Dynamic Haptic Interaction with Video

Nuray Dindar, A. Murat Tekalp, and Cagatay Basdogan
College of Engineering, Koc University, Istanbul, 34450, Turkey
ndindar, mtekalp, cbasdogan@ku.edu.tr

This chapter introduces the notion of passive dynamic haptic interaction with video, and describes the computation of force due to relative motion between an object in a video and the haptic interface point (HIP) of a user, given associated pixel-based depth data. While the concept of haptic video, that is, haptic rendering of forces due to geometry and texture of objects in a video from the associated depth data, has already been proposed, passive dynamic haptic interaction with video has not been studied before. It is proposed that in passive dynamic interaction, a user experiences motion of a video object and dynamic forces due to its movement, 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 acquired by the encoders of the haptic device. Mass values are assigned to the video object and HIP such that user interaction shall not alter the motion of the video object according to the laws of physics. Then, the dynamic force is computed by using Newton's second law. Finally, it is scaled 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 the difference in rendered forces with and without including the dynamics.

I. 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 (Srinivasan and Basdogan, 1997; Basdogan et al., 2000; Basdogan and Srinivasan, 2001; Magnenat-Thalmann and Bonanni, 2006; Salisbury et al., 2004). The potential of adding haptic force feedback into 2D broadcast video has first been discussed by O'Mohdrain and Oakley (O'Modhrain and Oakley, 2003, 2004). 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 interaction with video has advanced to include rendering structure and texture. Haptic structure refers to touching/feeling the 3D geometry of a video object. The 3D structure can be rendered using full 3D polygon or mesh models or 2.5D pixel-based depth data. A video together with associated depth data for each pixel is called a "2.5D representation" in the computer vision literature and "video-plus-depth representation" in the MPEG (Motion Picture Experts Group) community. Since recent MPEG standards for 3D video rely on such representations, for example the multi-view video plus depth (MVD) or layered depth video (LDV) representations of 3D video (Müller et al., 2008; Smolic and Kauff, 2005), it is expected that video together with associated depth images will be widely available in the future and can be used for haptic interaction. Haptic texture refers to rendering surface properties (e.g., roughness) of objects in a video, which can be recorded and encoded as separate video channels. Cha et al. have been the first to propose computing haptic information from depth images rather than full 3D object/scene models (Cha et al., 2006, 2005). They have modified the proxy graph algorithm (Walker and Salisbury, 2003) for haptic rendering of depth images by constructing a triangular surface model from the depth data and taking the projection of the line segment constructed between the current and previous haptic interface point (HIP) positions onto the depth image to determine a list of candidate triangles for the collision detection. Then, the collision between the HIP and the video object is detected using this list of candidate triangles. Using this approach, they displayed the rendered force due to the geometrical properties of the object via a three degree-of-freedom haptic interface.

More recently, rendering forces 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, though the haptic information due to the geometry has not been considered in rendering (Gaw et al., 2006). 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 (Yamaguchi et al., 2006). Cha et al. developed an authoring/editing framework for haptic broadcasting in the context of passive haptic interaction by extending MPEG binary format for scene (BIFS). They defined two modes of haptic playback, kinesthetic and tactile. In the kinesthetic mode, viewers are forced to follow a predefined motion trajectory with the help of a proportional-derivative (PD) controller, where the applied force is proportional to the distance between the object and the HIP position and the 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 scale level in the tactile effect channel embedded into the 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 (Cha et al., 2007). Rasool and Sourin add haptic information to image and videos as a function-based description, where they embed predefined functional representation of the force fields due to motion of a video object (Rasool and Sourin, 2010). Lastly, Ryden et al. demonstrate real-time haptic interaction with a moving object recorded by Kinect$^{\mathrm{TM}}$. They estimate the force displayed to the user using the contacts between the HIP and the point cloud representation of the object acquired by the camera without considering the relative motion between the object and the HIP (Ryden et al., 2011).

