DYNAMIC FORCE RENDERING DUE TO THE RELATIVE MOTION IN HAPTIC VIDEO

by

Nuray Dindar

A Thesis Submitted to the

Graduate School of Engineering

In Partial Fulfillment of the Requirements for

the Degree of

Master of Science

in

Electrical & Computer Engineering

Koç University

September, 2011

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Nuray Dindar

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

Prof. A. Murat Tekalp

Assoc. Prof. Çağatay Başdoğan

Assoc. Prof. Engin Erzin

Assoc. Prof. Yücel Yemez

Asst. Prof. T. Metin Sezgin

Date:

ABSTRACT

This thesis presents a new approach for haptic interaction with video and associated pixel-based depth data, including rendering of dynamic force due to relative motion between an object in a video and the haptic interface point (HIP) of the user. While the concept of haptic video, that is, haptic rendering of forces due to geometry and texture of objects in a video, has already been proposed, haptic rendering of force due to relative motion between a video object and the HIP has not been studied. We propose that the force experienced by a user should vary according to the movement of a video object relative to the HIP, even though the content of the video shall not be altered by this interaction. To this effect, the acceleration of a video object is estimated using video motion estimation techniques, while the acceleration of the HIP is estimated from the HIP position data provided by the haptic device. We assign mass values to the object and the HIP such that the maximum force rendered will be in the range that can be displayed by the haptic device. Then, the dynamic force due to the relative motion is computed by using Newton's second law and displayed to the user through the haptic device in addition to the static forces due to the geometry and texture of the object. Experimental results are provided to demonstrate haptic rendering of forces calculated with and without including dynamic forces.

ÖZETÇE

Bu tez video ve onunla piksel bazında ilişkilendirilmiş derinlik bilgisini, video içerisindeki cisim ve bu cisimle etkileşmek için kullanılan dokunsal arayüz noktası (HIP) arasındaki bağıl devinim sonucu oluşan dinamik kuvvetleri göz önünde bulundurarak haptic etkileşim alanına yeni bir yaklaşım sunmaktadır.Video içeriğine haptic (dokunsal) bilgi tümleşimi, cisimin geometrisi ve dokusuna bağlı haptic (dokunsal) giydirilmesi konusunda mevcut çalışmalar olmasına rağmen, bildigimiz dahilinde bağıl devinim kaynaklı dinamik kuvvetlerin haptic giydirilmesiyle ilgili bir çalışma bulunmamaktadır. Bizim yaklaşımımızda, video içeriği değitirilmesede, kullanıcı tarafından hissedilen kuvvet HIP'in görüntüdeki cisime göre yaptığı harekete bağlı olarak değişir. Bu amaçla HIP ivmesi sensörlerden elde edilen pozisyon bilgisinin numerik türevlemesiyle hesaplanırken, videodaki derinlik bilgisi ve devinim kestirim teknikleri kullanılarak da cisimin ivmesi kestirilmektedir. Daha sonra HIP ve görüntüdeki cisime kütle ataması yapılır. Bu atama yapılırken hesaplanacak kuvvetin haptic cihaz tarafından izin verilen azami kuvveti geçmemesi göz önünde bulundurulur. Son olarak bağıl devinim kaynaklı dinamik kuvvet Newton'un ikinci yasası kullanılarak hesaplanır ve haptik cihaz aracılığıyla kullanıcılara cisimin geometrik özelliklerinden kaynaklanan statik kuvvete ek olarak geri bildirilir. Dinamik kuvvetlerin eklendiği ve eklenmeden önceki haptic giydirilmesi sonucu hesaplanan kuvvetlerin deneysel sonuçları sağlanmıştır.

ACKNOWLEDGEMENTS

I would like to express my appreciation to my advisors Prof. A. Murat Tekalp and Assoc. Prof. Çağatay Başdoğan for their valuable time, abundant help, prolific suggestions and endless patience without which this thesis would have remained a dream. One simply could not wish for a better or friendlier supervisors.

I would also like to thank my committee members, Assoc. Prof. Engin Erzin, Assoc. Prof. Yücel Yemez and Asst. Prof. T. Metin Sezgin for their valuable suggestions and comments.

I also would like to acknowledge my colleges Arda Aytekin, Tolga Bağcı, Ergun Biçici, Burak Görkemli, Cihat Göktuğ Gürler, Gökhan Himmetoğlu, Güneş İngin, and Müge Karaman, who offered invaluable advice and friendship throughout the tenure of my Master's studies.

Special thanks to Ferda Ofli for making the Koc University years memorable.

I owe my deepest gratitude to my family; my parents, for supporting and helping me in every aspects of my life, my brother for keep reminding me the reality of life , and my sister for pushing me when I need it.

Finally, Salih Özgür Öğüz deserves the most credit for introducing me with haptics and being there for me no matter what happens.

TABLE OF CONTENTS

TABLE OF FIGURES

NOMENCLATURE

2-D	2-Dimensional
3-D	3-Dimensional
BIFS	Binary Format for Scene
DOF	Degree of Freedom
HIP	Haptic Interface Point
LDV	Layered Depth Video
MDV	Multi-view Video plus Depth
MPEG	Moving Picture Experts Group

Chapter 1: Introduction

The next-step in digital media technologies will be truly immersive 2D or 3D high definition media with the help of additional modalities. One of these new modalities will be touch, which is a powerful sense for humans. Haptic video refers to adding a sense of touch to video by pre-authoring shape and texture of certain objects, which offers viewers the possibility of passive immersive experiences in entertainment and education applications. With haptic video, viewers will be able to touch these video objects to experience their 3D shape, texture and motion..

Haptic interaction has been originally proposed for virtual environments **[**1**,**2**,**3**,**4**,**5**]**. The potential of adding haptic force feedback into 2D broadcast video has first been discussed by O'Mohdrain and Oakley **[**6**,**7**].** They have defined the term *presentation interaction*; which involves changing the location of the main character in a cartoon movie through the use of a two degree-of-freedom haptic force feedback interface without altering the structure of narrative. When the user moves the haptic cursor, the character's rendered location in the scene is changed and the force vector derived from the character's displacement from the center of the screen is displayed to the user through the device. In some applications, vibrotactile feedback is also displayed in addition to the force feedback.

Recently, haptic experience with video has advanced to include rendering *structure* and *texture*. *Haptic structure* refers to touching/feeling the geometry of a video object/scene. The 3D structure can be rendered using 3D full polygon or mesh models or pixel-based depth data such as those provided by recently proposed MVD or LDV 3D video representations **[**8**,**9**].** An image together with depth associated with each pixel is sometimes referred to as a 2.5D representation. Since recent standards for 3D video rely on such representations, we believe video together with associated depth images will be widely available in the future and can be used for haptic rendering. *Haptic texture* refers to rendering surface properties (e.g., roughness) of objects in the video, which can be recorded and encoded as separate video channels. Cha et al. have been the first to propose computing the haptic information from depth images rather than full 3D object/scene models [10,11]. They have modified the proxy graph algorithm [12] for depth image based haptic rendering by constructing a triangle based surface using the depth data and taking the projection of the line segment constructed between the current and previous HIP positions onto the depth image for determining the list of candidate triangles. Then, collision detection is performed on this list of candidate triangles. They display the rendered force due to the geometrical and material properties of object using a three degree-of-freedom haptic interface.

