3D Video Player System with Haptic Interaction based on Depth Image-Based Representation

Jongeun Cha, Sung-Yeol Kim, Yo-Sung Ho, *Member*, IEEE, and Jeha Ryu, *Member*, IEEE

Abstract — *In this paper, we propose a three-dimensional (3D) video player system that provides haptic interaction with objects in a video scene based on a depth image-based representation. In order to represent dynamic 3D scenes, 3D video media combining general color video and synchronized depth video (depth image sequences) containing per-pixel depth information were exploited. With the proposed system, viewers can touch and explore the shape of objects visible in a scene via a three degree-of-freedom (DOF) haptic interface. To support haptic interaction in the 3D video player, a modified proxy graph algorithm was proposed which calculates the contact force between the 3D video scene and the viewers' finger. In addition, the proposed system architecture was implemented which encompassed 3D video contents generation, depth data enhancement, compression, and haptic rendering. The proposed 3D video player system showed stable, smooth, and immersive interaction in real-time*[1].

Index Terms — 3D video player, haptic interaction, haptic rendering, depth image-based representation.

I. INTRODUCTION

With rapid developments in computing power, digital multimedia and immersive displays, we are currently able to enjoy and be immersed in high quality audio-visual media. Especially in the area of broadcast displays, technological advances have been aimed at providing viewers with a more immersive experience that blurs the traditional boundary between reality and displayed scenes, supporting the impression of 'being there', or *presence*.

For instance, wide-screen displays adopting high-definition (HD) video [1] offer a wide field of view for preventing the viewers from being disturbed by the real environment, and three-dimensional television (3D-TV) supports a natural viewing experience such that viewers are able to perceive objects in true dimensions and natural colors [2]. Additionally, 3D sound systems provide directional audio, further helping to increase the sense of presence in the viewing scene.

However, interaction, an important factor for presence, has not been seriously considered. When viewers have the ability to naturally interact with an environment, or are able to affect and be affected by environmental stimuli, they become more immersed in that environment [3]. Current forms of interactive TV only allow a limited amount of interaction, such as access to additional information transferred with the broadcast channel, connection to the Internet, and participation in polls [4], [5].

Yet these services are still solely information-oriented and merely provide indirect interactions with media contents. Direct interactions between viewers and the video scene would help immerse the viewers in the scene. To overcome this deficiency, one method of supplying direct interaction is that visual interaction which allows a viewer to freely choose viewpoint and viewing direction within the video scene has been adopted in recent years [6], [7].

In this paper, we focus on haptic interaction to enhance presence, such as the ability to directly touch and explore objects in the video scene beyond simply passive watching and listening. Since the human haptic system has the ability both to perceive sensations from the environment and act on the environment, haptic interaction has the potential to create a truly immersive experience [8].

Typically, haptic interaction has been integrated within fully synthesized virtual reality (VR) worlds. To this extent, O'Modhrain and Oakley [9], [10] discussed a potential role that haptic interaction might play in supporting a greater sense of immersion in broadcast content. In addition, they proposed *presentation interaction* which allows the relocation of a character's rendered position in a scene with a small two degree-of-freedom (DOF) force feedback interface. However, since media is based on 2D information, possible haptic interactions in this content are limited.

When the media includes 3D information, more useful interactions may become possible. Cha et al. [11] proposed a number of potential scenarios taking advantage of haptic interactions in broadcast environment in which video media have 3D information. For example, it is possible for viewers to touch and explore an object-of-interest such as a peculiar shaped object or the face of a famous actor.

In this paper, a novel 3D video player system is proposed that provides haptic interaction with objects in a video scene so as to enhance viewer's presence. By using a depth image-based representation, viewers can directly touch and explore the displayed scene as well as stereoscopically enjoy the 3D scene.

[1] This research was supported by the MIC, Korea, under the ITRC support program supervised by IITA (IITA-2005-C1090-0502-0022) through the Realistic Broadcasting Research Center (RBRC) at the Gwangju Institute of Science and Technology (GIST).

J. Cha is with the Department of Mechatronics, Gwangju Institute of Science and Technology (GIST), Gwangju, KOREA (e-mail: gaecha@gist.ac.kr)

S.-Y. Kim is with the Department of Information and Communications, GIST (e-mail: sykim75@gist.ac.kr)

Y.-S. Ho is with the Department of Information and Communications, GIST (e-mail: hoyo@gist.ac.kr)

J. Ryu is with the Department of Mechatronics, GIST (e-mail: ryu@gist.ac.kr).

Contributed Paper
Manuscript received April 15, 2006 0098 3063/06/$20.00 © 2006 IEEE

In general, polygonal 3D meshes are widely used to represent a 3D scene. However, they are not appropriate for 3D video media because of their redundancy in connectivity information, complex level-of-detail, progressive transmission, and compression [12]. Therefore, in order to bridge the gap between the simple 2D texture mapping and full 3D modeling of a video object, a depth image-based representation is proposed [13]. In this representation, 3D video media are the combination of a general color video and synchronized depth video (depth image sequences) containing per-pixel depth information, as shown in Fig. 1.

Fig. 1. General color video and synchronized depth video.

