. In order to achieve this, the streamed media needs to include a 3D representation. Typically, such representations are generated entirely computationally, and as such only apply to artificial, virtual scenes. Their creation is also laborious and time consuming. In our Realistic Broadcasting system, we are mainly concerned with the capture of real scenes, and so those techniques are largely inappropriate. As an alternative, we use depth imaging techniques to represent 3D information derived from raw camera data for static

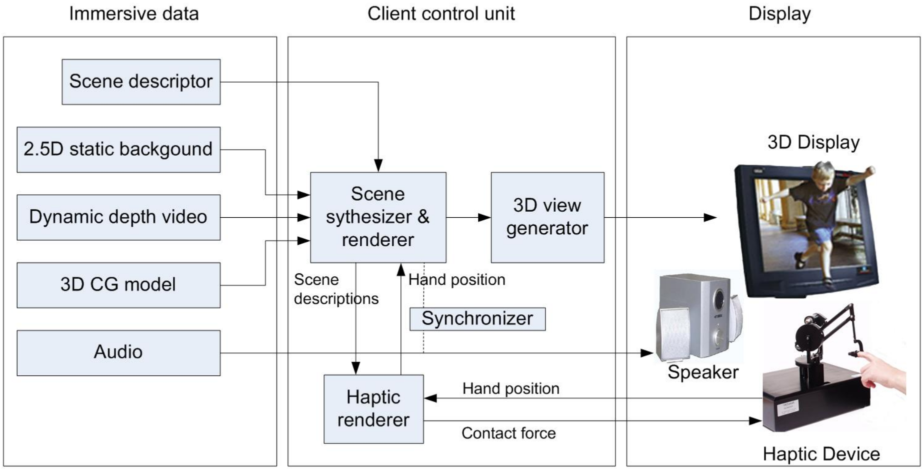


Fig. 1. Client system block diagram

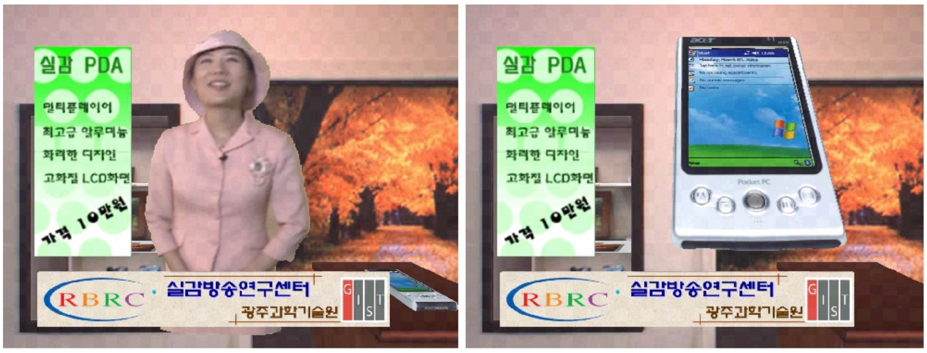


Fig. 2. Home shopping channel scenario

background and depth image captured directly from a depth camera for dynamic part of a scene. However, for objects that are intended for focused, detailed interaction we rely on traditional computer generated models, as they are of a high quality. For haptic rendering of these objects, we augment the graphical models with haptic properties, varying the stiffness, friction and texture as appropriate. Combining this 3D information together with a stereo audio stream yields an immersive media format that we believe will results in viewers attaining increased levels of immersion with the displayed content. These immersive media are then edited and coordinated in 3D space to make meaningful and interesting contents. A scene editor produces a scene descriptor which includes the displayed media identification and the location in the screen. The client system is mainly composed of two parts, a scene renderer and a haptic renderer as shown in Fig. 1. Using the 3D information embedded in the media stream, the scene renderer synthesizes the streamed immersive media by following the scene descriptor instruction and produces a stereoscopic view of the contents. In order to display this view, we simply draw the scene from two virtual camera locations, representing the position of each of the viewer's eyes. We can then use one of a number of 3D display technologies to actually present the image to viewers. In this prototype, we used shutter glasses, as they are an established and reliable technology. The haptic renderer receives the synthesized scene data from the scene renderer and acquires the position of the viewer's hand from a haptic device. It calculates a contact force from these two data, and transmits this back to the haptic device, enabling the viewer to touch the objects and environment shown in the scene. Two different haptic rendering algorithms are used in this prototype, one specifically designed for the depth video [2], the other for the virtual objects [3].