More recently, rendering haptic feedback to enable the viewer to follow/sense the motion trajectory of a video object has been proposed. Gaw et al proposed a system for manually annotating haptic motion trajectory information in sync with a movie, tough the haptic information due to the geometry has not been considered in rendering [13]. The system proposed by Yamaguchi et al, generated the haptic effect automatically from 2D graphics using the metadata description of the movement characteristics of the content [14]. Cha et al developed an authoring/editing framework for haptic broadcasting in the context of passive haptic interaction by extending MPEG BIFS. They defined two modes of haptic playback, kinesthetic and tactile. In kinesthetic mode, viewers are forced to follow a predefined motion trajectory with the help of a PD controller, where the applied force is proportional to the distance between the object and the HIP position and derivative of this distance. The tactile mode involves display of pre-determined vibrations to the user while the video plays. The magnitude of the vibration is determined by the gray level of the gray scale tactile effect channel embedded into video clip. The user experiences haptic sensation when a selected character in the video interacts with another character or an object, or performs an action where the trajectory of the motion can be sensed [15]. Rasool and Sourin add haptic information to image and videos as function-based description, where they embed predefined functional representation of the force fields due to motion of the video object [16]. Lastly, Ryden et al demonstrate real-time haptic interaction with a moving object recorded by the Kinect™. They estimate the force displayed to user through the

HIP using the point cloud representation of the object acquired by the camera without considering the relative motion between the object and HIP **[**17**].**

Previous literature on passive haptic interaction with a video deals with either displaying pre-recorded vibrations to the user when an object in the video interacts with another object, or displaying the pre-recorded forces due to the trajectory of a moving object in the scene, or rendering forces due to geometrical and material properties of a video object. There have been no works to consider dynamic forces due to relative acceleration of video objects. Motion is always observed and defined relative to a frame of reference. In a video, there may or may not be camera pan, in addition to one or more moving objects. For example, if the camera pans in the direction of motion of a moving object, the apparent motion of the object shall be slower than its actual motion. Furthermore, the HIP, which we use to sense haptic forces, may or may not be moving. In this paper, the term relative motion refers to the net apparent motion between the HIP and a video object. We note that, unlike in active haptic interaction in virtual reality or gaming applications, in passive interaction we cannot modify the geometry and motion of an object in the video by touching it, but we can experience how the geometry and motion of the video object affects haptic sensation. For example, an object moving towards a stationary HIP will exert a larger force at the instance of collusion compared to the case where the HIP moves along with the object.

The aim of this paper is to introduce a new haptic rendering framework for passive interaction with a video object, given a 2.5D video representation, including dynamic forces due to relative motion between the video object and haptic interaction point (HIP) of the user in addition to the static forces due to the geometry of the video object. The main novelties of this paper are: 1) joint haptic rendering of static and dynamic forces considering the relative motion between a video object and the HIP, and 2) a new depth-based haptic rendering method purely from pixel-based depth data without partial reconstruction of the object surface. The organization of the paper is as follows: Chapter 2 discusses the computation of static and dynamic forces in detail, as well as describing the mappings between graphical and haptic workspaces and the force interpolation process. This chapter also provides information about the necessary video post-processing methods for acquiring more smooth haptic feedback. The next section, Chapter 3 presents the experimental setup and results, while the conclusions of the study are given in Chapter 4.

Chapter 2: Dynamic Force due to Relative Motion in Haptic Video

In this chapter, we describe our implementation for adding haptic effects to a 2D video. We can divide our implementation into two subsections. The first one is the calculation of the haptic rendering forces experienced by the user, while the second one discusses the applied video processing techniques for determining the parameters needed in force calculation.

2.1 Computation of Forces Experienced by the User

Haptic video requires computation of forces that a user would experience as if the user actually touches an object in the video. Since we are using a haptic interface point (HIP), the user would feel as touching the object with a single finger. The total force experienced by the user, $\vec{\mathrm{F}}_{\mathrm{user}}$ can be modeled as the sum of static force, $\vec{\mathrm{F}}_{\mathrm{static}}$, based on the geometrical and material properties of an object and dynamic force, $\vec{\mathrm{F}}_{\mathrm{dynamic}}$, due to the relative motion between an object in the video and the cursor, also called the HIP.

$$\vec{\mathrm{F}}_{\mathrm{user}}(\mathrm{t}) = \vec{\mathrm{F}}_{\mathrm{static}}(\mathrm{t}) + \vec{\mathrm{F}}_{\mathrm{dynamic}}(\mathrm{t}) \qquad 1$$

In computing $\overrightarrow{\mathrm{F}}_{\mathrm{dynamic}}$, the acceleration of a video object is estimated from the video using motion estimation techniques, while the acceleration of the HIP is computed from HIP position data via numerical differentiation.

The force $\vec{\mathrm{F}}_{\mathrm{user}}(\mathrm{t})$ is calculated when the HIP penetrates into the video object. If we define $(x_{haptic}(t), y_{haptic}(t), z_{haptic}(t))$ as the current position of the HIP in the 3D haptic space, we calculate $\vec{\mathrm{F}}_{\mathrm{user}}(\mathrm{t})$ only if $z_{haptic}(\mathrm{t})$ corresponds to the surface or inside the object as determined from the object depth map value at $h_{dep}(x_{img}(t), y_{img}(\mathrm{t}))$, where $(x_{img}(t), y_{img}(\mathrm{t}))$ denotes the HIP position mapped into the image coordinates. This mapping and further implementation details are discussed in Section 2.1.3.

2.1.1 Static Force

This section discusses application of the method proposed by Ho et al. **[**18**]** for haptic rendering of textures mapped onto a 3D geometrical object to the case of calculating the static force (Eqn. 1) for a video frame from an associated depth image, instead of constructing a full 3D geometrical model for the object. Ho et al. **[**18**]** perturb the surface normal of the object, where the HIP contacts object, computed from a 3D geometric model, using the gradient of the texture field, which is defined as a texture height for each pixel.