In order to provide a contact force between the displayed scene and the viewer's hands, a haptic rendering algorithm needs to be employed; the process of displaying the haptic attributes of surface and material properties of virtual objects in real time via a haptic interface. Walker and Salisbury [14] proposed a proxy graph algorithm for extremely large static topographic maps of over 100 million triangles, essentially the same dataset as a single depth image. However, this proxy graph algorithm cannot be directly applied to depth video due to the nature of moving pictures. The first problem encountered is that the proxy graph algorithm fails to detect some collisions between the depth video and a haptic interaction point (HIP), representing a user's hand position in the virtual world. Second, it produces piecewise-continuous contact forces due to significantly different video and haptic update rates. Finally, it generates sudden large changes in force at scene transitions, making the haptic interface unstable. Therefore, a modified proxy graph algorithm with depth video is proposed in this paper.

This paper is organized as follows. In Section II, we suggest and describe a 3D video player system architecture containing a haptic subsystem that enables haptic interaction. Section III presents a brief overview of a general proxy-based haptic rendering algorithm and the proxy graph algorithm. A modified proxy graph algorithm is proposed in Section IV. Section V implements the suggested system that encompasses 3D video contents generation, depth data enhancement, compression, and haptic rendering. Section VI then conducts performance tests of the modified proxy graph algorithm, and our conclusions are given in Section VII.

II. System Architecture

While current video player systems including TV sets usually deal with audiovisual media and consist of content generation, compression, transmission, and display subsystems, the proposed system architecture for a haptic interaction enabled 3D video player should contain a haptic subsystem that enables haptic interaction, a haptic renderer and a haptic interface corresponding to a visual renderer and display device. This system can be divided into two parts, a sender and a receiver. A sender captures and sends color and depth videos after signal processing, and a receiver displays the transmitted contents to viewers with a 3D visual display and haptic interface. Fig. 2 describes the entire block diagram of the proposed system.

As can be seen in the figure, the color and depth videos are first obtained from capturing systems, either a depth camera system or a rendering package. Then, depth data needs to be enhanced using a noise filtering technique based on depth image-based modeling (DIBM) for smooth haptic interaction, since data originally include optical noises and shape distortions. After converting color video format into YUV space, the color and depth video are respectively coded by a H.264 coder. Finally, bit streams for color and depth videos are obtained and transmitted to a receiver through a channel.

At the receiver, the transmitted bit streams are decoded so as to recover the color and depth videos, and then each video is synchronized. The graphic renderer provides a 3D stereoscopic view by using the depth video as geometric data, and the color video as texture data, through the 3D display. The haptic renderer receives the depth video and acquires the viewer's hand position from a haptic interface. It calculates the contact force from these two data by using the modified proxy graph algorithm, and transmits the force back to the haptic interface, enabling the viewer to touch and explore the video scene.

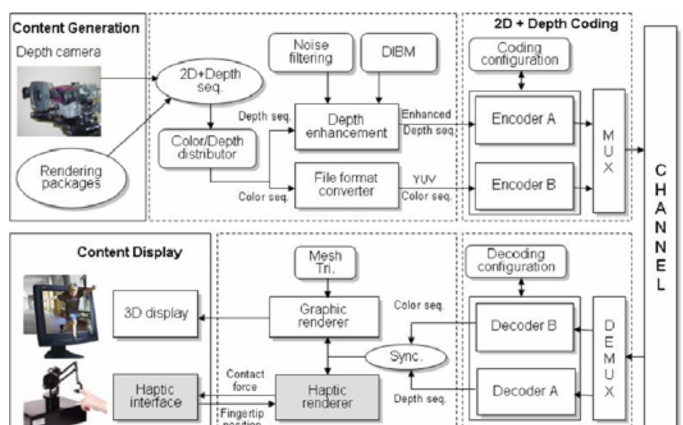


Fig. 2. Overall system architecture for a haptic interaction enabled 3D video player system.

III. Brief Overview of Haptic Rendering Algorithm

In order to explain the proposed modified proxy graph algorithm, this section provides a brief overview of proxy-based algorithms with a summary of a proxy graph algorithm.

Haptic rendering algorithms are essentially composed of the coupled processes of collision detection and force response computation between a scene and a haptic interaction point (HIP) representing a user's hand position (or finger tip). After a collision is detected between objects in a scene and HIP, forces from this interaction are computed to enable a user to touch and explore objects. Proxy-based algorithms based on these concepts are commonplace, as they are simple to implement, robust and reliable. Typically, the user controls the position of a point in the scene with a hand, and a force vector is generated based on the distance between this point and a proxy point constrained to the surface of objects in the scene in the proxy-based algorithms.

This can be easily explained in the example of a user exploring the surface of an object. The collision detection process initially examines the collision between the surface and the line segment between HIP and the previous proxy (dotted lines in Fig. 3). The line segment refers to the path of the users as they move between discrete haptic updates. When no collision occurs, and the user is not in contact with the surface, the proxy and HIP are coincident and no forces are applied, as shown in Fig. 3(a). As the user moves onto, and then penetrates the surface, the proxy remains on the surface while HIP penetrates the surface, and the user feels forces proportional to the distance between these two points. A typical algorithm renders these forces using a linear spring model as shown in Fig. 3(b).