In order to create a practical example of Realistic Broadcasting, we focused on a home shopping scenario. Fig. 2 shows a snapshot of the shopping channel scene. At the beginning of the scenario, a shopping host gives opening comment and starts to introduce a product. While explaining the product appearance and function, the host disappears and the product comes to foreground of the scene. Then, the host guides viewers to touch and manipulate the product with a haptic display. We showcased three products, a PDA, a sofa and a gold mask in the scenario. Viewers are able to touch the depth video as well as the products.

3 Immersive Media Acquisition and Edit

Image acquisition in our system was split into two parts, one encompassing the static and the other the dynamic elements within a scene. In our home shopping scenario the dynamic parts of the scene were essentially limited to the actress playing the show host, while the static parts were the background or setting against which she was situated. Distinguishing between these two elements is commonplace in broadcasting, and is facilitated by "blue screen" systems that enable actors to appear in front of arbitrary backgrounds. The technologies we used to capture depth video for both of these components are discussed below.

3.1 3D Static Background

Image-based 3D reconstruction of the background environment is a crucial factor in creating a visually realistic scene. Furthermore, a photo-realistic background allows an actor to manipulate augmented virtual objects while walking around the generated background by removing/augmenting virtual objects and interacting with them. Fortunately, off-the-shelf multi-view cameras are able to generate a background model of sufficient quality for our prototype.

The process of generating the photo-realistic background model is as follows. First an appropriate physical set is constructed. Lighting is adjusted to match the both the background and dynamic foreground objects. We then use a Digiclops,

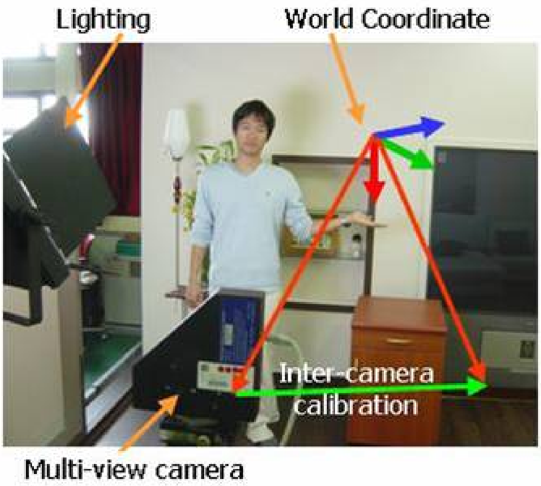


Fig. 3. Environment and System setup for capturing 3D static background

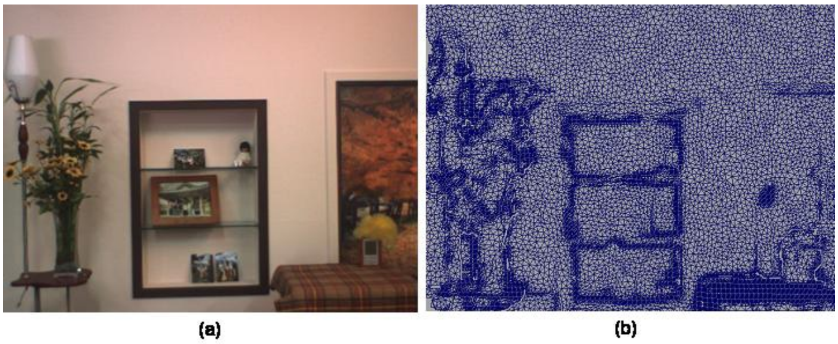


Fig. 4. Captured photorealistic background. (a) Textured view (b) Wireframe view.

a multi-view camera, to acquire a pair of images [4]. The Digiclops calculates 3D coordinates through disparity estimation. After generating a 3D point cloud for each camera position, a projection-based registration method is used to align adjacent 3D point clouds [5], creating a basic depth image. This image is then refined based on the spatio-temporal properties of the 3D point clouds by using adaptive uncertainty regions. Further refinement takes place by searching for correspondences in the projection of the 3D point clouds with a modified KLT feature tracker [6]. Next, the 3D point clouds are fine-registered by minimizing errors. Finally, each 3D point is evaluated with reference to correspondences, and a new color is assigned. In Fig. 3, the process for generating the photo-realistic background is illustrated with a real environment, Fig. 4 (a), one part of the generated background and Fig. 4 (b) the corresponding 3D model.