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 pre-recorded forces due to the trajectory of a moving object in the scene, or rendering forces only due to the geometrical and material properties of a video object. There is not any earlier study that considers the computation of dynamic forces due to the relative acceleration of video objects. Two important observations regarding the passive dynamic haptic interaction are: i) Motion is always observed and defined relative to a frame of reference. For example, if there is camera pan 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 is used to interact with the object, may or may not be moving. In this chapter, the term “relative motion” refers to the apparent motion of a video object. ii) Unlike in active haptic interaction such as in virtual reality or game applications, in passive interaction, neither the geometry nor the motion of an object in the video can be modified by interacting with it, but the geometry and motion of the video object can be experienced by users simultaneously. To this effect, the HIP should act like a “fly” interacting with a “horse”. Even though a fly may have acceleration, it cannot change the velocity or direction of the motion of a horse. However, a fly moving on the surface of a horse can feel its geometry as well as the dynamic forces due to its motion.

The aim of this chapter is to introduce passive dynamic haptic interaction with a video object, given a 2.5D video representation, considering the dynamic forces due to the relative motion of a video object with respect to the HIP in addition to the static forces due to the geometry of the video object. The main novelties of our approach include: 1) a method for computation of dynamic forces for passive interaction considering the relative motion between a video object and the HIP, and 2) a new haptic rendering method based on pixel-based depth data without any partial reconstruction of the object surface. The organization of the chapter is as follows: Section II discusses the computation of static and dynamic forces, as well as describing the mapping between graphical/image and haptic workspaces and force interpolation. Section III describes video processing methods for acquiring more smooth haptic feedback. Section IV elaborates on the experimental setup and results, while conclusions are given in Section V.

II. COMPUTATION OF FORCE EXPERIENCED BY THE USER

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

$$\vec{F}_{user} = \vec{F}_{static} + \vec{F}_{dynamic} \tag{1}$$

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

The force $\vec{F}_{user}(t)$ is calculated when the HIP penetrates into the video object. If $(x_{haptic}(t), y_{haptic}(t), z_{haptic}(t))$ are defined as the current position of the HIP in the 3D haptic space, $\vec{F}_{user}(t)$ can be calculatede only if $z_{haptic}(t)$ is inside the object as determined from the object depth map value at $h_{dep}(x_{img}(t), y_{img}(t))$, where $(x_{img}(t), y_{img}(t))$ denotes the HIP position mapped into the image coordinates. This mapping and further implementation details are discussed in Section II-C.

A. *Static Force*

This section discusses a modification of the method proposed by Ho et al. (1999) for haptic rendering of textures mapped onto a 3D geometrical object. Ho et al. (1999) 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, it can be assumed 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. Then the surface normal at each pixel are computed using the perturbation method proposed by Ho et al. (1999) The most general form of the expression for calculating the perturbed surface normal, proposed by Ho et al.(1999), is given by

$$\vec{M}_{geo} = \vec{N}_s - \vec{\nabla} h_{dep} + (\vec{\nabla} h_{dep} \vec{N}_s)\vec{N}_s \tag{2}$$

where, the perturbed surface normal, $\vec{M}_{geo}$, is defined as the difference between the original surface normal of the object, $\vec{N}_s$, and the local gradient of depth map value, $\vec{\nabla} h_{dep}$, 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 $\vec{\nabla} h_{dep}$ with $\vec{N}_s$ is equal to zero, which simplifies (2). Then, for stability reasons as suggested by Ho et al. (Ho et al., 1999), the direction and magnitude of the force vector experienced by the user can be calculated by

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

$$d = \alpha h_{dep}(x_{img}(t), y_{img}(t)) - z_{haptic}(t) \tag{4}$$

In (3), K is a scalar that depends on the surface texture properties, while h is the normalized depth value at the current HIP position. In (4), d denotes depth of penetration into the object surface. Penetration is the difference between the depth image (map) value h_{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 h_{dep} is scaled by α in order to convert the depth code value to the physical units of the haptic device workspace.

