

Immersive Haptic Interaction with Media

N. Dindar, A. M. Tekalp, and C. Basdogan
College of Engineering, Koç University, 34450 Istanbul, Turkey

ABSTRACT

New 3D video representations enable new modalities of interaction, such as haptic interaction, with 2D and 3D video for truly immersive media applications. Haptic interaction with video includes haptic structure and haptic motion for new immersive experiences. It is possible to compute haptic structure signals from 3D scene geometry or depth information. This paper introduces the concept of haptic motion, as well as new methods to compute haptic structure and motion signals for 2D video-plus-depth representation. The resulting haptic signals can be rendered using a haptic cursor attached to a 2D or 3D video display. Experimental results and a demo system are available.

Keywords: Video-plus-depth representation, haptic interaction, haptic structure, haptic motion, video post-processing

1. INTRODUCTION

“Immersive” is a frequently-used term in applications such as virtual reality, computer games, and human-computer interaction. The goal of immersive systems is to artificially create all stimuli that would be experienced in that environment in reality. Recently, with the introduction of new 3D video representations and formats, new modalities of interaction with video for truly immersive media applications are becoming feasible.

Touch is a powerful sense for humans; hence, haptic interaction with video offers users new immersive experiences with applications in entertainment and education media. Haptics in virtual reality and human-computer interaction has been widely studied. However, association of haptic signals with video is quite recent and not fully explored subject¹⁻¹⁰. In passive immersive interaction with media, the viewer can experience whatever haptic information has been pre-authored with the encoded video, such as an haptic movie which may feature haptic information about certain objects. Authoring haptic effects requires computation (possibly off-line) of haptic signals from a 3D video representation during video production. These haptic signals must then be recorded and/or transmitted with the media as a separate track and rendered during video display using a haptic cursor (or another haptic device) attached to a 2D or 3D video display. The haptic interactions include the following haptic effects: i) *Haptic structure*, which refers to touching/feeling the geometry of an object/scene. ii) *Haptic texture*, which refers to rendering surface properties (e.g., roughness) of various objects in the scene, that should be scanned from the scene and encoded as a separate one or more channels haptic texture image. iii) *Haptic motion* which refers to rendering forces related to moving (accelerating) objects in the scene, which can be computed from frame-by-frame estimated motion fields. A mass can be assigned to each object in the scene during authoring, and inertial forces can be calculated for haptic rendering using the computed acceleration.

This paper introduces the concept of haptic motion for the first time, as well as proposing new methods for computation and rendering of forces for haptic interaction with a video. Section 2 provides a brief review of 3D video representations needed to compute haptic structure. Section 3 discusses computation of haptic structure and haptic motion signals in detail, as well as describing associated video post-processing methods. Section 4 shows the demo set-up and some experimental results. Conclusions are presented in Section 5.

2. 3D VIDEO REPRESENTATIONS

Haptic information can be associated with 2D or 3D video. In either case, the structure/geometry of the scene must be captured in the video representation. This scene/object structure can be represented by full 3D surface (e.g., mesh) or

volumetric (e.g., voxels) or implicit models (e.g., B-splines) or by a 2.5D depth-map. While it is difficult to compute and transmit full 3D models for each frame of real-life video scenes, it is possible to acquire depth-maps in real-time by depth cameras or reliably compute the depth-map from multi-view video. Hence, we will focus on the video-plus-depth representation for computation of haptic effects.

The video-plus-depth format includes a depth-map for each video frame as seen in Figure 1. It has been standardized by ISO as MPEG-C Part 3, Auxiliary view syntax of H264/AVC for stereoscopic video representation and transmission¹². The depth images can be compressed as a separate video at low bitrates. MPEG is now considering extending this format to a more general multi-view-plus-depth (MVD) representation for free-view 3D video.

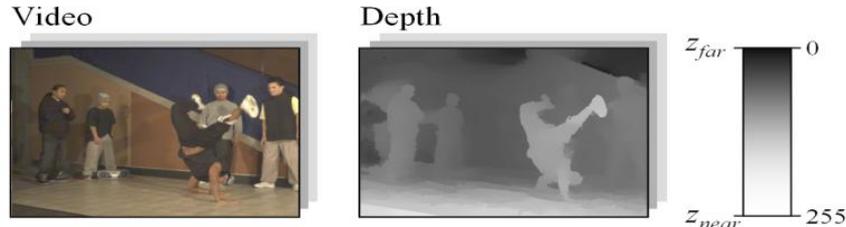


Figure 1: A video frame and its associated depth image¹¹.