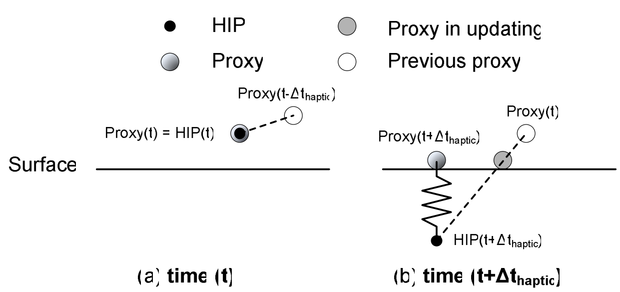


Fig. 3. Typical proxy-based haptic rendering algorithm with plane surface.

A key process in proxy-based algorithms is to update the proxy point that minimizes the distance to HIP. As HIP moves within a scene and the collision is detected between objects, the proxy must shift to an appropriate location on its surface in order to provide a realistic force. Typically, the previous proxy is first moved on the collision point between the surface and the line segment, and then slips on the destination surface. This is trivial in the example of the flat plane shown in Fig. 3(b), but is more complex when considering realistic objects such as polygonal meshes. Detailed processes of this nature are out of the scope of this paper and as such are explained further in the referenced papers [15]-[17].

Haptic rendering algorithms require significantly higher update rates (on the order of 1 kHz) than graphic rendering for both smooth and stable force calculations. The main bottleneck of the haptic rendering process is in collision detection. Whereas a pre-computed "hierarchical bounding boxes" approach has been employed to achieve fast collision detection in the majority of previous works [16]-[20], it is not applicable in this case with respect to depth video that requires the construction of new bounding boxes for each frame. Moreover, a large model such as a depth image considerably slows collision detection (e.g. 697,430 triangles in a 720×486 resolution depth image).

Walker and Salisbury [14] have recently proposed a proxy graph algorithm for an extremely large static topographic map of over 100 million triangles (see Fig. 4(a)) used to describe a landform from satellite images, and essentially the same dataset as a single depth image. This dataset consists of an evenly spaced 2D array of elevations which correspond to gray-scale pixels. A triangle-based surface can be derived from this array by the simple precedent of adding horizontal, vertical and diagonal lines between each element, as shown Fig. 4 (b) and Fig. 4(c).

With their proxy graph algorithm, Walker and Salisbury reduced the overall number of collision detection operations and increased the speed of the proxy update process. In particular, they drastically decreased the computational time of the collision detection process by capitalizing on the special structure of the vertically monotone depth image. In addition, they determined that the hierarchical bounding boxes were not needed.

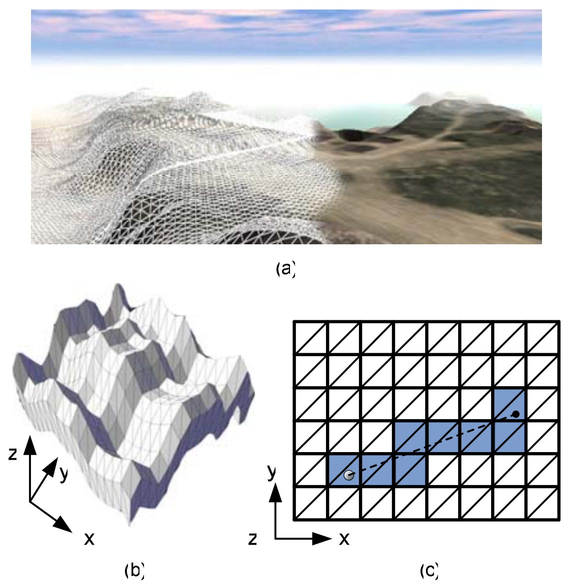


Fig. 4. (a) Height field topographic map. (b) 3D side view. (c) Top view and local search for collision detection.

In the collision detection process, the line segment between HIP and the previous proxy can first be projected onto the 2D representation of the depth image in order to generate a list of candidate triangles which should be checked for collisions (shaded area in Fig. 4(c)). Then, cells within this candidate list possessing elevation values below that of the line segment can be discarded. These optimizations yield an algorithm which executes extremely rapidly when applied to a depth image.

IV. Modified Proxy Graph Algorithm

The proxy graph algorithm in Section III can be successfully applied to one static depth image. However, when applied to a depth video, a stream of depth images, the proxy graph algorithm exhibits a number of problems. In this section these problems are discussed in further detail, and a modified proxy graph algorithm is proposed for depth videos.

A. Collision Detection Correction with Depth Video

A proxy graph algorithm was developed for haptic interaction with a static topographic map. In contrast, a surface triangulated from a depth video is dynamic; the height values of the vertices that correspond to each pixel's gray values changes frame by frame. Consequently, collision detection occasionally fails when the current surface transforms into the next surface, allowing the haptic interface to go through the surface, as shown in Fig. 5(a) and Fig. 5(b).

In Fig. 5(a), HIP and the proxy are coincident in the vicinity of the surface with no collision detection at time (t). When the current HIP is updated and the surface goes up at time ($t+\Delta t_{haptic}$), the surface passes through the previous proxy. As no collision is detected, the current proxy goes coincident with HIP and no force is generated. In Fig. 5 (b), even though a collision is detected and the proxy position is determined on the surface at time (t), the algorithm regards the previous proxy at time ($t+\Delta t_{haptic}$) as inside the surface when the surface goes up. Again, no collision is detected.