In order to generate the 3D structure of a video object, we assume that the depth values at each pixel, obtained from a given depth image (which is a height value defined for each pixel), are superposed onto a 2D planar surface. We then compute the surface normal at each pixel using the perturbation method proposed by Ho et al. **[**18**]** The most general form of the expression for calculating the perturbed surface normal, proposed by Ho et al**[**18**]** is given by

$$\vec{\mathrm{M}}_{\mathrm{geo}} = \vec{\mathrm{N}}_{\mathrm{s}} - \vec{\nabla}\mathrm{h}_{\mathrm{dep}} + (\vec{\nabla}\mathrm{h}_{\mathrm{dep}}.\vec{\mathrm{N}}_{\mathrm{s}})\vec{\mathrm{N}}_{\mathrm{s}} \qquad 2$$

where the perturbed surface normal, $\vec{\mathrm{M}}_{\mathrm{geo}}$, is defined as the difference between the original surface normal of the object, $\vec{\mathrm{N}}_{\mathrm{s}}$, and the local gradient of depth map value, $\vec{\nabla}\mathrm{h}_{\mathrm{dep}}$, and added to the projection of the local gradient onto the surface normal at the contact point. Since the image plane is initially 2D, the surface normal at any given point is taken as (0,0,1) by default. Furthermore, the local gradient of the depth map, which is calculated by the Sobel operator around a given point, has only x and y components. Hence, the dot product of $\vec{\nabla}\mathrm{h}_{\mathrm{dep}}$with $\vec{\mathrm{N}}_{\mathrm{s}}$ is equal to zero, which simplifies Eqn. 2. Then, for stability reasons as suggested by Ho et al **[**18**],** the direction and magnitude of the force vector experienced by the user can be calculated by

$$\vec{\mathrm{F}}_{\mathrm{static}} = \begin{cases} (d - \mathrm{Kh})\vec{\mathrm{N}}_{\mathrm{s}} \;\; + \; \mathrm{Kh}\vec{\mathrm{M}}_{\mathrm{geo}} & \text{if } d \geq \mathrm{Kh} \\ d\vec{\mathrm{M}}_{\mathrm{geo}} & \text{if } d \leq \mathrm{Kh} \end{cases} \qquad 3$$

$$d = \alpha \mathrm{h}_{\mathrm{dep}}\big(x_{img}(t)\,, y_{img}(t)\big) - \; z_{haptic}(t) \qquad 4$$

In Eqn. 3, K is a scalar that depends on the properties of the surface texture, while h is the normalized depth value at the current HIP. As shown in Eqn. 4, d denotes depth of penetration into the object surface. Penetration is the difference between the depth image (map) value $\mathrm{h}_{\mathrm{dep}}$ at the HIP position mapped into image coordinates $(x_{img}(t)\,, y_{img}(t))$ and the z-component of the HIP in the haptic workspace, $z_{haptic}(t)$. Note that, $\mathrm{h}_{\mathrm{dep}}$ is scaled by α in order to convert the depth code value to the physical units of the haptic device workspace.

2.1.2 Dynamic Force

One of the main novelties of this paper is introducing the concept of dynamic force experienced by a user due to relative motion between the video object and the HIP. The motion of a video object actually represents the relative motion between the corresponding scene object and the camera. For example, in the case of a camera pan, the apparent motion of the video object may feel larger or smaller than its actual motion depending on the direction of the pan. Since typically we do not have access to camera calibration parameters, force experienced by the user is calculated based on the apparent motion of the video object relative to the HIP. Since the HIP can be stationary, or moving towards or away from the video object, the dynamic component of the force experienced by the user will vary according to the relative motion between the video object and the HIP.

For simplicity, we assume that the object has a constant velocity from one frame to another and calculate its acceleration using

$$\vec{a}_{object} = \frac{\vec{\Delta} v}{\Delta t} = \frac{25\, \vec{v}_{flow}}{10^{-3}} \qquad \frac{pixel}{(sec)^2} \qquad 5$$

where $\vec{\mathrm{v}}_{\mathrm{flow}}$ is velocity of the object found by the optical flow algorithm (defined in Section 2.2.2) and $\vec{\mathrm{a}}_{\mathrm{object}}$ is the acceleration of the video object. Note that, we scale $\vec{\mathrm{v}}_{\mathrm{flow}}$ by 25 assuming that the video runs at 25 fps and we calculate object acceleration for each haptic loop, hence Δt is taken as 1 msec.

We then assign a mass to the video object such that the maximum rendered force will be in the range of the maximum allowable force of the haptic device. Finally, we calculate the force due to motion using Newton's second law given by

$$\vec{F}_{dynamic} = \beta m_{object}\vec{a}_{object} - m_{cursor}\vec{a}_{cursor} \quad \text{(Newton)} \qquad 6$$

where m_{object} and m_{cursor} are the masses assigned to the object and the cursor respectively, $\vec{a}_{cursor}$ is the acceleration of the cursor, while β is a constant to convert the object acceleration from units of the image workspace to that of the haptic workspace.

2.1.3 A New Implementation of Depth Image Based Haptic Rendering

Another novelty of this paper is to introduce a new depth-based haptic rendering method using only pixel-based depth data without any partial reconstruction of object surface. The steps of defining a mapping between image and haptic space coordinates and interpolation of depth at the HIP position are described below. We assume that the user interacts with video using a 3 degree-of-freedom HIP, which provides force feedback. The x-y coordinates of the point of contact of the HIP with the video is shown as a 3D cone on the image plane. The size of the cone is scaled according to the z coordinate of the HIP to provide the user with a sense of depth on the screen.

Mapping between Image Coordinates and Haptic Coordinates: As a first step, the position of the HIP in the physical workspace of the haptic device must be registered with the image coordinates. We can map the x-y coordinates of HIP position $\left(x_{haptic}(t), y_{haptic}(t), z_{haptic}(t)\right)$ in the physical workspace to $(x_{img}(t), y_{img}(t))$ on the image plane using the linear relations:

$$x_{img}(t) = a_x\, x_{haptic}(t) + b_x \qquad 7$$

$$y_{img}(t) = a_y\, y_{haptic}(t) + b_y \qquad 8$$

where the coefficients

$$a_x = \frac{w}{(x_{haptic}^{max} - x_{haptic}^{min})} \qquad b_x = -a_x x_{haptic}^{min} \qquad 9$$

$$a_y = \frac{h}{(y_{haptic}^{max} - y_{haptic}^{min})} \qquad b_y = -a_y y_{haptic}^{min} \qquad 10$$

Here $(x_{haptic}^{min}\ ,\ y_{haptic}^{max})$ and $(x_{haptic}^{max}\ ,\ y_{haptic}^{min})$ correspond to the top-left, and bottom-right corners of the haptic workspace and w and h represent the width and the height of the image. The bias terms b_x and b_y are needed since x_{haptic}^{min} and y_{haptic}^{min}can be negative and the pixel indexes of the image must be integers in the range of $[0,w)$ and $[0,h)$, respectively.

Interpolation of Depth Map: In order to determine the image depth at the current HIP position $\left(x_{img}(t)\,, y_{img}(t)\right)$ in the image plane, which is in general a non-integer value, we employ bilinear interpolation from the depth values at the neighboring pixels as depicted in Figure 1.