B. Dynamic Force

One of the main novelties of this chapter is introducing the concept of dynamic force experienced by a user due to the 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 access to camera calibration parameters is limited, 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, it is assumed that the object has a constant acceleration from one frame to another and calculate its acceleration using

$$\vec{a}_{object} = \frac{\vec{\Delta} v}{\Delta t} = \frac{25\vec{v}_{flow}}{10^{-s}} \quad \frac{pixel}{sec^2} \tag{5}$$

where $\vec{v}_{flow}$ is velocity of the object found by the optical flow algorithm (defined in Section III-B) and $\vec{a}_{object}$ is the acceleration of the video object. Note that, $\vec{v}_{flow}$ is scaled by 25 assuming that the video runs at 25 fps and object acceleration is calculated for each haptic loop, hence Δt is taken as 1 msec. Then, the dynamic force experienced by the user can be calculated as

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

where m_{object} and m_{cursor} are mass values assigned to the video object and cursor, respectively, as discussed in Section II-C, $\vec{a}_{cursor}$ is the acceleration of the cursor, and β is a constant to convert the object acceleration from units of image workspace to that of the haptic workspace.

C. Implementation of Depth-Image Based Haptic Rendering

Another novelty of this chapter 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. It is assumed 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. The x-y coordinates of the HIP position $(x_{haptic}(t), y_{haptic}(t), z_{haptic}(t))$ in the physical workspace are mapped to $(x_{img}(t), y_{img}(t))$ on the image plane using the following linear relations:

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

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

where coefficients

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

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

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 height of the image, respectively. 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

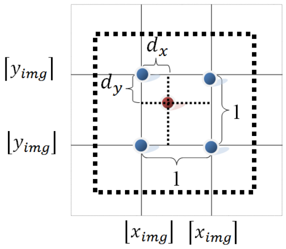


Fig. 1: Bilinear interpolation of the scene/object depth at the HIP position $(x_{img}(t), y_{img}(t))$ on the image plane, shown by the red dot, using the depth map values at the neighboring image pixels, shown by the blue dots.

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 $(x_{img}(t), y_{img}(t))$ on the image plane, which is in general a non-integer value, bilinear interpolation are employed from the depth values at the neighboring pixels as depicted in Fig. 1.

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

$$d_x = |x_{img} - \lfloor x_{img} \rfloor| \quad d_y = |y_{img} - \lceil y_{img} \rceil| \tag{11}$$

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

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

The penetration, which was defined in (4), can then be calculated as the difference between 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 along the z-axis in the haptic workspace.

Selection of Mass for Video Object and HIP: Because in general there is no access to the physical video object, the selection of proper mass values for the HIP and video object should be addressed carefully considering applicable physical laws. Because the velocity of a video object cannot be altered as a result of the user interaction, it is assumed 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 user interaction. This is in analogy to a fly interacting with a horse as mentioned in the Introduction section.

In an active interaction, the force experienced by the user should vary according to how tight the user grips the HIP, which should be reflected into the mass of the HIP. However, in our passive dynamic interaction scenario, the effect of the grip of the user is neglected in accordance to the horse-fly analogy. Therefore, the apparent mass of the haptic device at the tip is set to 0.045 kg, as measured by the manufacturer of the device and stated in the data specifications sheet (Sensable Tech, 2010). Then, the mass of the video object is set equal to at least an order of magnitude larger than this value, such that $m_{object} \gg m_{cursor}$, again in accordance to the horse-fly analogy.

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

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

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

Then, the acceleration can be computed similarly as the difference of HIP velocity in time.

The object acceleration are converted to the units of mm/sec^2 using the inverse of linear mapping defined in (7), and then $\vec{F}_{dynamic}$ is calculated using (6). The sum of $\vec{F}_{dynamic}$ and $\vec{F}_{static}$ is rendered and displayed to the user through the haptic device. For smooth and stable haptic interaction, all forces are interpolated from the video frame rate to the force update rate of the haptic device.