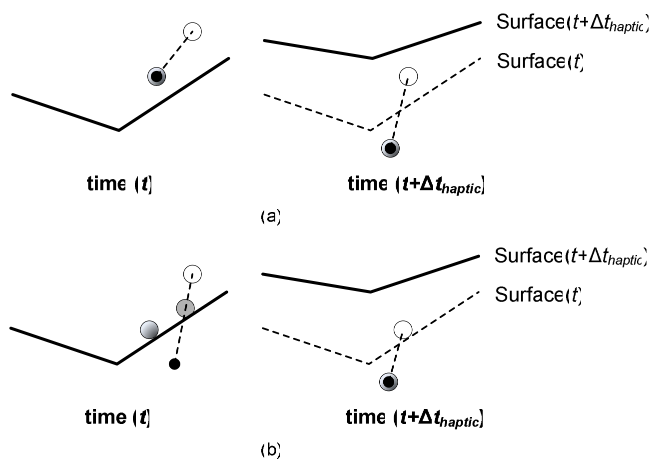


Fig. 5. Collision detection failure of the proxy graph algorithm with depth video (y-direction view of the depth video surface).

These kinds of problems also occur in normal haptic interaction with a moving rigid-body virtual object which passes through the previous proxy. Ruspini [21] corrected this problem by moving the previous proxy position with respect to the movement of the object into the same local coordinate, as shown in Fig. 6.

For each haptic update process, the algorithm stores the proxy position with respect to the local coordinate of the rigid-body object, and when the object moves the proxy position also moves into the same coordinate of the object as before the collision detection process. Therefore, the rigid-body surface cannot pass through the proxy. In Fig. 6, the previous proxy contacting the rigid-body surface at time (t) moves along with the rigid-body and maintains the same contact point in the local coordinate at time ($t+\Delta t_{haptic}$). This time, the collision is detected.

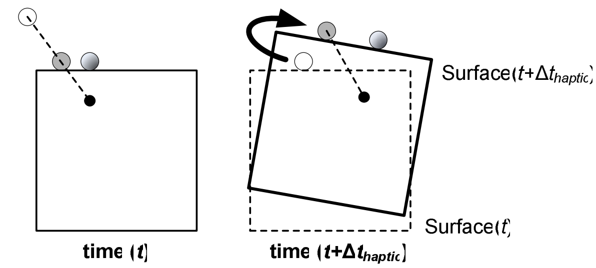


Fig. 6. Collision detection correction in a proxy-based algorithm with a rigid-body object.

However, the same method cannot be applied to depth video because the surface changes its shape frame by frame relative to its local coordinate, whereas the surface of the rigid-body object keeps its shape with respect to its local coordinate. To remedy this situation, when the surface moves upward, the previous proxy also moves by an equal amount, as shown in Fig. 7. The candidate triangle, which goes through the proxy, is easily determined by projecting the previous proxy onto the surface, as in Fig. 4(c). Before the collision detection process, the previous proxy is moved by the amount of the distance by which the candidate triangle moves from time (t) to time ($t+\Delta t_{haptic}$) (the upward z-direction vector in Fig. 7). This ensures that the surface cannot pass through the proxy.

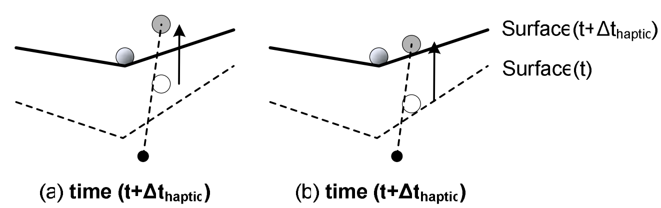


Fig. 7. Correction of the previous proxy position by the amount of surface movement.

B. Local Surface Interpolation

Whereas a high haptic update rate (>1000 Hz) is generally required for a smooth and stable force display, video refresh rates (30 Hz) are much slower than the haptic update rate. Since humans can sense force vibrations well in excess of 300 Hz, this performance gap may cause the contact force to be piecewise continuous; a viewer will perceive the discrete change of the depth video surface between frames.

Ruspini [21] proposed a proxy blending method in haptic interaction with deformable objects, which interpolates an intermediate proxy between the old proxy constrained to the old surface, and the current proxy constrained to the current surface. This method requires additional time to calculate the old proxy position, and the resultant force is delayed by about one graphic update period due to the fact that the old surface defined at previous graphic update time is used.

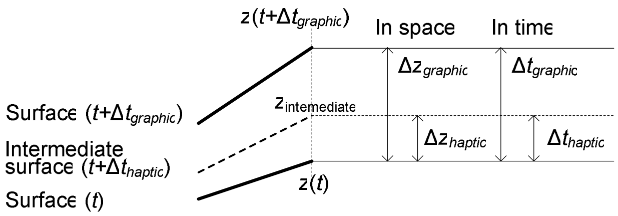


Fig. 8. Intermediate surface interpolated between the current surface and the next buffered surface

As shown in Fig. 8, we do not interpolate the precise proxy position, but rather the local intermediate surface between the current and subsequent depth video surfaces by interpolating the vertices of the surface using Eq. (1). This is possible as a subsequent depth image can be obtained at the current time by buffering. Since this interpolation is performed only on the local surfaces in the collision detection process (shaded area in Fig. 4(c)), and on the vertices the proxy slips in the proxy update process, this calculation time is not critical.

$$\begin{aligned} z_{intermediate} &= z(t) + \Delta z_{haptic} = z(t) + \Delta z_{graphic} \frac{\Delta t_{haptic}}{\Delta t_{graphic}} \\ &= z(t) + \left(z(t+\Delta t_{graphic}) - z(t) \right) \frac{\Delta t_{haptic}}{\Delta t_{graphic}} \end{aligned} \quad (1)$$

where, $z(t)$ and $z(t+\Delta t_{haptic})$ are the depth values of the vertices of the surfaces at time t and $t+\Delta t_{haptic}$, respectively, $\Delta t_{graphic}$ is the graphic update time (about 33 ms), and Δt_{haptic} is the elapsed time from time t to the current haptic update time. Δt_{haptic} can be obtained by counting the clocks from t. Since our system runs on MS Windows, which is not a real-time OS, $\Delta t_{graphic}$ is not an exact constant. However, subjective evaluation by users confirms that this interpolation process is sufficient for proving an apparently continuous force.