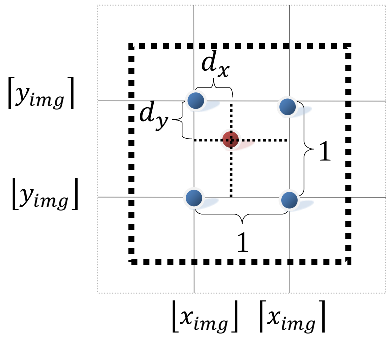


Figure 1 : Bilinear interpolation of the scene/object depth at the HIP position $(\boldsymbol{x_{img}}\,, \boldsymbol{y_{img}})$ in the image plane, shown by the red dot, using depth map values at the neighboring image pixels, shown by blue dots.

In Figure 1, the red dot shows the actual HIP position in the image plane $\left(x_{img}(t)\,, y_{img}(t)\right)$ and blue dots show the neighboring pixels. The weights used in the bilinear interpolation are inversely proportional to the distances of the neighboring pixels to $\left(x_{img}(t)\,, y_{img}(t)\right)$

$$d_x = \left|x_{\mathrm{img}} - \lfloor x_{\mathrm{img}} \rfloor\right| \qquad d_y = \left|y_{img} - \lfloor y_{\mathrm{img}} \rfloor\right| \qquad 11$$

The depth value corresponding to the HIP position is then calculated using the bilinear interpolation as

$$\begin{aligned} h_{dep}(x_{img}, y_{img}) = {} & d_x d_y\, h_{dep}(\lfloor x_{img} \rfloor, \lceil y_{img} \rceil) + (1-d_x)\, d_y h_{dep}(\lceil x_{img} \rceil, \lceil y_{img} \rceil) \\ & + d_x(1-d_y)\, h_{dep}(\lfloor x_{img} \rfloor, \lfloor y_{img} \rfloor) + (1-d_x)(1-d_y)\, h_{dep}(\lceil x_{img} \rceil, \lfloor y_{img} \rfloor) \end{aligned} \qquad 12$$

The penetration, which was defined in Eqn. 4, can then be calculated as the difference between $\mathrm{h_{dep}}$ and the z-coordinate of the HIP. However, in order to take this difference, the dynamic range of the depth image needs to be mapped to the workspace of the haptic device along the z-axis linearly such that the range of depth values between 0-255 corresponds to the range of movements of HIP between 0-30 mm along the z-axis in the haptic workspace.

Calculation of Accelerations and Forces: For the computation of dynamic force, we need to estimate the acceleration of the video object and the HIP. To this effect, we first estimate HIP velocity as the difference of the HIP position in time

$$v_{cursor_x} = x_{haptic}(t) - x_{haptic}(t-1) \quad 13$$

$$v_{cursor_y} = y_{haptic}(t) - y_{haptic}(t-1) \quad 14$$

and then the acceleration can be computed similarly as the difference of HIP velocity in time.

We convert the object acceleration to the units of mm/sec^2 using the inverse of linear mapping defined in Eqn. 7. Then, we can calculate the $\vec{F}_{\mathrm{dynamic}}$ using the Eqn. 6. The sum of $\vec{F}_{\mathrm{dynamic}}$ and $\vec{F}_{\mathrm{static}}$ is rendered as the total force that is displayed to the user through the haptic device. . For smooth and stable haptic display, we interpolate static and dynamic forces in between the video frames to the rate of the haptic device as described in the next subsection.

Temporal Interpolation of Forces to the Rate of the Haptic Device: Smooth and stable force rendering requires the haptic display rate to be significantly higher than the video/graphic frame rate **[**5**].** For example, the haptic display rate is typically 1 kHz, while the video/graphics frame rate is 25 Hz. Hence, we need to interpolate forces calculated at the video frame rate to the haptic display rate for effective haptic experience. To this effect, cubic/spline interpolation can be used assuming that the cursor position is fixed between consecutive video frames. Hence, if the cursor is at the position (x_1, y_1) in frame i and the calculated total force is $F_{total}^{i}(x_1, y_1)$, we interpolate forces between $F_{total}^{i}(x_1, y_1)$ and $F_{total}^{i+1}(x_1, y_1)$, which is the total force calculated at cursor position (x_1, y_1) in frame i+1, in the temporal dimension using cubic interpolation.

2.1.4 Selection of Parameters for Dynamic Force

The selection of proper mass values for the cursor and video object should be addressed carefully considering applicable physical laws, since we do not have access to the physical video object. Since the velocity of a video object cannot be altered as a result of the interaction by the user, we assume that collision of the HIP with a video object shall only affect the HIP; whereas in real life, collision would affect both parties. Hence, for a real-life like experience, the mass of the video object should be chosen much larger than that of the HIP so that the velocity of the video object would not be altered by the interaction. For example; the cursor would act like a fly interacting with a horse. Even though a fly may have acceleration, it cannot change the velocity or direction of motion of the horse. However, a fly can experience the motion of the horse and dynamic forces due to its movement.

The way users grip the haptic device affect the experienced force by them, as well as the measured velocity, acceleration of the HIP by the device sensors. Hence, normally we need to approximate the effective mass of the cursor for each step of the interaction, which would result in a changing mass for the cursor and make the force calculation complex. Therefore, we neglected the effect of the grip and considered the apparent mass of the haptic device at the tip, which is 0.045kg as measured by the manufacturer of the device and stated in the data specifications sheet **[**19**].** Then we assign the value for the video object so that $m_{object} \gg m_{cursor}$.

2.2 Video Processing for Haptic Rendering

In order to compute reliable haptic information from 2D video clip and associated depth images, depth images for the video object should be smooth and detailed. The depth images are typically quantized with 8 bits/pixel, and hence, the depth values are in the range [0,255]. If the range of depth in the scene (depth of field) is large, the quantization can cause loss fine detail regarding the structure/geometry of some objects in the scene. Therefore, post processing of depth images may be required to smooth depth data and improve the haptic experience. Furthermore, motion estimation from video and associated depth images is required to estimate the acceleration of video objects in order to render dynamic forces. These post-processing operations are described in detail in the following.

2.2.1 Enhancement of the Depth Images

The first video processing step is to segment “the touchable object” or the so-called “hot object” from the rest of the scene at each frame. Hence, we would represent the geometrical information of touchable video object more precisely within the range of the depth-map. To segment the object, we use both the color and depth images. We apply the well-known Canny edge detection algorithm to find the edges of the object of interest **[**20**]**. We fuse the color edge images and edges of depth images, in order to create the object segmentation mask as shown in Figure 2, where segmentation of the horse from rest of the frame is shown.

Figure 2: Illustration of video object segmentation using (a) original frame [21] (b) corresponding depth image, (c) computed object segmentation mask, (d) enhanced depth image.

After the video object is segmented, depth values outside the object mask are set equal to zero, and depth values within the object mask are rescaled between the minimum and maximum depth values throughout the video clip. Finally, we apply a Gaussian low pass filter to smooth depth images for a more realistic haptic experience.