Temporal Interpolation of Forces : Smooth and stable force rendering requires the haptic display rate to be significantly higher than the video/graphic frame rate (Salisbury et al., 2004). For example, the haptic display rate is typically 1 kHz, while the video/graphics frame rate is 25 Hz. Hence, interpolated forces calculated at the video frame rate to the haptic display rate for effective haptic experience are needed. 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 $\vec{F}^{i}_{total}(x_1, y_1)$, forces between $\vec{F}^{i}_{total}(x_1, y_1)$ and $\vec{F}^{i+1}_{total}(x_1, y_1)$, which is the total force calculated at the cursor position (x_1, y_1) in frame $i+1$, are interpolated in the temporal dimension using cubic interpolation.

III. 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 of 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.

A. Enhancement of 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, the geometrical information of touchable video object would be represented more precisely within the range of the depth-map. To segment the object, both the color and depth images are used. The well-known Canny edge detection algorithm is applied to find the edges of the object of interest (Canny, 1986). The color edge images and edges of depth images are fused, in order to create the object segmentation mask as shown in Fig. 2, where segmentation of the horse from rest of the frame is shown.

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, a Gaussian low pass filter is applied to smooth depth images for a more realistic haptic experience.

B. 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 the z-component of the object acceleration (in addition to x and y components) can also be computed for

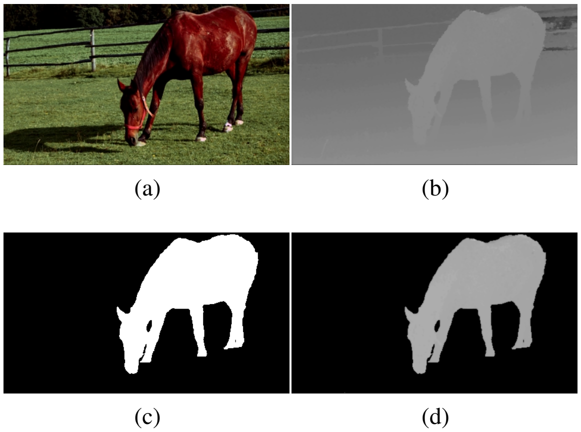


Fig. 2: Illustration of video object segmentation and depth image enhancement: (a) original frame (Mobile3DTV) (b) corresponding depth image, (c) computed object segmentation mask, (d) enhanced depth image.

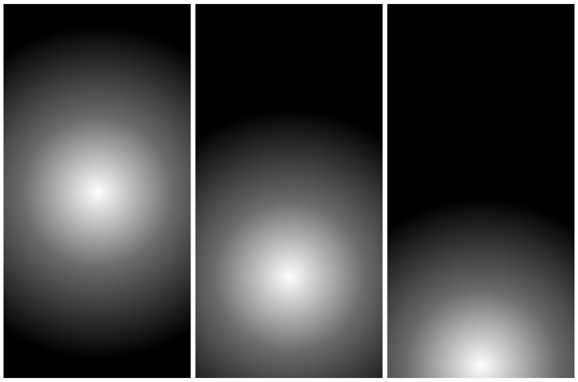


Fig. 3: Sample frames from the synthetic video with $\vec{a}_{object_z} = 0$. Here, a ball moves vertically down in a plane that is perpendicular to the optical axis of the camera.

dynamic force calculation using (6). Estimation of 3D motion and structure from video has been extensively studied in the literature and some widely used methods and procedures exist (Robertson and Cipolla, 2009). 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 to extract 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 (Hartley and Sturm, 1994; Harltey and Zisserman, 2000) or the factorization method proposed by Takosi and Kanade (Tomasi and Kanade, 1992) 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 (Lucas and Kanade, 1981) optical flow algorithm over the segmented textured images (see Fig. 3 for some sample frames of a synthetic video). This algorithm provides us with the x and y components 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.