3. COMPUTATION AND RENDERING OF HAPTIC SIGNALS

Computation and rendering of haptic signals requires three steps: i) video processing for post-processing of the depth map and estimation and post-processing of the optical flow field, ii) computation of static and dynamic forces at video frame rate, iii) interpolation of haptic signals to haptic rendering rate of 1 KHz. These steps are described below in detail.

3.1 Video Processing

3.1.1 Post-Processing of the Depth-Map

In order to perform object-based enhancement of haptic structure/depth effects, we perform object segmentation using both color and depth images for each frame. We threshold the depth image and fuse the result with the edge map of the color image computed by the Canny edge detection algorithm¹⁵ to find an object segmentation mask.

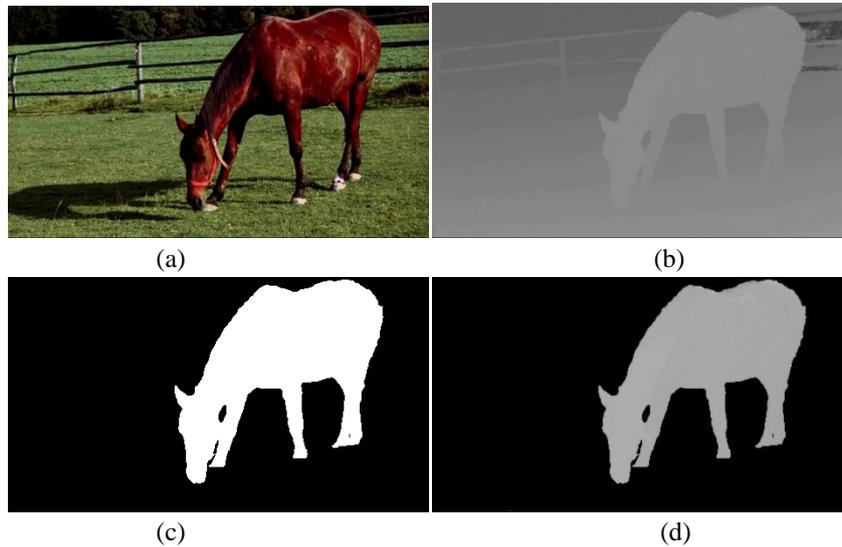


Figure 2: Illustration of the depth image processing: (a) original frame, (b) original depth image, (c) object segmentation mask, (d) post-processed depth image.

Once the segmentation map is computed, the depth-map for each object can be rescaled using the maximum and minimum depth value of the object throughout a temporal scene. We employ an automatic gain control algorithm to stretch the histogram of the depth mask for the object. Finally, the stretched depth-map is filtered by a Gaussian low pass filter in order to achieve a smoother depth-map, which will be used for computation of the static haptic structure forces. Segmentation of the horse object and scaled depth image are shown in Figure 2. The histograms of depth images before and after are depicted in Figure 3.