3.2 Dynamic Depth Video

It is more challenging to acquire depth information from dynamic scenes as they need to be captured and processed in real time. To obtain the dynamic depth stream in our prototype, we used a depth video camera [7] that produces both RGB color and depth signals in real time. Fig. 5 shows the capture system setup in the studio. The depth camera captures a depth value for each screen pixel in its field of view (FOV) using a time-of-flight technique. It is capable of generating an 8-bit (256 level) depth image and makes it possible to record the depth information that pertains to real moving objects at video frame rate.

As we filmed, we set the capture depth range to cover the movement of the actress as illustrated in Fig. 6. For this reason, objects outside of this range were

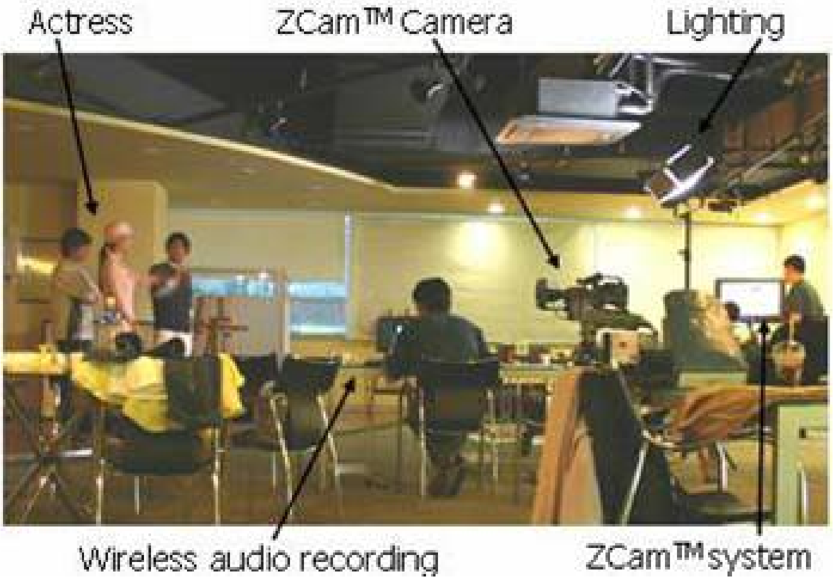


Fig. 5. System setup for capturing the depth video

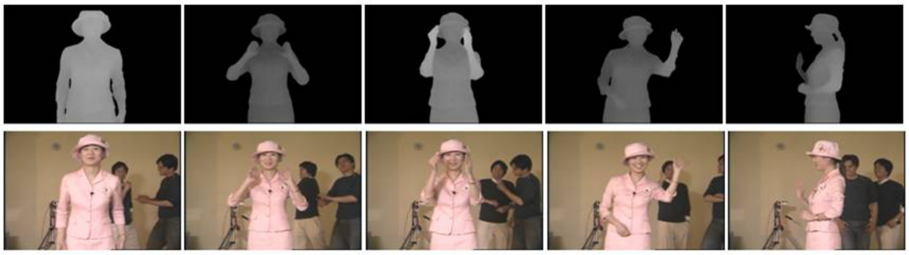


Fig. 6. Selected images of depth video

not detected, and the depth value is reported to be zero for these pixels. This process ensures our depth image has the highest possible resolution around the objects that we are interested in and allows us to easily segment the data using a threshold histogram.

Each depth image represents the 3D position of the actress, but also contains quantization errors and optical noise as the depth value represents the scaled distance from the camera to the actress in only 8 bits. To increase the quality of the depth image we apply a 3D reconstruction technique [8]. The depth image is processed with segmentation, noise filtering, and adaptive sampling techniques based on the depth variation. From the refined depth image we generate a smooth 3D mesh using the Delaunay triangulation method. We then apply a further Gaussian smoothing technique. Finally, the 3D surfaces are graphically rendered, and we generate a final depth image from Z-buffer produced by this process.