C. Sudden Change of the Proxy Position

There can be dramatic changes in depth values between individual frames when a scene changes or when objects in the scene move while playing depth video. In Fig. 9(a), it can be seen that as one scene changes to another, the previous proxy abruptly goes up onto the higher current surface. In Fig. 9(b), the proxy around the object edge is seen to go up onto another higher moving object. If these problems occur, the proxy may very rapidly move with significant force, potentially damaging the device and/or injuring the user. Therefore, this situation should be avoided so as to provide a stable interaction for the viewer.

Thus, in order to solve the above problem, the calculated force is constantly monitored and variations of that value are confirmed. In terms of safety, if a set threshold is exceeded, the proxy is made to collocate with HIP to make the force zero. In other words, in the case where the force changes abruptly, the proxy is allowed to pass through the surface. Consequently, at the moment of scene change the viewer will not experience a dangerous force increase.

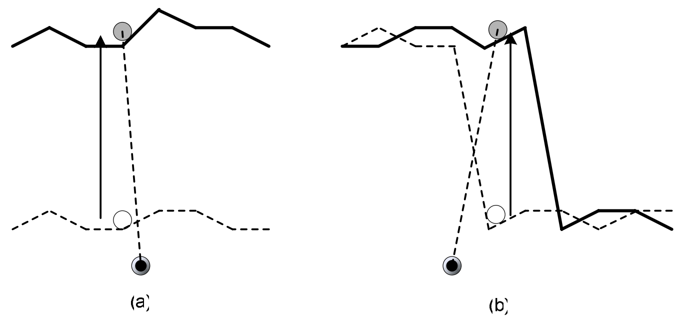


Fig. 9. Sudden change of the proxy position.

V. IMPLEMENTATION

The proposed system architecture was implemented and a sample experiment performed to verify the effectiveness of the proposed modified proxy graph algorithm.

A. 3D Content Generation

The depth video can easily be synthesized by rendering computer graphics animation in off-line rendering packages, where each depth image can be saved by reading the Z-buffer frame by frame, as shown in Fig. 10(a). In another method, a depth camera system can be used for filming a natural dynamic environment. Although a stereo camera system can also produce depth data through stereo matching techniques, a depth camera system provides more accurate depth data for smoother haptic interaction. Moreover, a depth camera allows for seamless changes between long distance shots and close-ups, maintaining the quality of depth data and camera geometry. In recent years, with the technological advancement of active depth sensors such as the ZCam™ depth camera [22], depth video can be directly captured in real-time, as shown in Fig. 10(b).

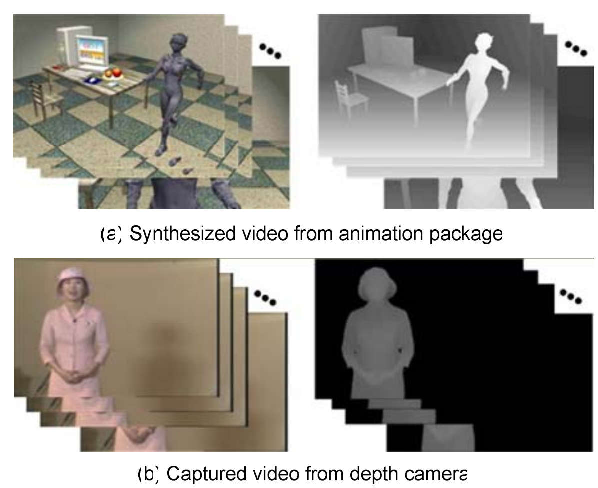


Fig. 10. Color and depth videos.

B. Depth Data Enhancement

Essentially, depth video obtained by a depth camera contains quantization errors and optical noises, mainly due to the reflectivity or color variation of the objects being filmed. When the raw depth map is applied to haptic interactions, these high frequency geometric errors produce distortion such as jagged textures and tremors.

Therefore, in order to provide a smooth haptic interaction, depth data enhancement is required [23]. To accomplish this, a median-filtering technique was first applied to reduce noise. Next, feature points were adaptively sampled using first-gradient analysis, as depth variation greatly affects the quality of a reconstructed surface. The Delaunay triangulation technique was then used to situate the feature points in 2D space. Although a 3D surface may be reconstructed by projecting a 2D triangular mesh into a 3D space based on the filtered depth value, it still contains local noise that produces jagged surfaces. For this reason, Gaussian smoothing was applied to the 3D mesh to enhance the smoothness of the surface. Finally, the 3D surface was rendered with a commercial modeling tool to generate a smooth depth map from the reconstructed 3D surface.