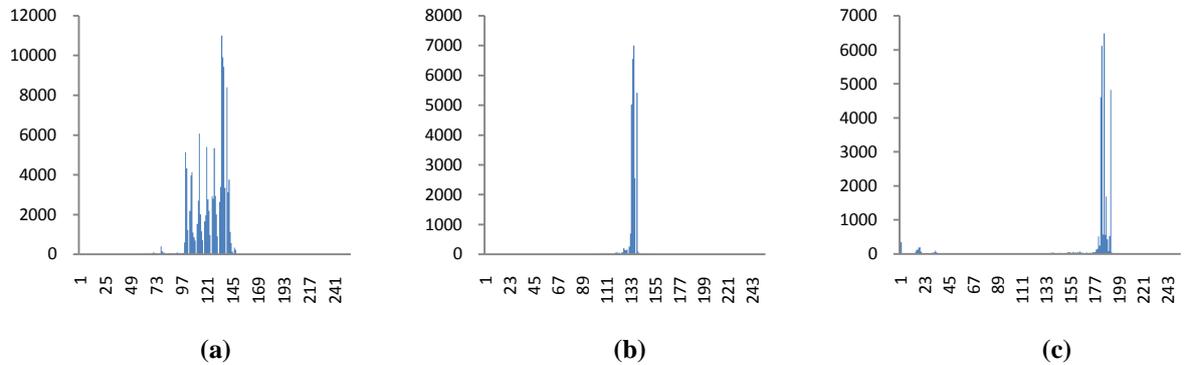


Figure 3. Histogram of depth images: (a) Histogram of the entire depth image, (b) histogram of the depth-map for Horse object before depth scaling, (c) histogram of the depth-map for Horse object after depth scaling.

3.1.2 Estimation and Post-processing of the Velocity Field

We compute the velocity field of the scene/object by the Lukas-Kanade optical flow estimation algorithm. In order to eliminate outlier vectors, and obtain a smooth motion field, 2D median filtering is applied as a post-processing step. The estimated motion fields before and after post-processing are shown in Figure 4 (a) and (b), respectively.

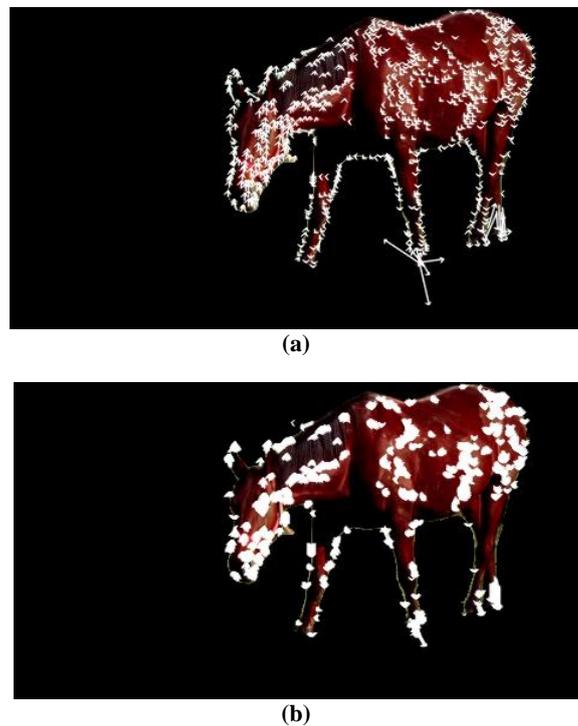


Figure 4. The motion field for the Horse object: (a) estimated by the Lukas-Kanade method, (b) after post-processing filtering.

3.2 Computation of Forces

We model the total force as the sum of static and dynamic forces. While the static force is related to the geometry, material and surface properties of an object, dynamic force relates to the relative motion between the object and haptic interface point (HIP) as shown below

$$\vec{F}_{total} = \vec{F}_{static} + \vec{F}_{dynamic} \quad (1)$$

$$\vec{F}_{static} = \vec{F}_{geometry} + \vec{F}_{material} \quad (2)$$

$$\vec{F}_{dynamic} = m \cdot \vec{a} \quad (3)$$

The dynamic range of the depth image is mapped to haptic workspace linearly such that 0-255 code range corresponds to 0-30 mm in the haptic workspace. We determine the maximum applicable force magnitude using the first N frames to be consistent throughout the video, and scale the computed force using this approach.

3.2.1 Static Force

We render static forces by employing the depth-map rather than a full 3D model, and by segmenting the depth-map values to k -levels. We approximate each depth level by a plane. Force calculation at each pixel can then be performed by the method proposed in [16] using the surface normals of these planes without full or partial 3D object reconstruction. We determine the contact point by checking the HIP position with respect to the depth value of the pixel coordinate of the cursor. If haptic interface point is inside the object, then we calculate the force.