3.3 3D CG Model

In our initial production, the three virtual models in Fig. 7 are the subject of the home shopping segment; the actress describes the features of these objects. The models were purchased on the Internet. The inclusion of these three models allows us to experiment with the effects of 3D display and haptic exploration of virtual objects in our Realistic Broadcasting scenario. Typically, virtual model data consists of geometrical information and surface properties such as colour and visual texture. For haptic rendering we augmented the models with physical properties such as stiffness, friction, and roughness. We also attached a button force model to the buttons of the PDA virtual object, which enabled the viewers to push against them, and feel the resultant resistance and clicks.

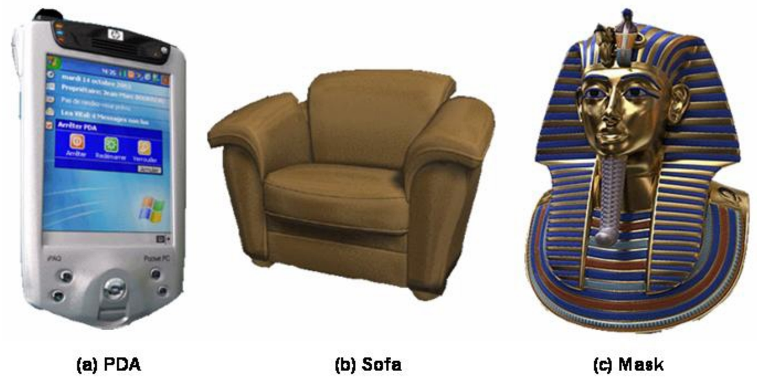
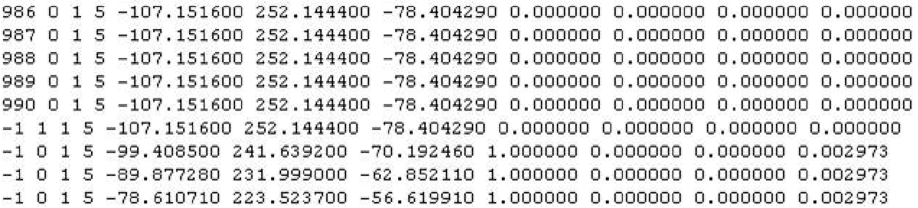

(a) PDA (b) Sofa (c) Mask

Fig. 7. CG models are composed into real scene

3.4 Immersive Media Editing : Scene Descriptor

In order to synthesize the immersive media into a single timeline, we generated a 3-D scene descriptor. The 3-D scene descriptor specifies the translation and rotation information of the computer graphics models for each frame of the depth video. The structure of a 3-D scene descriptor is summarized by

[NumDepthImage][ObjectUpdateFlag][NumObject][ObjectTag][ObjectData]



```
986 0 1 5 -107.151600 252.144400 -78.404290 0.000000 0.000000 0.000000 0.000000
987 0 1 5 -107.151600 252.144400 -78.404290 0.000000 0.000000 0.000000 0.000000
988 0 1 5 -107.151600 252.144400 -78.404290 0.000000 0.000000 0.000000 0.000000
989 0 1 5 -107.151600 252.144400 -78.404290 0.000000 0.000000 0.000000 0.000000
990 0 1 5 -107.151600 252.144400 -78.404290 0.000000 0.000000 0.000000 0.000000
-1 1 1 5 -107.151600 252.144400 -78.404290 0.000000 0.000000 0.000000 0.000000
-1 0 1 5 -99.408500 241.639200 -70.192460 1.000000 0.000000 0.000000 0.002973
-1 0 1 5 -89.877280 231.999000 -62.852110 1.000000 0.000000 0.000000 0.002973
-1 0 1 5 -78.610710 223.523700 -56.619910 1.000000 0.000000 0.000000 0.002973
```

Fig. 8. A part of the scene descriptor used in our system

NumDepthImage represents the depth image to be rendered in the current frame. The value of NumDepthImage will be NULL when there is no depth image for the current frame. ObjectUpdateFlag indicates whether the 3-D scene is to be updated or not. NumObject is the total number of graphic-based models currently in the scene and ObjectTag is the index of graphic-based model to be rendered in the current frame. Finally, ObjectData indicates the translation and rotation information for the virtual models. Figure 8 shows a part of the 3-D scene descriptor.