C. Compression of Color and Depth Video Data

In order to compress the color and depth videos, H.264 coders were used. As already reported in the experimental exploration (EE) of 3D audio video (3DAV) in MPEG [24], H.264 outperforms any other codec provided by MPEG standards for coding color and depth videos. Color space was converted into YUV space prior to encoding, as most color data were acquired in RGB space. YUV 4:2:0 was used to encode color data; however, YUV 4:0:0 was used for encoding depth image sequences, as depth image sequences are regarded as luminance images.

D. Graphic Renderer and 3D Display

In order to display these two videos three-dimensionally, the column index, row index and depth value were assigned to x, y and z positions in OpenGL space, respectively. Additionally, each point was set as a color value corresponding to the captured color image, and then all points were triangulated by adding a diagonal edge. To produce a stereoscopic view, the graphic renderer simply drew the scene from two virtual camera locations using OpenGL API, representing the position of each of the viewer's eyes. One of a number of 3D display technologies could then be used to actually present the 3D scene to viewers. In this implementation, a non-glass type 3D LCD display was used, as shown in Fig. 11, as they are an established and reliable technology.

E. Haptic Renderer and Haptic Interface

Haptic interaction was provided using the PHANToM premium 1.5/3 DOF made by SensAble Technologies [25]. PHANToM haptic interfaces provide high-performance 3D positioning and force feedback plus 3 DOF orientation sensing gimbals. By wearing and moving a thimble attached to the end of the interface, viewers could move a sphere avatar in the video scene and touch the displayed video objects (see Fig. 11).

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

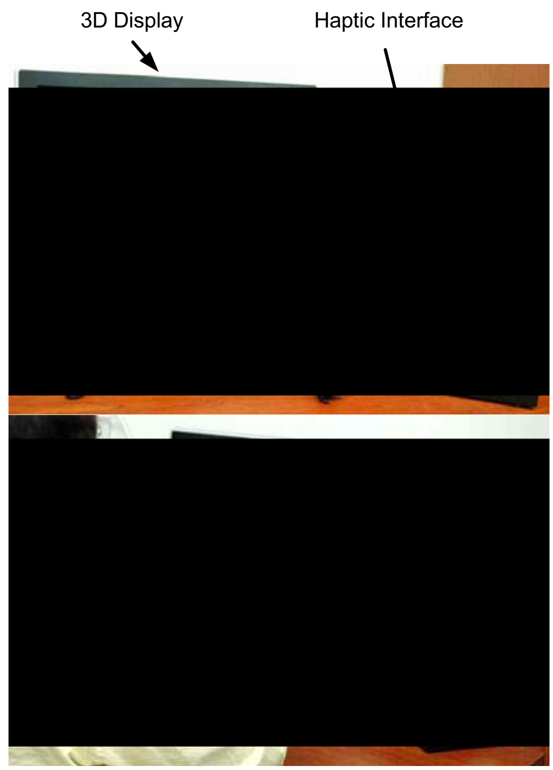


Fig. 11. System implementation for 3D video player enabling haptic interaction.

VI. HAPTIC RENDERING PERFORMANCE EVALUATION

A 3D video player system was implemented on an Intel based PC (Dual 3.0 GHz Pentium IV Xeon, 3 GB DDRRAM, nVidia QuadroFX 4400 PCI-Express) under Microsoft Windows XP. We captured SD (standard definition, 720×486 pixels) color and depth videos by using a depth camera system, as shown Fig. 10(b). Two experiments were conducted to evaluate the performance of the proposed modified proxy graph algorithm for stable and smooth haptic interaction with the depth video.

In general, the haptic update rate should be as fast as possible (typically 1 KHz, i.e. haptic computational time needs to be within 1 millisecond) for stable and transparent haptic interaction. In order to estimate the performance of the modified proxy graph algorithm for stable haptic interaction, the haptic computational time for each haptic update was measured using a high-resolution timer provided in the Windows XP OS. Table 1 shows the results of the performance test for the computational time as presented in [14]. The maximum value was obtained by moving the haptic interface as rapidly as possible across the contact surface. The results show that the proposed modified proxy graph algorithm operates comfortably within a 1 millisecond range, producing a stable force.

TABLE 1
HAPTIC COMPUTATIONAL TIME DURING EACH UPDATE

Dataset Depth Video Resolution	Average Haptic Update Time (milliseconds)
720×486 pixels	0.017-0.070

For smooth haptic interaction, a local surface interpolation of the depth video was proposed in the modified proxy graph algorithm. As mentioned in Section IV, force discontinuity originates from the differences between the video and the haptic update rates. The video update rate was set as 15 Hz and the haptic update rate 1 kHz. The force was then calculated and acquired from the modified proxy graph algorithm.

