Augmented Touch without Visual Obtrusion

Francesco I. Cosco
Università della Calabria

Carlos Garre
URJC Madrid

Fabio Bruno
Università della Calabria

Maurizio Muzzupappa
Università della Calabria

Miguel A. Otaduy
URJC Madrid

Figure 1: From left to right: image of a visuo-haptic mixed reality scene where the haptic device produces visual obtrusion of the background; visual removal of the haptic device; background completion based on image-based rendering; and final composite scene.

ABSTRACT

Visuo-haptic mixed reality consists of adding to a real scene the ability to see and touch virtual objects. It requires the use of see-through display technology for visually mixing real and virtual objects, and haptic devices for adding haptic interaction with the virtual objects. However, haptic devices tend to be bulky items that appear in the field of view of the user. In this work, we propose a novel mixed reality paradigm where it is possible to touch and see virtual objects in combination with a real scene, but without visual obtrusion produced by the haptic device. This mixed reality paradigm relies on the following three technical steps: tracking of the haptic device, visual deletion of the device from the real scene, and background completion using image-based models. We have developed a successful proof-of-concept implementation, where a user can touch virtual objects in the context of a real scene.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, Augmented, and Virtual Realities

1 INTRODUCTION

Mixed reality has typically dealt with the visual addition of virtual objects to a real scene. But, in order to achieve a full combination of virtual and real objects, the rest of the sensory modalities must also be capable of perceiving the mixed environment. In this work, we address challenges related to visuo-haptic mixed reality, where a user can *see and touch* virtual objects in combination with real objects in the scene. Among others, visuo-haptic mixed reality has been introduced in medical applications [7], virtual prototyping, e.g., for the automotive industry [13], or digital entertainment [12].

In the typical desktop virtual-reality setup, the user looks at a screen, and visual and haptic stimuli are presented in a de-located manner. However, a mixed reality setup allows the user to perceive visual and haptic stimuli in a collocated manner, i.e., the user can see and touch virtual objects at the same spatial location. Collocation improves the sensory integration of multimodal cues and makes the interaction more natural, but it also comes with technological challenges. The inclusion of haptic interaction in a mixed reality scene requires the use of a haptic actuator, but most haptic actuators are bulky devices that occupy a large space in the visual region of interest, i.e., in the location where the interaction is actually taking place. Therefore, in a collocated visuo-haptic mixed reality setup, the haptic device becomes an obtrusive visual element.

Given this challenge, our main contribution is a novel visuo-haptic mixed reality paradigm. It provides augmented touch, as well as collocated visual and haptic interaction, but without visual obtrusion produced by the haptic device. This mixed reality paradigm allows a user to touch virtual objects that are added to a real background or context scene, without suffering from obtrusive occlusion of this context scene.

We propose an algorithm that makes this visuo-haptic mixed reality paradigm possible. It is based on visual removal of the haptic device from the context scene, together with image-based background completion. Given view and haptic configurations, we first identify the region of the image plane occupied by the haptic device, i.e., the region where the device is producing visual obtrusion. Then, given prerecorded views of the context scene, we substitute the haptic device with a view-dependent image of the background, using image-based rendering techniques. Last, we add the virtual objects, both visually and haptically, to produce the visuo-haptic mixed reality scene.

The importance of unobtrusive haptic interaction has been addressed in the past, and the proposed answers relied on mechanical solutions that placed the haptic actuators far from the region of interest using string-based haptic devices [16], or optical solutions based on retroreflective paint and a head-mounted projector [9]. Here, we propose instead a computational solution, which we believe increases the versatility of visuo-haptic interaction setups.

2 RELATED WORK

Several researchers have addressed the importance of collocating visual and haptic stimuli, as this allows virtual tasks to be carried out from a first-person point of view [5]. Visual and haptic de-location is however quite common, because the construction of a de-located setup is far simpler. Congedo et al. [4] emphasize that, in tasks where the contribution of touch is important, great effort should be undertaken to collocate vision and touch, so that the weight of the non-dominant modality, i.e., touch, is not penalized. Spence et al. [14] summarize crossmodal congruency effects involving vision and haptics.

Visuo-haptic collocation can be achieved in several ways, and the most popular ones include workbenches with stereo projection systems [2, 16], mirror-based projection systems where the virtual image occludes the real scene [15], or head-mounted displays with head and device tracking [1]. Our approach uses see-through head-mounted display technology, and our mixed reality paradigm is in-

IEEE International Symposium on Mixed and Augmented Reality 2009
Science and Technology Proceedings
19 -22 October, Orlando, Florida, USA
978-1-4244-5419-8/09/$25.00 ©2009 IEEE

tegrated in the computational pipeline for rendering the mixed reality images.