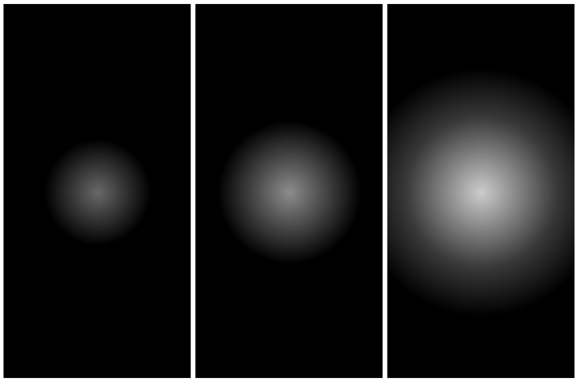


Fig. 4: Sample frames from the synthetic video with $\vec{a}_{object_x} = \vec{a}_{object_y} = 0$. Here a ball moves towards the camera along the optical axis of the camera.

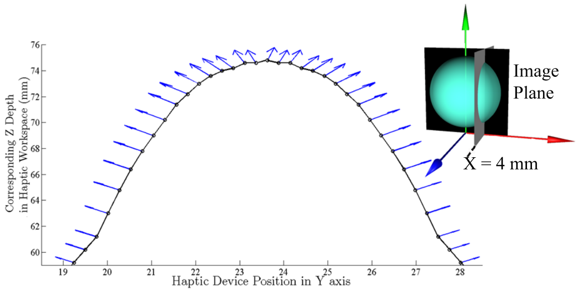


Fig. 5: Static forces rendered for a slice of the ball projected onto the y-z plane in haptic workspace.

Special case $\vec{a}_{object_x} = \vec{a}_{object_y} = 0$: If the motion of the object is purely along the z direction, object motion can be estimated from the variation of the depth observed from the depth map images (see Fig. 4 for some sample frames of a synthetic video). Since it is assumed that 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.

IV. CASE STUDIES AND RESULTS

The proposed concepts are tested using the Omni haptic device (by Sensable Technologies) having 6 degrees of freedom (dof) sensing and 3-dof force feedback. In our systeme, the user interacts with video passively through the HIP. Both static and dynamic forces are experienced while interacting with an object moving in the video. The video can be also paused at a particular frame to feel the geometry of the video object alone (i.e., the static force).

In order to demonstrate the passive dynamic haptic interaction concept in a controlled experiment, two synthetic video clips of a bouncing ball are generated. In the first video clip, shown in Fig. 3, the ball moves vertically up and down perpendicular to the camera, while in the second one, shown in Fig. 4, it moves back and forth towards the camera along the z axis. Moreover some results on a real "horse" video with associated depth data are provided (Mobile3DTV).

A. Simulation Results

Video Paused - Static force only: A slice of the ball on the y-z plane at a particular frame is shown in Fig. 5. The depth is zero at the reference plane (background) and increases as the object approaches the camera along the +z axis. The rendered static forces are depicted using the blue arrows while the black curve represents the depth values of the ball at the specified y positions. Note that the HIP moves along the y axis only, hence only the y and z axes are shown in Fig. 5.

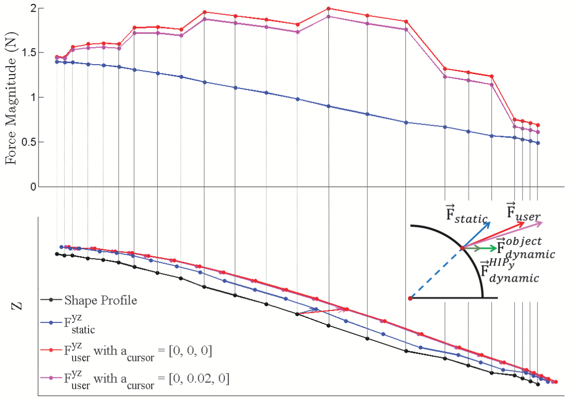


Fig. 6: The profile of the force vectors rendered for a half slice of the ball a) when only the static forces exist (blue curve), b) when the cursor is not moving but the object has an acceleration along the +y direction (red curve) and c) when the cursor and ball accelerates along the +y direction (purple curve). The corresponding force magnitudes for each HIP position are shown in the upper plot.