4 Client System Prototype Implementation

4.1 Hardware

The prototype was implemented on an Intel based PC (Dual 3.2Ghz Pentium IV Xeon, 3GB DDRRAM, nVidia QuadroFX 4400 PCI-Express) under Microsoft Windows XP. We used PHANToM premium 1.5/3 DOF made by SensAble Technologies as our haptic display as shown in Fig. 9 [9]. The PHANToM provides high-performance 3D positioning and force feedback plus a 3 DOF orientation sensing gimbal. By wearing and moving a thimble attached to end of the device, viewers can move a sphere avatar on screen and touch the virtual scene. The stereoscopic display was provided through CrystalEyes shutter glasses.

4.2 Software

Initially, we experimented with SD (standard definition, 720x486 pixels) depth video. In order to display stereoscopic depth videos with OpenGL, we assigned

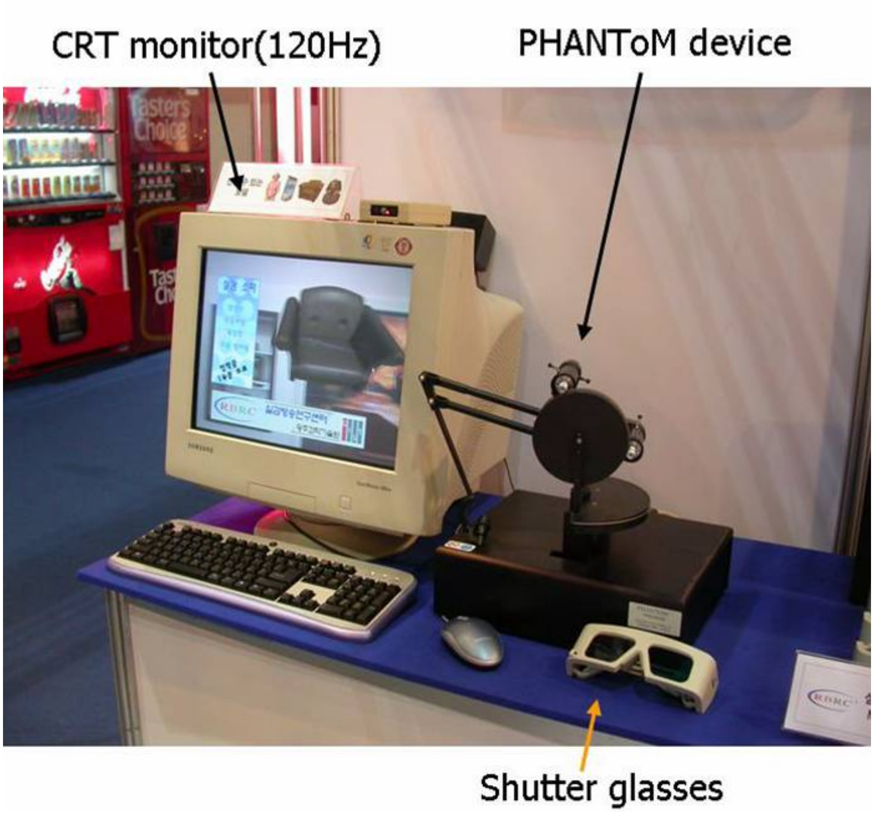


Fig. 9. Client system prototype

column indexes, row indexes and depth values to x, y and z positions. Color information was taken from captured RGB image. Finally, all points were triangulated by adding a diagonal edge. However, after initial tests indicated our system was too slow for real time display, we reduced the graphical resolution to 360x243 pixels. However, we did not reduce the haptic resolution; it remained at SD levels.

The haptic rendering algorithm was implemented using PHANToM Device Drivers Version 4.0 and HDAPI. The HDAPI is a low-level foundational layer for haptics and provides the functions to acquire 3D positions and set the 3D forces at a near real-time 1Khz servo rate.