The main idea of force calculation is perturbing the surface normal using the gradient of the depth image pixel values¹⁶. We used this approach as if we are applying two layer textures to the plane. First layer perturbs the surface normal based on the geometry information of the object, and is defined as

$$\vec{M}_{geo} = \vec{N}_s - \vec{\nabla}h_{dep} \quad (4)$$

where \vec{M}_{geo} represents the perturbed surface normal and \vec{N}_s is the surface normal of a plane, by default (0,0,1), h_{dep} is the depth-map value, and $\vec{\nabla}h_{dep}$ is the local gradient at the haptic interaction point computed by the central difference. After the direction of the force is determined, we calculate force given by

$$\vec{F}_{geometry} = \begin{cases} (d - Kh)\vec{N}_s + Kh\vec{M}_{geo} & \text{if } d \geq Kh \\ d\vec{M}_{geo} & \text{if } d \leq Kh \end{cases} \quad (5)$$

In Eqn. 5, d represents the magnitude of the penetration of the object surface, the difference between the depth value and HIP position, and \vec{M}_{geo} is the surface normal after geometrical mapping. K is a scalar that depends on the properties of surface texture and h is the normalized depth value.

3.2.2 Dynamic Force

Moving objects in a video should exert a haptic force due to the relative acceleration between them and the haptic cursor. For example, if the haptic cursor perfectly tracks a moving object then the relative acceleration between them is zero, and no haptic force will be experienced. On the other hand, if the haptic cursor moves towards an accelerating object, force experienced increases. This dynamic force is given by Eqn. (3), where we assign an approximate mass to objects in the scene. Hence, we first compute the relative velocity between the object and the haptic cursor point. Then, the relative acceleration of the object is given by

$$\vec{a} = \frac{\vec{\Delta}v}{\Delta t} = \frac{\left(25 * \frac{\vec{v}_{flow} (pixel/sec)}{1000 (msec/sec)} \right)}{1 msec} - \vec{a}_{cur} \quad (6)$$

where \vec{a}_{cur} is the acceleration of the cursor per millisecond and \vec{v}_{flow} is velocity of the object found by the optical flow algorithm. Here, we scale the \vec{v}_{flow} with 25/1000 since the video runs at 25 fps and the haptic loop runs at 1000 fps.

3.2.3 Interpolation of Haptic Signals

Haptic signals must be rendered with a sampling rate of at least 1 kHz in order to avoid haptic jitter. Hence, haptic information computed at the video frame rate must be interpolated to a rate of 1 kHz using standard filtering methods. This is achieved by calculating the forces at frame i and $i+1$, $\vec{F}_i(x_1, y_1)$ and $\vec{F}_{i+1}(x_1, y_1)$, based on the corresponding depth value of HIP position (x_1, y_1) in frame i and $i+1$. Then a linear interpolation is applied between $\vec{F}_i(x_1, y_1)$ and $\vec{F}_{i+1}(x_1, y_1)$ as shown in Figure 5. While determining the force value in frame $i+1$, we initially assume that the haptic interaction point would stay at the same pixel coordinate (x_1, y_1) , and used that corresponding depth value. If HIP position changes to (x_2, y_2) before image frame $i+1$ is rendered, then interpolation is computed between last rendered force, $\vec{F}_{i_last}(x_1, y_1)$, and $\vec{F}_{i+1}(x_2, y_2)$.

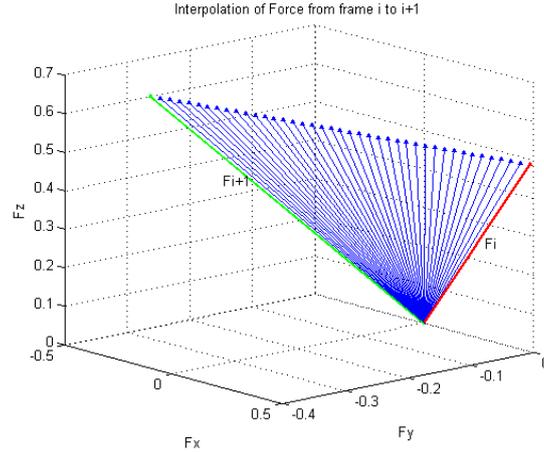


Figure 5. Interpolation between $\vec{F}_i(x_1, y_1)$ and $\vec{F}_{i+1}(x_1, y_1)$

4. DEMO SET-UP AND EXPERIMENTAL RESULTS

Our demo set-up for haptic interaction with video by means of an haptic cursor interface is illustrated in Figure 6. Here, the haptic cursor is connected to a 2D video display. In our work, we use the horse video¹⁴. Originally, the video has 140 frames, however in first 110 frames; there is little motion. Therefore, our proposed approach is tested on the last 30 frames. As the user navigates the scene, she experiences the haptic structure and haptic motion of the horse. The lower part of the screen displays the depth-map of the scene for information purpose so that the user can relate the forces she experience with the scene structure and motion.