Case 1: The ball moves perpendicular to the camera axis:

1. The HIP is stationary ($||a_{cursor}|| = 0$): The ball can move along the x axis, y axis, or both. In our example (see Fig. 6), the ball accelerates along the +y direction and the HIP is stationary. Hence the total force experienced by the user, $\vec{F}_{user}$, shown as red, becomes larger than the static force, $\vec{F}_{static}$ alone, shown as blue. The x and z components of the total force are due to the geometry of the video object only, while the y component is due to both the acceleration and the geometry of the object. Note that, since the acceleration of the object is modified intentionally between the frames, the effect of dynamic force is changing.

2. The HIP has a constant acceleration (a_{cursor} is constant): The effect of relative motion of the video object with respect to the HIP motion is also shown in Fig. 6. The HIP and the object have a constant acceleration along the +y direction. Hence, the total force is perturbed slightly towards the y direction (see the purple curve in Fig. 6). Note that, the acceleration of the HIP is in the same direction as the video object and hence the magnitude of the force experienced by the viewers in this case (purple curve) is less than what she/he feels when the HIP is stationary (red curve).

Case 2: The ball moves towards the camera:

1. The HIP is stationary ($||a_{cursor}|| = 0$): It is assumed that the object is rigid and its structure is not changing as it moves towards the camera. Hence, the changes in the size of the object and depth values at each pixel are due to the motion of the object. Since the object moves towards the camera, there is a dynamic force along the +z direction. The effect of this dynamic force is shown by the red curve in Fig. 7. The blue and red curves again represent the static force and total force including the dynamic force, respectively.

2. The HIP has a constant acceleration (a_{cursor} is constant): The effect of the relative motion of the ball with respect to the HIP when the object moves towards the camera is shown as a purple curve in Fig. 7 The HIP has a constant acceleration, having component along the y direction. Since the force due to the acceleration of the ball is in the z direction only, the y component of the total force is due to both the acceleration of the HIP and the geometry of the video object.

B. Results with a Real Video Clip

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

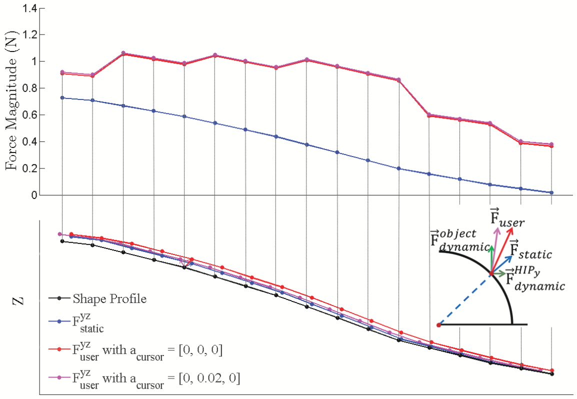
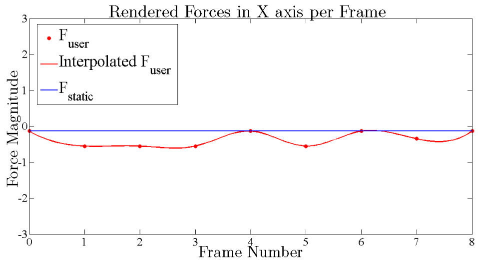
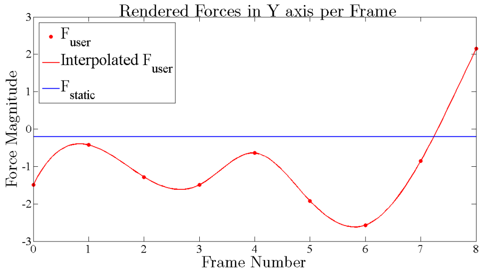


Fig. 7: The profile of the force vectors rendered for a half slice of the ball a) when only the static forces exist (blue curve), b) when the cursor is not moving but the object has an acceleration along the +z direction (red curve) and c) when the object accelerates along the +z direction and the cursor accelerates along the +y direction (purple curve).The corresponding force magnitudes for each HIP position are shown in the upper plot.