5 Conclusion

Our prototype was demonstrated at the ITRC Forum held at the COEX exhibition center in Seoul, Korea on the 9th to 11th of June 2005. Figure 10 shows snapshots of the demonstration. Over its three day run, this event attracted many visitors, including IT experts and academics as well school children and the general public. To gain initial impressions of our prototype, we took advantage of this event by creating a questionnaire and attempting to gauge user reactions. While we acknowledge that this sort of evaluation is no replacement for formal empirical study, we found the process useful and informative, and were able to both take encouragement from its results, and to use the comments we received to shape our thinking and influence the next generation of our designs. Most users showed interest in touching both the products and the shopping host. Since the majority of visitors had not experienced force interaction with virtual objects through a haptic device, it was necessary to guide them in their initial explorations. For the most part, young people easily adapted to being able to touch and explore the scene. However, some users had trouble positioning the haptic device on the virtual objects and stated that they found the haptic interaction unnatural.

We attribute this response to a number of factors. The first problem was the display discrepancy between the haptic and graphic workspaces. The graphical display on the screen commands user attention and is the mechanism through which they regulate their position in the virtual scene. However, the workspace of the haptic device they are manipulating in order to perform this control is not coincident to the graphical display. In our system, the haptic device sits to the side of the screen. Although this is a common configuration for haptically enabled VR systems, this discrepancy can be challenging for novice users. The second problem was the use of a sphere avatar to represent the user's fingertip; some users did not immediately understand that it represented their position, which led to some confusion. This problem can be easily solved by substituting the sphere with an avatar that resembles a human hand, or perhaps a simple tool such as a pen. The final problem was simply that it was difficult to perceive the z position, or depth, of the sphere avatar within the scene. Although we provided a stereoscopic view to try to prevent this sort of problem, the differently scaled scene and device positioning made it difficult to precisely locate the depth of the cursor. A potential solution to this problem would be to render shadows, or some other more explicit representation of depth.

Fig. 10. Demonstration in ITRC forum 2005

Some people also commented that the background scene and the shopping host didn't merge seamlessly. This is due to the different lighting conditions in effect during the filming. Although we attempted to use similar light conditions, this process needs refining before we produce more media segments. Finally, a few users commented that they disliked the shutter glasses. The full value of this prototype can only be appreciated when viewed with a 3D display system. However, we acknowledge that the 3D display used must not negatively impact upon the TV experience to which viewers are accustomed. It must allow them the freedom to sit anywhere, with no need for special glasses, and it must be comfortable for prolonged viewing. We are currently considering the adoption of alternative 3D display systems that meet these requirements.

Acknowledgements

This work was supported by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting IT Research Center (RBRC) at Gwangju Institute of Science and Technology (GIST).

References

1. Kim, S. Y., Yoon, S. U., Ho, Y. S.: Realistic Broadcasting Using Multi-modal Immersive Media, Proc. 6th Pacific-Rim Conf. Multimedia (2005)
2. Cha, J., Kim, S. M., Oakley, I., Ryu, J., Lee, K. H.: Haptic Interaction with Depth Video Media, Proc. 6th Pacific-Rim Conf. Multimedia (2005)
3. Zilles, C., Salisbury, K.: A Constraint Based God-Object Method For Haptic Display, Proc. IEE/RSJ Int. Conf. Intelligent Robots and Systems, Human Robot Interaction, and Cooperative Robots, Vol. 3 (1995) 146-151
4. Point Grey Research Inc., http://www.ptgrey.com/ (2002)
5. Kim, S., Woo, W.: Indoor Scene Reconstruction using a Projection-based Registration Technique of Multi-view Depth Images, Proc. 6th Pacific-Rim Conf. Multimedia (2005)
6. KLT: Kanade-Lucas-Tomasi Feature Tracker,
   http://www.ces.clemson.edu/ stb/klt/ (2005)
7. 3DV Systems, http://www.3dvsystems.com/ (2005)
8. Kim, S. M., Cha, J., Ryu, J., Lee, K. H.: Depth Video Enhancement for Haptic Interaction using a Smooth Surface Reconstruction, Special Issue on Artificial Reality and Telexistence, IEICE Transactions, Jan. (2006) (to appear)
9. SensAble Technologies Inc., http://www.sensable.com/ (2005)