2.2.2 Motion Estimation

In the most general case, precise calculation of the relative motion/acceleration between a video object and the HIP requires estimation of 3D motion of the object from the video and associated depth images, so that we can also compute the z-component of the object acceleration (in addition to x and y components) for dynamic force calculation using Eqn. 6. Estimation of 3D motion and structure from video has been extensively studied in the literature and some widely used methods and procedures exist [22]. These methods can be classified as 3D motion and structure using point correspondences or using dense 2D optical flow. In the former, the first step is extracting common features between successive frames which are registered automatically or manually. The next step requires estimation of the 3D motion and structure parameters using either the essential matrix method [23,24] or the factorization method proposed by Takosi and Kanade [25] that can be formulated for affine or perspective projection models. Since the haptic interaction by the user cannot affect the motion of objects in the video, the computation of 3D motion parameters can be done offline.

In certain special cases, such as 2D motion on a plane perpendicular to the camera or motion purely along the z direction (towards to or away from the camera), there are simpler approaches to estimate acceleration $\vec{a}_{object_x}$, $\vec{a}_{object_y}$, and $\vec{a}_{object_z}$.In the former $\vec{a}_{object_z} = 0$, in the latter $\vec{a}_{object_x} = \vec{a}_{object_y} = 0$, and $\vec{a}_{object_z}$ can be estimated by processing depth map images.

Special case $\vec{a}_{object_z} = 0$*:* If the video object performs 2D planar motion in a plane that is perpendicular to the camera, the dense 2D velocity, and hence acceleration, can be estimated using the Lucas-Kanade **[**26**]** optical flow algorithm over the segmented textured images (see Figure 3 for some sample frames of a synthetic video). This algorithm provides us with the x and y component of the 2D object motion with respect to the previous frame. The outlier motion vectors can be filtered using a 2D median filter to obtain a smoother motion field.

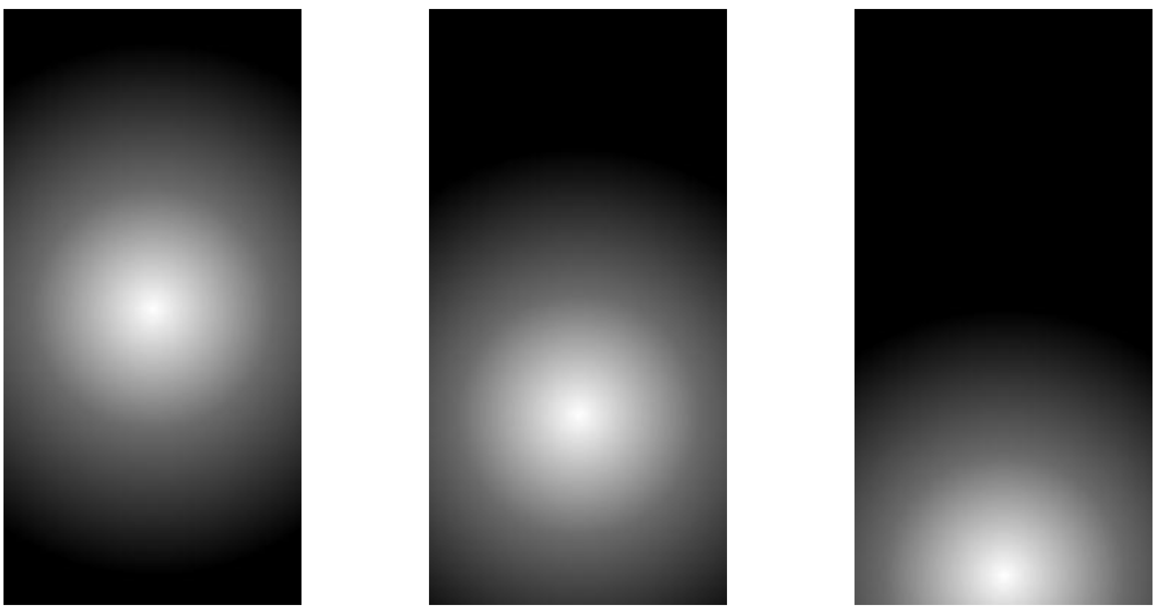


Figure 3: Sample frames from the synthetic video with $\vec{a}_{object_z} = 0$. Here, a ball moves vertically down in a plane that is perpendicular to the optical axis of the camera.

Special case $\vec{a}_{object_x} = \vec{a}_{object_y} = 0$*:* If the motion of the object is purely along the z direction, we can estimate object motion from the variation of the depth observed from the depth map images (see Figure 4 for some sample frames of a synthetic video). Since we assume the object structure does not change (rigid object) through the video, the computed differences in depth between the frames will be due to the motion of the object.

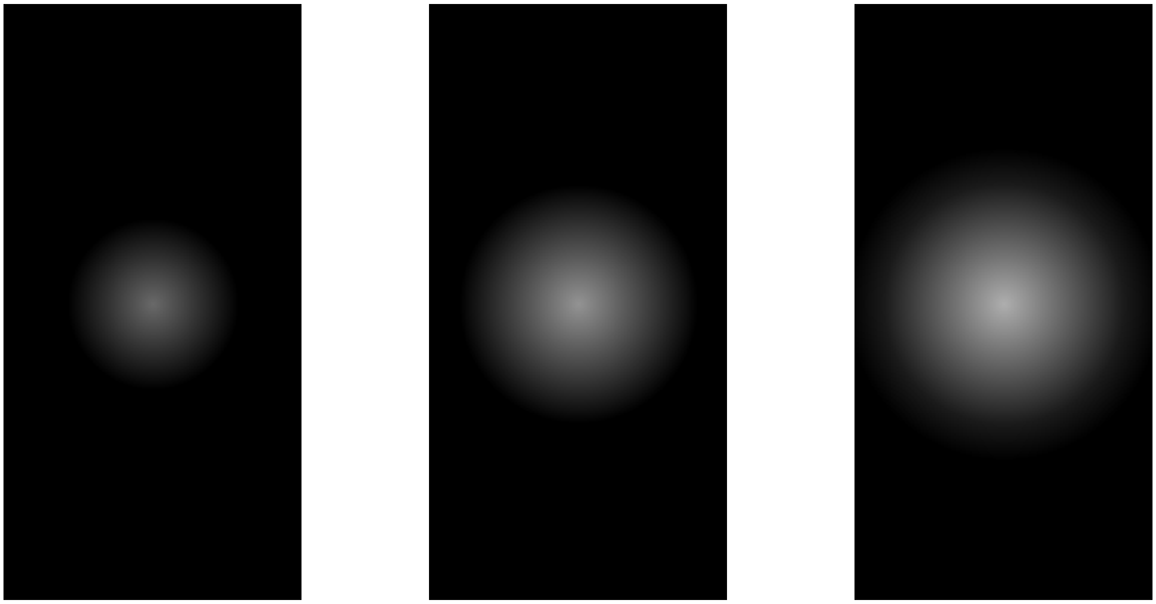


Figure 4: Sample frames from the synthetic video with $\vec{a}_{object_x} = \vec{a}_{object_y} = 0$. Here a ball moves towards the camera along the optical axis of the camera.