Fig. 8: Sample frames from a real video clip (Mobile3DTV). Here the horse slowly raises his head.



(a)



(b)

Fig. 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 of 1kHz.

clip using the motion estimation techniques discussed above. The variation in the static and the total forces in the x and y axes as a function of the selected frames are shown in Fig. 9a and 9b, respectively. In rendering the dynamic forces, the cursor is virtually attached to a fixed spot on the horse's head (i.e., it is stuck to the same spot on the horse as the head moves), hence the static force is constant since the geometry does not change; but the dynamic force varies due to the acceleration of the horse's head. The red curve in each plot represents the total force due to the "sticking" effect and the acceleration of horse head at the HIP position.

The user can also pause the video and feel the static force alone 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.

V. CONCLUSION

This chapter introduces the concept of passive dynamic haptic interaction with a video and describes a method for rendering dynamic forces due to relative motion of the video object. Earlier studies on this subject have focused on the rendering of only static forces due to the geometry of video objects and neglected the dynamic effects. However, in passive haptic interactions with videos involving 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 due to the inertia of the video object (e.g., the ball, bullet, and jet plane) to the user is important for a more immersive experience. As shown in Fig. 6, depending on the mass and the acceleration of the video object with respect to those of the HIP, the contribution of the dynamic force to the total force can be significant (e.g. compare the separation distances of the red and blue colored force profiles from the black colored depth profile in Fig. 6) and may alter the user experience. Note that the masses of the video object and the HIP are set to ensure that the user is a passive observer; i.e., the momentum transferred to the video object from the user (through the HIP) is minimal and does not cause a change in its state. In other words, the user cannot change the state of the object in the video by pushing or pulling it via the HIP. Since the mass of the video 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. Experimental results demonstrate that adding dynamic forces results in an observable difference in the total force which leads to more realistic haptic experience.

In conclusion, ”immersive” and ”interactive are frequently used adjectives in applications such as virtual reality, computer games, and human-computer interaction. However, their use with real media (both 2D and 3D media) entertainment, broadcast, and communications services is new and in its infancy, since stimuli represented by modalities beyond 3D video and spatial audio, such as touch, heat, smell, etc., need also be captured, authored, transmitted, and rendered and displayed at the receiving side to even create a fully immersive passive haptic media experience. Furthermore, new modalities of interaction with media, such as active haptic interaction have not yet been considered. Novel ideas and approaches that would enable haptic experience and haptic interaction with real media open new horizons that will significantly impact next generation media technologies.

ACKNOWLEDGMENT

A. Murat Tekalp acknowledges support from Turkish Academy of Sciences (TUBA).

REFERENCES

Cagatay Basdogan and Mandayam A. Srinivasan. Haptic rendering in virtual environments. In K.M. Stanney, editor, *Handbook of Virtual Environments: Design, Implementation, and Applications*, pages 117–134. Lawrence Erlbaum Associates, New Jersey, 2001.

Cagatay Basdogan, Chih-Hao Ho, Mandayam A. Srinivasan, and Mel Slater. An experimental study on the role of touch in shared virtual environments. *ACM Transaction Computer-Human Interaction*, 7(4):443–460, 2000.

J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, June 1986. ISSN 0162-8828.

Jongeun Cha, Seung man Kim, Ian Oakley, Jeha Ryu, and Kwan Lee. Haptic interaction with depth video media. In *Advances in Multimedia Information Processing - PCM 2005*, pages 420–430. Springer Berlin / Heidelberg, 2005.

Jongeun Cha, Sung-Yeol Kim, Yo-Sung Ho, and Jeha Ryu. 3D video player system with haptic interaction based on depth image-based representation. *IEEE Transaction Consumer Electronics*, 52(2):477–484, 2006.