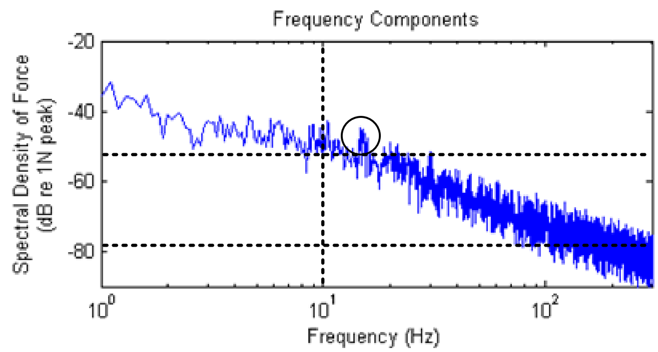
In order to view an improvement in the smoothness, the experimental subject was asked to hold HIP on a fixed point of the surface in a scene with uniform force so as to acquire the force change induced from the scene update. We then selected a contact point on the face of the woman in the scene, since she moved her head more than any other part, thus more frequently changing the surface depth values. The contact force magnitudes with and without the local surface interpolation were acquired at a 1 millisecond rate for 100 seconds. In order to show the degree of smoothness of the force, time-domain data was transferred to the frequency-domain and power spectral densities were obtained. Ten spectral densities corresponding to the ten 10-second segments of the force magnitude data were computed and averaged for noise reduction. Fig. 12(a) and Fig. 12(b) show the power spectral densities of the acquired force magnitudes without and with the local surface interpolation, respectively. The dotted lines are grids for showing differences of the spectral densities. In Fig. 12(a), the circle in the graph indicates the peak spectral densities (-45 dB) around 15 Hz as compared to the surrounding frequencies. We can therefore infer that the 15 Hz video update rate caused a corresponding tremor in the user's hand. In Fig. 12(b), the interpolation method mitigates the peak (-63 dB) around 15 Hz, as well as the entirety of the high frequency spectral densities over 10 Hz. This result verifies that we could interact with a moving object in the video scene and feel smoother movement.

VII. CONCLUSION

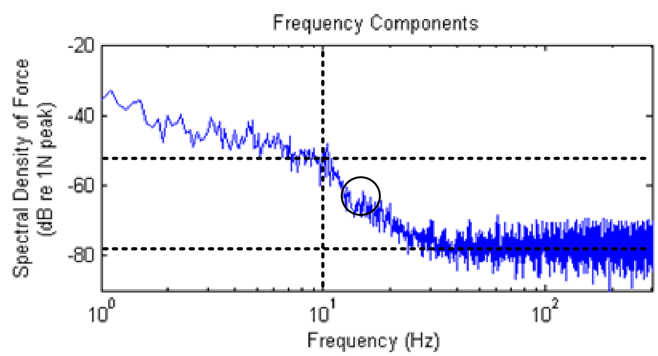
This paper proposed and implemented a 3D video player system that provides a haptic interaction with objects in the video scene in order to enhance a viewer's presence. By using a depth image-based representation, viewers can directly touch and explore a displayed shape in the scene as well as stereoscopically enjoy the 3D scene.

Additionally, in order to provide a contact force between the 3D video scene and the viewers' hand positions, a modified proxy graph algorithm was proposed by adopting the proxy graph algorithm and solving problems associated with the moving depth video.

Our scheme will be an initial stage for providing viewers with a more realistic and interactive experience. At the present time, the proposed 3D video player system and algorithm offer the ability to touch a scene in 3D video media, an application scenario for touching unusual or interesting on-screen objects, such as an actor's face, acquiring shapes or feeling his skin. We believe that the synergy of the consumers' demands, the producer's creativity, and this kind of technical capability will make touch-enabled media rich and abundant.



(a) Before the local surface interpolation



(b) After the local surface interpolation

Fig. 12. Power spectral densities of the acquired forces before and after the local surface interpolation.

ACKNOWLEDGMENT

This research was supported by MIC, Korea, under the ITRC support program supervised by IITA (IITA-2005-C1090-0502-0022) through the Realistic Broadcasting Research Center (RBRC) at the Gwangju Institute of Science and Technology (GIST).

REFERENCES

[1] M. Kawashima, K. Yamamoto, and K. Kawashima, “Display and projection devices for HDTV,” *IEEE Trans. Consumer Electron.*, vol. 34, no. 1, pp. 100-110, Feb. 1988

[2] K. Balasubramanian, “On the realization of constraint-free stereo television,” *IEEE Trans. Consumer Electron.*, vol. 50, no. 3, pp. 895-902, Aug. 2004

[3] B. Witmer and M. Singer, “Measuring presence in virtual environments: A presence questionnaire,” *Presence: Teleoper. Virtual Env.*, vol. 7, no. 3, pp. 225-240, Jun. 1998

[4] M. Bukowska, "Winky Dink half a century later. Interaction with broadcast content: concept development based on an interactive storytelling application for children," Technical Report, Media Interaction Group, Philips Research Laboratories, Eindhoven, Aug. 2001

[5] P. Newton, D. Kelly, J. Tan, J. Shi, and L. Gan, “Recording interactive TV,” *IEEE Trans. Consumer Electron.*, vol. 51, no. 2, pp. 540-544, May 2005

[6] P. Bao and D. Gourlay, “Superview 3D image warping for visibility gap errors,” *IEEE Trans. Consumer Electron.*, vol. 49, no. 1, pp. 177-182, Feb. 2003

[7] A. Smolic and P. Kauff, “Interactive 3-D video representation and coding technologies,” *Proc. IEEE*, vol. 93, no. 1, pp. 98-110, Jan. 2005

[8] M. Reiner, “The role of haptics in immersive telecommunication environments,” *IEEE Tran. Circuits Syst. Video Technol.*, vol. 14, no. 3, pp. 392-401, Mar. 2004

[9] S. O’Modhrain and I. Oakley, “Touch TV: Adding feeling to broadcast media,” *Proc. 1st European Conf. Interactive Television: from Viewers to Actors*, pp. 41-47, Dec. 2003