Chapter 3: Experimental Setup and Results

For our experiments, we used the Omni haptic device (by Sensable Technologies) providing 6 degrees of freedom (dof) sensing and 3-dof force feedback. The users interacted with the video passively interact with the video through the HIP. They experienced both the static and dynamic forces during the interactions of the HIP with a running video. They could also pause the video at a particular frame and feel the geometry of the object alone (i.e. static force).

In order to demonstrate the fundamental concepts discussed in this paper, we generated two synthetic video clips of a bouncing ball. In the first video clip, the ball moves vertically up and down perpendicular to the camera, while in the second one it moves back and forth towards the camera along the z axis. We used these synthetic videos to perform controlled experiments. We also provided some results on a real “horse” video.

3.1 Simulation Results

Static force only: In Figure 5, a slice of the ball is shown. The rendered forces are drawn using the blue arrows while the black curve represents the depth values of the ball at the specified x and y positions of HIP. The cursor moves only along the y axis, hence only the y and z axes are shown in Figure 5.

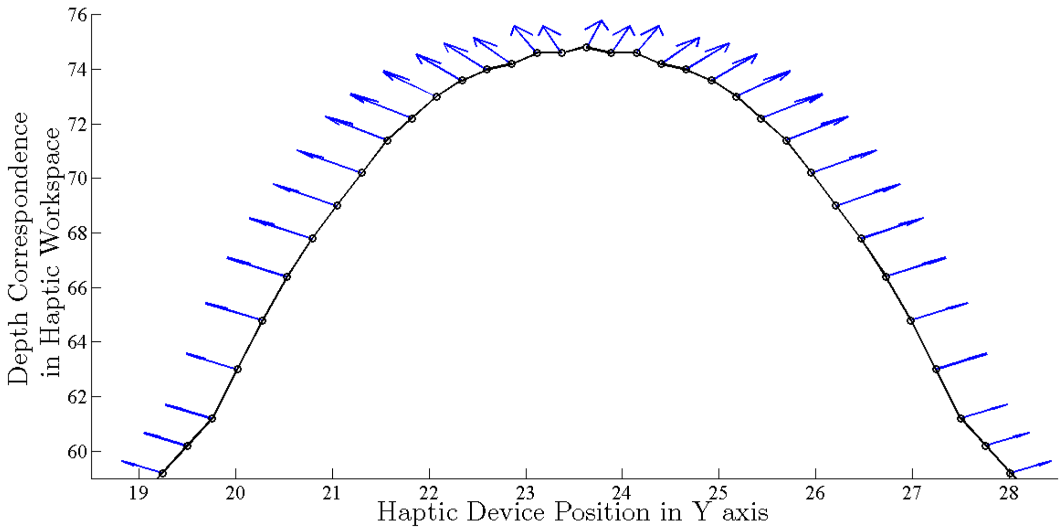


Figure 5: The static forces rendered for a slice of the ball.

Special case 1: the ball moves perpendicular to the camera:

1. When a_{cursor} equals zero: In Figure 6, blue curve shows the static force rendered when the cursor is stable and the red curve shows $\vec{F}_{\text{user}}$, the total force rendered when the motion of the video object is added to the static force when the cursor is not moving. Since x_{haptic} is not changing, we plotted the rendered forces in y vs. z axes. The x component of the force is constant and due to the geometry of the video object. When the cursor is not moving, there isn't any force exerted due to its motion. Since the ball's acceleration is along the +y direction, the total force experienced by the user, shown as red, is larger than the static force alone, shown as blue. In addition to that, the acceleration of the object is changing between the frames. Hence, the effect of dynamic force is changing due to the relation defined in Eqn. 6

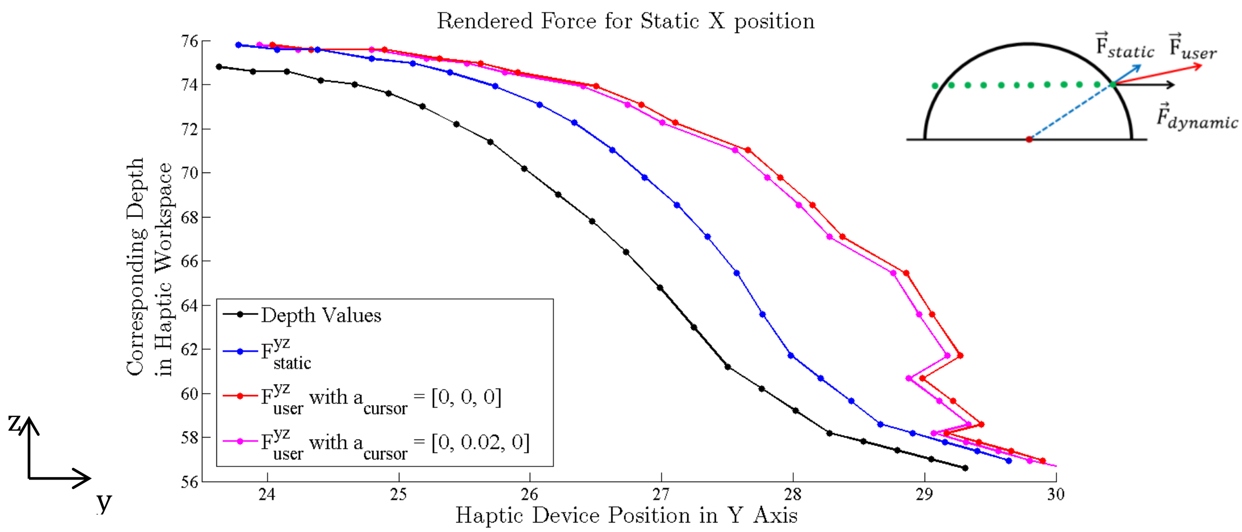


Figure 6: The profile of the force vectors rendered for a half slice of the ball a) when only the static forces exist (blue curve), b) when the cursor is not moving but the object has an acceleration along the +y direction (red curve) and c) when the ball moves relative to the HIP and also perpendicular to the camera along the +z direction (pink curve).

2. When a_{cursor} is constant: The effect of relative motion of the video object with respect to the HIP motion is shown in Figure 6. The HIP has a constant acceleration along the x and the y directions. Hence, the total rendered forces are perturbed in the x and the y directions with respect to the rendered forces shown in Figure 6.