Jongeun Cha, Yongwon Seo, Yeongmi Kim, and Jeha Ryu. An authoring/editing framework for haptic broadcasting: Passive haptic interactions using mpeg-4 bifs. In *Proceedings of the Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, WHC '07, pages 274–279. IEEE Computer Society, 2007.

Derek Gaw, Daniel Morris, and J. Kenneth Salisbury. Haptically annotated movies: Reaching out and touching the silver screen. In *14th Symposium on Haptic Interfaces for Virtual Enviroment and Teleoperator Systems*, pages 287–288, 2006.

Andrew Harltey and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000. ISBN 0521623049.

Richard I. Hartley and Peter Sturm. Triangulation. In *American Image Understanding Workshop*, pages 957–966, 1994.

Chih-Hao Ho, Cagatay Basdogan, and Mandayam A. Srinivasan. Efficient point-based rendering techniques for haptic display of virtual objects. *Presence: Teleoper. Virtual Environ.*, 8(5):477–491, October 1999. ISSN 1054-7460.

Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings DARPA IU Workshop,*, pages 121–130, 1981.

Nadia Magnenat-Thalmann and Ugo Bonanni. Haptics in virtual reality and multimedia. *IEEE MultiMedia*, 13(3):6–11, 2006.

Mobile3DTV. Mobile3dtv: Video plus depth database.

Karsten Müller, Aljoscha Smolic, Kristina Dix, Peter Kauff, and Thomas Wiegand. Reliability-based generation and view synthesis in layered depth video. In *Proceedings of IEEE Int'l Workshop on Multimedia Signal Processing (MMSP 2008)*, pages 34–39, 2008.

S. O'Modhrain and I. Oakley. Touch tv: Adding feeling to broadcast media. *In Proc. European Conference Interactive Television: form Viewers to Actors*, pages 41–47, December 2003.

S. O'Modhrain and I. Oakley. Adding interactivity: Active touch in broadcast media. In *Proceedings of 12th Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems (HAP TICS'04) March 2004, Chicago, IL, USA, Proceedings*, pages 293–294. IEEE Computer Society, 2004. ISBN 0-7695-2112-6.

Shahzad Rasool and Alexei Sourin. Towards tangible images and video in cyberworlds-function-based approach. In *CW*, pages 92–96, 2010.

Duncan P. Robertson and Roberto Cipolla. *Structure from Motion*, chapter 13. John Wiley, 2009.

Fredrik Ryden, Howard Jay Chizeck, Sina Nia Kosari, Hawkeye King, and Blake Hannaford. Using kinect and a haptic interface for implementation of real-time virtual fixtures. In *RGB-D: Advanced Reasoning with Depth Cameras Workshop, RSS 2011*, 2011.

J. K. Salisbury, F. Conti, and F. Barbagli. Haptic rendering: Introductory concepts. *IEEE Computer Graphics and Applications*, 24(2):24–32, March 2004.

Inc Sensable Tech. *PHANTOM Omni Datasheet*. Sensable Technologies Inc, 2010.

Aljoscha Smolic and Peter Kauff. Interactive 3-D video representation and coding technologies. In *Proceedings of IEEE Special Issue on Advances in Video Coding and Delivery*, volume 93, pages 98–110, 2005.

Mandayam A. Srinivasan and Cagatay Basdogan. Haptics in virtual environments: taxonomy, research status, and challenges. *Computers & Graphics*, 21(4):393–404, 1997.

Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

Sean P. Walker and J. Kenneth Salisbury. Large haptic topographic maps:marsview and the proxy graph algorithm. In *Proceedings of ACM SIGGRAPH*, pages 83–92, 2003.

Takehiko Yamaguchi, Ayumu Akabane, Jun Murayama, and Makoto Sato. Automatic generation of haptic effect into published 2d graphics. In *Proceedings of EuroHaptics 2006*, 2006.