As discussed in the introduction, the addition of haptic devices introduces bulky obtrusive objects in an augmented reality scene. One possible solution to visual obtrusion is to use stringed haptic devices, such as the SPIDAR [11]. Stringed haptic devices place the actuators far from the region where manipulation and interaction are actually happening, and transfer force and torque to the end effector using tensed strings. With sufficiently thin strings, the haptic device barely occludes the rest of the scene. Ortega and Coquillart [13] applied this visuo-haptic interaction paradigm in the context of an automotive virtual prototyping application. Moreover, they used as end-effector a geometric prop of the actual tool, and mounted a transparent structure around the prop for adequately attaching the strings. Stringed haptic devices have been integrated in a workbench that provides view-dependent stereo vision [16].

Another possible solution to visual obtrusion is optical camouflage [10], which consists of covering the obtrusive elements with retroreflective paint, and use a projector to render on top of them the desired background image. This approach was proposed by Inami et al. [9] for solving the visual obtrusion produced by haptic devices in mixed reality scenes. Our mixed-reality paradigm can be perhaps interpreted as a computational approach to optical camouflage. In practice, optical camouflage is also closely related to diminished reality [17].

The main computational aspects of our mixed reality paradigm make use of image-based rendering (IBR) techniques. In particular, we follow Buehler's unstructured lumigraph rendering [3], with a strong focus on view-dependent texture mapping [6].

3 Visual Subtraction of the Haptic Device

Given a prerecorded set of images of the context scene, sufficient for accurately producing an image-based model, the visuo-haptic mixed reality paradigm works in the following way: An image of the scene is captured with a camera at run-time; the region of the image occluded by the haptic device is identified; the image is processed to erase the haptic device; the erased region is re-painted using an IBR algorithm and the prerecorded images; virtual objects are composed in the scene; and the final result is displayed to the user. This algorithm works fully on image-space.

Based on the computational approach to the problem, this mixed reality paradigm is best implemented using video see-through head mounted displays. Our demonstration examples were tested on a stereo system, which basically requires running the visual subtraction algorithm on each image separately, but here we will describe the algorithm for one image.

Next, we describe in detail the input data needed by the algorithm, as well as its two main steps: the definition of the occluded region to be re-painted, and the re-painting itself.

3.1 Input Data

The algorithm takes two types of data as input: static data which can be acquired in a preprocessing step, and dynamic data that is acquired at every frame. The static data consists of:

- The intrinsic camera parameters.
- The arrangement of marker geometry in the world.
- A set of images of the background scene, sufficient for creating the image-based model from all possible run-time viewing angles and for all possible run-time device configurations.
- For each image, the extrinsic camera parameters, i.e., the camera pose.
- Some geometry proxies, i.e., world points with an approximately known position.
- The transformation between the local reference system of the haptic device and the global reference system.
- An approximate geometric model of the haptic device.

At run-time, the algorithm needs the following dynamic data:

- The extrinsic camera parameters, i.e., the camera pose.
- The configuration of the haptic device in its local frame.

As a preprocessing, we calibrate the intrinsic camera parameters using Matlab's Calibration Toolbox. And we calibrate the transformation from the global to the local reference system of the haptic device by placing the tip of the end effector at a set of known world positions, and then optimizing for the transformation by solving a least squares problem. At run-time, we estimate the current camera pose using ARToolKit's marker-based tracking. Each marker is composed of a black square with some shape inside for defining its orientation. It is sufficient if one marker is visible at each time. Note that the visuo-haptic mixed reality paradigm may as well work with other calibration and tracking algorithms .

3.2 Occluded Region in Screen-Space

Given the approximate geometric model of the haptic device, the transformation from the global to the local reference frame of the device, and the current configuration of the device (e.g., joint angles), we can place the approximate geometric model in the world. Fig. 2-left shows the approximate geometric model blended on top of the actual haptic device in one of our demonstration examples.

The extrinsic camera parameters for the current frame complete the definition of the modelview and projection matrices, and we can render the approximate geometry of the device onto screen-space. When performing this rendering step, we activate a mask in the stencil buffer for the rendered pixels. This mask defines the occluded region in screen-space (shown in green in Fig. 2). The background completion algorithm to be described next needs to be executed inside the masked occluded region.

There are other possible options for identifying the occluded region in screen-space. One of them is to paint the device in a characteristic color and use it as a chroma. This approach would be more robust under possible calibration problems, but it could in turn suffer from color and/or lighting issues. In our experiments, the tracking robustness of the haptic device appeared to be sufficient, and we found our approach based on mask rendering sufficiently accurate.