<u>Special case 2: When the ball moves towards the camera:</u>

1. When a_{cursor} equals zero: We assume that the structure of the object is not changing when it moves towards the camera, though its visual size changes due to the perspective effects. Hence, the depth changes in the scene are due to the motion of the object and not due to the camera. Since the object moves towards the camera, there is an additional dynamic force along the +z direction. The effect of this additional force is shown as red curve in Figure 7. Note that the force for the indices less than 46.6 along the z axis is zero since there is no collision between the HIP and the video object. The HIP position is shown as the green dashed line, while blue and red curves represents the static force and the total rendered force with dynamic force added respectively.

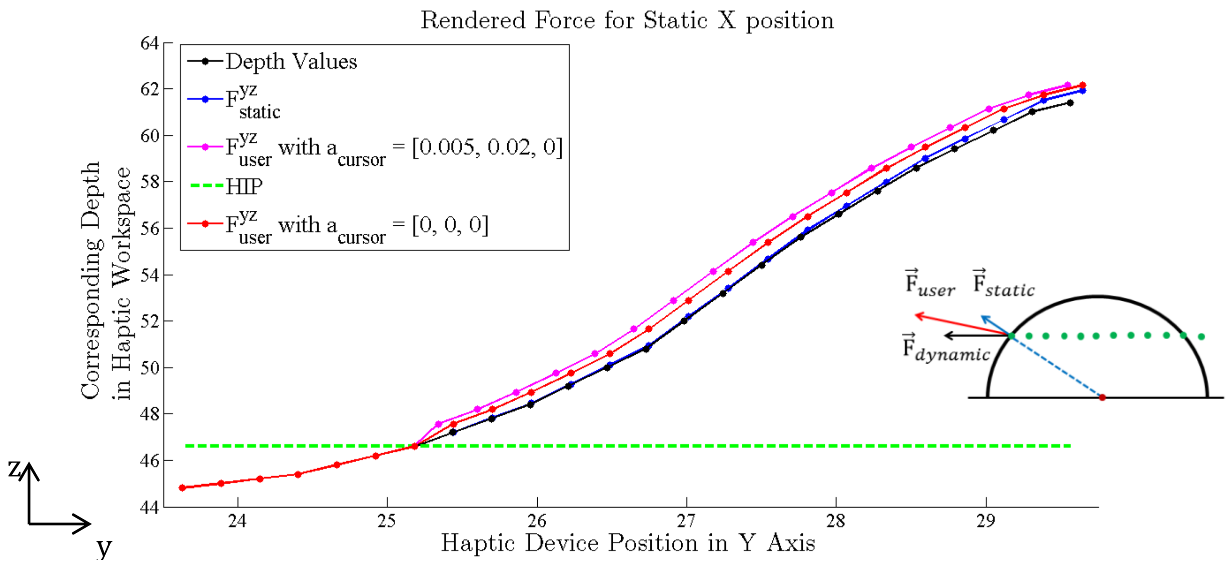


Figure 7: The profile of the static and the total forces rendered for a slice of the ball when the ball moves relative to the HIP and also towards the camera

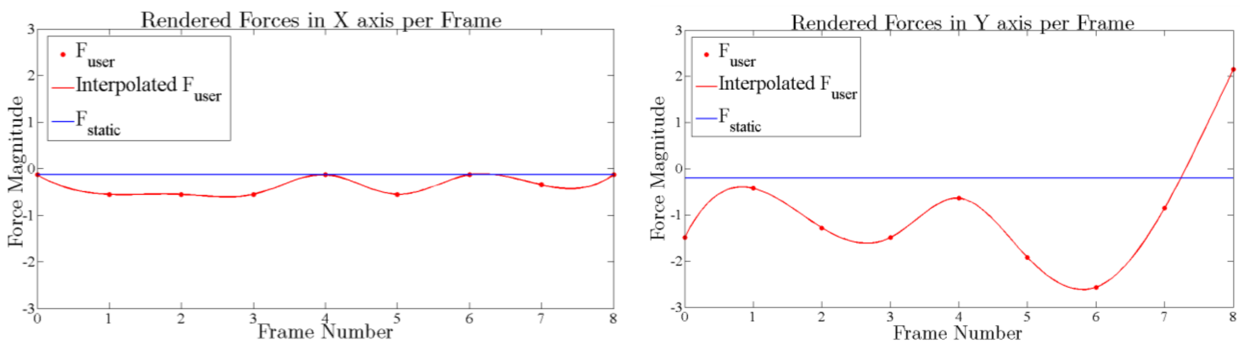
2. When a_{cursor} is constant: The effect of the relative motion of the ball with respect to the movements of the HIP when the object moves towards the camera is also shown in Figure 7 . The HIP has a constant acceleration along the x and the y directions. Since the force due to the acceleration of the ball has a component along the z direction only, the x and y components of the total rendered force is due to the movements of the cursor. In Figure 7, we see that the acceleration of the cursor is in the same direction of the video object and hence the experienced force by the viewer is decreasing along the y direction. In addition to that, since the object moves along the +z direction, the depth value increases from one frame to another which causes the $\vec{F}_{static}$ to increase. However, we can see from the figure that the change in static force is not that significant when compared with the change in dynamic force due to the motion of the video object.

3.2 Real Video Clip Results

The "horse" video, where the horse raises his head, has been used as a real video clip for testing the proposed haptic rendering approach (see **Figure 8** for some sample frames). While calculating the acceleration of the horse's head, since the movement of horse towards the camera is relatively small to its movement perpendicular to the camera, we assumed $\vec{a}_{object_z} = 0$. Hence we only calculated the $\vec{a}_{object_x}$ and $\vec{a}_{object_y}$ from the video clip.

Figure 8 : Sample frames from a real video clip [**21**]**.** Here the horse slowly raises his head.

In Figure 9, plots of the static and total forces in the x and y axis are shown. In rendering the dynamic forces, the cursor is like virtually coupled (i.e. attached) to the same spot on the horse, hence the static force due to the "sticking" effect is constant; but the dynamic force varies due to the acceleration of the horse head in the video clip. The red curves on the plots represent the total force due to the "sticking" effect and the acceleration of horse head at the HIP position.



a b

Figure 9: The plot of the static and the total rendered forces in (a) x-axis and (b) y-axis. A cubic interpolation is used to connect the discrete force values calculated at each video frame and then display them to the user at the haptic update rate 1kHz.

The user can also pause the video and feel the static force due to the geometry of the horse, generated from the depth data. The smoothness of the depth data is important for the user experience since it directly affects the gradient calculation and hence the direction of the static force vector displayed to the user through the haptic device. In order to reduce the noise in the depth data and make the haptic experience more enjoyable, a low pass filter was applied to the depth data.