[10] S. O’Modhrain and I. Oakley, “Adding interactivity: Active touch in broadcast media,” *Proc. 12th Int. Symp. Haptic Interfaces Virtual Env. Teleoper. Syst. (HAPTICS’04)*, pp. 293-294, Mar. 2004

[11] J. Cha, J. Ryu, S. Kim, S. Eom, and B. Ahn, “Haptic interaction in realistic multimedia broadcasting,” *Proc. 5th Pacific-Rim Conf. Multimedia (PCM’04)*, Part III, pp. 482-490, Nov. 2004

[12] A. Ignatenko and A. Konushin, “A framework for depth image-based modeling and rendering,” *Proc. Graphicon-2003*, pp. 169-172, Sep. 2003

[13] P. Kauff, E. Cooke, C. Fehn, and O. Schreer, “Advanced incomplete 3D representation of video objects using trilinear warping for novel view synthesis,” *Proc. Picture Coding Symp.*, pp. 429-432, Apr. 2001

[14] S. Walker and K. Salisbury, “Large haptic topographic maps: MarsView and the proxy graph algorithm,” *Proc. ACM SIGGRAPH’03*, pp. 83-92, Jul. 2003

[15] C. Zilles and K. Salisbury, “A constraint based god-object method for haptic display,” *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS’95)*, vol. 3, pp. 146-151, Aug. 1995

[16] D. Ruspini, K. Kolarov, and O. Khatib, “The haptic display of complex graphical environments,” *Proc. ACM SIGGRAPH’97*, pp. 345-352, Aug. 1997

[17] C. Ho, C. Basdogan, and M. Srinivasan, “Efficient point-based rendering techniques for haptic display of virtual objects,” *Presence: Teleoper. Virtual Env.*, vol. 8, no. 5, pp. 477-491, Oct. 1999

[18] K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles, “Haptic rendering: Programming touch interaction with virtual objects,” *Proc. ACM SIGGRAPH’95*, pp. 123-130, Apr. 1995

[19] S. Gottschalk, M. Lin, and D. Manocha, “OBB-Tree: A hierarchical structure for rapid interference detection,” *Proc. ACM SIGGRAPH’96*, pp. 171-180, Aug. 1996

[20] SensAble Technologies, Inc., “GHOST™: Software developer’s toolkit, Programmer’s Guide,” 1997

[21] D. Ruspini and O. Khatib, “Dynamic models for haptic rendering systems,” *Proc. Adv. Robot Kinematics (ARK’98)*, pp. 523-532, Jun. 1998

[22] 3DV Systems Ltd., http://www.3dvsystems.com

[23] S. Kim, J. Cha, J. Ryu, and K. Lee, “Depth video enhancement for haptic interaction using a smooth surface reconstruction,” *IEICE Trans. Inf. Syst.*, vol. E89-D, no.1, pp. 37-44, Jan. 2006

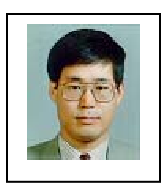
[24] ISO/IEC JTC1/SC29/WG11 m9561, Coding results for EE4 in MPEG 3DAV, 2003

[25] SenSable Technologies, Inc., http://www.sensable.com

Jongeun Cha received his B.S. degree in Mechanical Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1999 and M.S. degree in Mechatronics at the Gwangju Institute of Science and Technology (GIST), Korea, in 2001. He is currently working towards his Ph.D. degree in the Mechatronics Department at GIST, Korea. His research interests include haptic rendering algorithms, haptic interactions in broadcasting and emotional communication with haptic interaction.

Sung-Yeol Kim received his B.S. degree in Information and Telecommunication engineering from Kangwon National University, Korea, in 2001 and M.S. degree in Information and Communication Engineering at the Gwangju Institute of Science and Technology (GIST), Korea, in 2003. He is currently working towards his Ph.D. degree in the Information and Communications Department at GIST, Korea. His research interests include digital signal processing, video coding, 3D mesh representation, 3D mesh compression, 3D television, and realistic broadcasting.

Yo-Sung Ho received both B.S. and M.S. degrees in electronic engineering from Seoul National University, Korea, in 1981 and 1983, respectively, and Ph.D. degree in Electrical and Computer Engineering from the University of California, Santa Barbara, in 1990. He joined the Electronics and Telecommunications Research Institute (ETRI), Korea, in 1983. From 1990 to 1993, he was with Philips Laboratories, Briarcliff Manor, New York, where he was involved in development of the advanced digital high-definition television (AD-HDTV) system. In 1993, he rejoined the technical staff of ETRI and was involved in development of the Korea direct broadcast satellite (DBS) digital television and high-definition television systems. Since 1995, he has been with the Gwangju Institute of Science and Technology (GIST), where he is currently a professor in the Information and Communications Department. His research interests include digital image and video coding, image analysis and image restoration, advanced coding techniques, digital video and audio broadcasting, 3D television, and realistic broadcasting.



Jeha Ryu received his B.S. degree at Seoul National University, Korea, in 1982, M.S. degree at the Korea Advanced Institute of Science and Technology, Korea, in 1984, and Ph.D. degree at the University of Iowa, USA, in 1991. He has been a professor in the Mechatronics Department at the Gwangju Institute of Science and Technology (GIST) since 1994. His research interests include haptics, haptic rendering, haptic interaction control, design and control of haptic devices, as well as haptic-included modeling and broadcasting.