Figure 6. The demo set-up with a 2D video display and a haptic cursor. The pixel touched is indicated by the cursor, which helps user navigate and experience the haptic scene and haptic motion.

5. CONCLUSIONS

We describe a new mode of interaction with media, called haptic interaction, where we compute object-based haptic structure and haptic motion signals. The haptic signal is rendered using a haptic cursor attached to a 2D video display. However, it is also possible to render haptic effects with 3D video displays. The demo system shows the feasibility and potential of haptic interaction with entertainment and educational media.

REFERENCES

1. S. O'Modhrain and I. Oakley, "Touch TV: Adding feeling to broadcast media," Proc. of European Conf. on Interactive Television, Brighton, UK, pp.41–47, 2003.
2. M. Reiner, "The role of haptics in immersive telecommunication environments," IEEE Trans. Circuits and Syst. for Video Tech, vol. 14, no. 3, pp. 392-401, 2004.
3. S.-M. Kim, J. Cha, J. Ryu, and K. H. Lee, "Depth video enhancement for haptic interaction using a smooth surface reconstruction," IEICE Trans. Inf. & Syst., Vol. E89–D, no.1, Jan. 2006.
4. D. Gaw, D. Morris, and K. Salisbury, "Haptically annotated movies: Reaching out and touching the silver screen," Proc. of Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems, 287–288, 2006.
5. T. Yamaguchi, A. Akabane, J. Murayama, and M. Sato "Automatic generation of haptic effect into published 2D graphics," Proc. of EuroHaptics, 2006.
6. J. Cha, S.-Y. Kim, Y.-S. Ho, and J. Ryu, "3D video player system with haptic interaction based on depth image-based representation," IEEE Trans. on Consumer Electronics, vol. 52, no. 2, pp. 477-484, May 2006.
7. N. Magnenat-Thalmann and U. Bonanni, "Haptics in virtual reality and multimedia," IEEE Multimedia, vol. 13, no. 3, pp. 6–11, July 2006.
8. K. Moustakas, G. Nikolakis, K. Kostopoulos, D. Tzovaras and M. G. Strintzis: "The force field haptic rendering method: Application in haptic access to visual data for the training of the visually impaired," IEEE MultiMedia, vol. 14, no. 1, pp. 62-72, January 2007.
9. J. Cha, M. Eid, and A. El Saddik, "Touchable 3D video system," ACM Trans. on Multimedia Computing, Comm. and Applications (*ACM TOMCCAP*), in press.
10. J. Cha, I. Oakley, Y.-S. Ho, Y. Kim, and J. Ryu, "A framework for haptic broadcasting," IEEE Multimedia, in press.
11. A. Smolic and P. Kauff, "Interactive 3-D video representation and coding technologies," Proc. of the IEEE vol. 93, no. 1 pp. 98–110, Jan. 2005.
12. <http://multimediacommunication.blogspot.com/2009/02/mpegs-vision-on-3d-video-coding.html>
13. C.-H. Ho, C. Basdogan, and M. A. Srinivasan, "Efficient point-based rendering techniques for haptic display of virtual objects," Presence: Teleoperators and Virtual Environments, vol. 8, no. 5 pp. 477–491, 1999.
14. Mobile3DTV: Video plus depth database. Available online at: www.mobile3dtv.eu/video-plus-depth/
15. J. Canny, "A computational approach to edge detection", IEEE Trans. Pattern Analysis and Mach. Intelligence, vol. 8:679-714, November 1986.

About the Authors: A. Murat Tekalp is a Professor and Fellow of IEEE. He is the Editor-in-Chief of Signal Processing: Image Communication since 2000. Cagatay Basdogan is an Associate Professor of Mechanical Engineering and an Associate Editor of IEEE Trans. on Haptics. Nuray Dindar is a graduate student in Electrical and Computer Engineering.