Chapter 4: Conclusion

We proposed a method for haptic rendering of the dynamic force due to the motion of the video object relative to the HIP interacting with the object. In our approach, we render both the static force due to the geometry of the object and the dynamic force due to the relative motion of the object for enhancing the user experience. The earlier studies in this area have focused on the rendering of the static forces due to the geometry of the video object and neglected the dynamic effects. However, in video applications involving highly dynamic scenes such as a ball kicked by a soccer player, a bullet fired by a shooter, or a jet plane making acrobatic movements in the air, displaying the forces to the user due to the inertia of the video object (the ball, the bullet, and the jet plane in the examples) is important for the immersive experience. As shown in Figure 6, depending on the mass and the acceleration of the video object with respect to that of the HIP, the magnitude and the direction of these forces can significantly alter the total force displayed to the user (compare the separation distances of the red and blue colored force profiles from the black colored depth profile in Figure 6). However, it is important to emphasize that the user in our system feels what is visually displayed to him/her in the video and cannot edit the content. Due to this limitation, the user in our system is a passive observer. In other words, she/he cannot change the state of the object in video by pushing or pulling it. This also enforces us to select the mass of the video object much larger than the effective mass of the device at HIP, which is reported as 45 mg by the manufacturer of the device. Hence, the momentum transferred from the user to the video object is minimal and does not cause a change in its state. Since the mass of the object is selected significantly higher than that of the cursor, the dynamic force felt by the user is mainly due to the acceleration of the video object. We also assumed that that the scene motion along the z direction is relatively small compared to the ones in x-y plane and can be neglected to simplify the computation of dynamic force. As a result, the optical flow computed from the video directly corresponds to the movements of the video object. Hence in our implementation, we deal with the objects having motion either in x, y directions or z direction. When there is motion along the z direction, we assumed the object is rigid and not changing its structure or the pose.

BIBLIOGRAPHY

[1] M. A. Srinivasan and C. Basdogan, "Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges," *Computers and Graphics*, vol. 21, no. 4, pp. 393-404, 1997.

[2] C. Basdogan, C. Ho, M. A. Srinavasan, and M. Slater, "An Experimental Study on the Role of Touch in Shared Virtual Environments ," *ACM Human Computer Interactions*, vol. 7, no. 4, pp. 440-463, 2000.

[3] C. Basdogan and M. A. Srinavasan, "Haptic Rendering in Virtual Environments," in *Handbook of Virtual Environments: Design, Implementation, and Applications*, K.M. Stanney, Ed. New Jersey: Lawrence Erlbaum Associates, 2001, pp. 117-134.

[4] N. Magnenat-Thalmann and U. Bonanni, "Haptics in Virtual Reality and Multimedia," *IEEE Multimedia*, vol. 13, no. 3, pp. 6-11, July 2006.

[5] K. Salisbury, F. Conti, and F. Barbagli, "Haptic Rendering: Introductory Concepts," *IEEE Computer Graphics and Applications*, vol. 24, no. 2, pp. 24-32, March 2004.

[6] S. O'Modhrain and I. Oakley, "Touch TV: Adding Feeling to Broadcast Media," *In Proc. European Conf. Interactive Television: from Viewers to Actors*, pp. 41-47, December 2003.

[7] S. O'Modhrain and I. Oakley, "Adding Interactivity: Active Touch in Broadcast Media," *In Proc. Int. Symp. Haptic Interfaces for Virtual Environments and Teleoperator Systems (HAP TICS'04)*, pp. 293-294, March 2004.

[8] K. Müller et al., "Reliability-based Generatıon and View Synthesis in Layered Depth Video," *Proc. IEEE International Workshop MMSP*, October 2008.

[9] A. Smolic and P. Kauff, "Interactive 3-D Video Representation and Coding Technologies," *Proc. of the IEEE*, vol. 93, no. 1, pp. 98-110, January 2005.

[10] J. Cha, S. Kim, Y. Ho, and J. Ryu, "3D Video Player System with Haptic Interaction based on Depth Image-Based Representation," *IEEE Trans. Consumer Electronics*, vol. 52, no. 2, pp. 477-484, May 2006.

[11] J. Cha, S-M. Kim, I. Oakley, J. Ryu, and K. Lee, "Haptic Interaction with Depth Video Media," in *Proc. PCM (1)*, 2005, pp. 420-430.

[12] S. P. Walker and J. K. Salisbury, "Large Haptic Topographic Maps: MarsView and the Proxy Graph Algorithm," *Proc. ACM SIGGRAPH*, pp. 83-92, 2003.

[13] D. Gaw, D. Morris, and K. Salisbury, "Haptically Annotated Movies: Reaching out and Touching the Silver Screen," *14th Symposium on Haptic Interfaces for Virtual Enviroment and Teleoperator Systems*, pp. 287-288, 2006.

[14] T. Yamaguchi, A. Akabane, J. Murayama, and M. Sato, "Automatic Generation of Haptic Effect into Published 2D Graphics," *In Proc. EuroHaptics 2006*, July 2006.

[15] J. Cha, Y. Seo, Y. Kim, and J. Ryu, "An Authoring/Editing Framework for Haptic Broadcasting: Passive Haptic Interactions using MPEG-4 BIFS ," *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*, pp. 274-279, 2007.

[16] S. Rasool and A. Sourin, "Towards Tangible Images and Video in Cyberworlds - Function-based Approach," *2010 International Conference on Cyberworlds*, pp. 92-96, 2010.

[17] F. Ryden, H. J. Chizeck, S. N. Kosari, H. King, and B. Hannaford, "Using Kinect and a Haptic Interface for Implementation of Real-Time Virtual Fixtures," *RGB-D: Advanced Reasoning with Depth Cameras Workshop, RSS 2011*, June 2011.

[18] C. H. Ho, C. Basdogan, and M. A. Srinavasan, "Efficient Point-Based Rendering Techniques for Haptic Display of Virtual Objects," *Presence: Teleoperators and Virtual Environments*, vol. 8, no. 5, pp. 477-491, October 1999.

[19] http://www.sensable.com/documents/documents/PHANTOM_Omni_Spec.pdf.

[20] J Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-714, November 1986.

[21] Mobile3DTV: Video plus depth database. www.mobile3dtv.eu/video-plus-depth/.

[22] D. P. Robertson and R. Cipolla, "Structure from Motion," in *Practical Image Processing*

and Computer Vision, M. Varga, Ed.: John Wiley, 2009.

[23] R. I. Hartley and P. Sturm, "Triangulation," *American Image Understanding Workshop*, pp. 957-966, 1994.

[24] R Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*.: Cambridge University Press, ISBN: 0521623049, 2000.

[25] C. Tamosi and T. Kanade, "Shape and Motion from Image Streams under Ortograhpy: A Factorizaion Method ," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137-154, 1992.

[26] T. Kanade and B. D. Lucas, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. DARPA IU Workshop*, pp. 121-130, 1981.

[27] E. Sallnas, K. Rasmuss-Grohn, and C. Sjostrom, "Supporting Presence in Collabrative Enviroments by Haptic Force Feedback," *ACM Transactions on Computer Human Interaction*, vol. 7, pp. 461-476, 2000.