3.3 IBR-Based Background Completion

Our algorithm for image-based background completion exploits the unstructured lumigraph approach [3]. The scene light field is considered to be known for a set of irregularly distributed rays, and we use view-dependent texture mapping [6] to interpolate light field data from those rays. More precisely, we compute camera blend weights for a discrete set of vertices on the occluded region, we mesh the vertices and define a camera blend field over the occluded region, and compute the final image by blending the results of view-dependent texture mapping. We use two types of input data in a combined manner: a set of prerecorded images of the background scene together with their associated camera positions, and a very rough geometric approximation of the background.

We describe next the full IBR algorithm in four steps: (i) description of the geometric proxy, (ii) sampling of the occluded region with known camera rays, (iii) meshing of the occluded region, and (iv) efficient view-dependent texture mapping.

3.3.1 Geometric Proxy of the Background

As a preprocess, we construct a very rough geometric proxy of the background. This proxy consists of planar desks, planar walls, and an average-depth plane that approximates other irregular surfaces. For example, in our test setting in Fig. 2, we use a plane for the desk, another plane for the left wall, and a third plane that approximates the window area on the right.

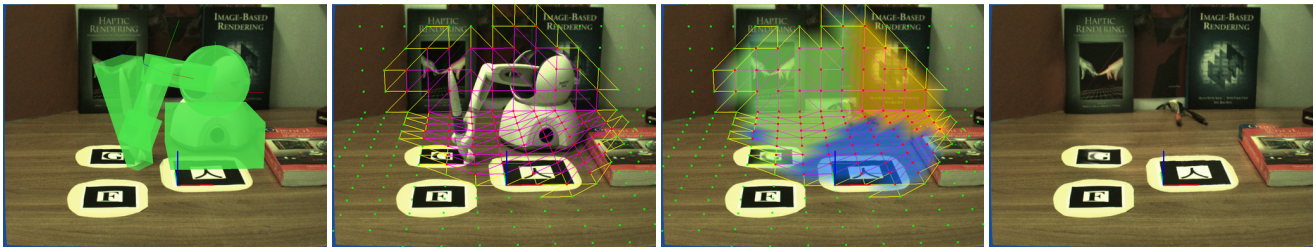

Figure 2: IBR-based background completion algorithm. From left to right: mask blended over the region occluded by the device; triangle mesh formed by geometry and camera vertices; camera blend field; and final IBR result.

We sample regularly the proxy planes. These samples, after being projected onto the image plane, will constitute the vertices for view-dependent texture mapping, as described next. The geometric proxies serve as depth estimates for defining the homography transformation of view-dependent texture mapping. It would also be possible to employ other geometric proxies, perhaps precomputed from the prerecorded images, or even run-time feature extraction with depth estimation.

3.3.2 Sampling the Occluded Region

Following unstructured lumigraph rendering, we sample a camera blend field in the occluded region with two types of vertices:

Geometry vertices. They are the projections onto the image plane of the points sampled on the geometric proxies. Since their depth is (approximately) known, they can be used for defining a homography transformation between the output camera and the camera pose of one of the prerecorded images.

Camera vertices. They are the projections of camera centers of the prerecorded images onto the image plane. A camera vertex is discarded if the line of projection does not fit in its field of view. Camera vertices exploit epipole consistency [3]. A camera vertex needs a depth estimate for computing homography transformations. We obtain this depth estimate by interpolating the depths of nearby geometry vertices in image-space.

3.3.3 Meshing and Camera Blend Field

Each vertex in the occluded region is associated to one of the input cameras. For a camera vertex, the camera is trivially selected as the one that produced the vertex itself. For geometry vertices, we select the camera with smallest angle w.r.t. the line of projection.

At each vertex, we select the blending weight of the associated camera as one, and the weights of all other cameras as zero. We define the blending field in the rest of the occluded region by meshing the vertices and interpolating the camera blending weights inside each triangle. We construct the mesh through Delaunay triangulation of the vertices.

In practice, we set camera and geometry vertices in a portion of the viewport that surrounds the occluded region, as shown in Fig. 2. We also add an extra layer of triangles where we blend the original image and the result of IBR, in order to alleviate possible issues due to calibration inaccuracies. Fig. 2 also shows an example of camera blend field.

3.3.4 View-Dependent Texture Mapping

Given a triangle of the mesh, with vertices $\{a,b,c\}$, we compute the output image inside the triangle by warping and blending the prerecorded background images. In particular, for each vertex a we compute a homography transformation based on the depth associated to the vertex, the pose of the associated input camera, and the pose of the output camera. Given this transformation, we compute, for all three vertices, texture coordinates in the input image associated to vertex a. The interpolation capabilities of graphics hardware compute directly texture coordinates inside the triangle, and define the mapping from the input image to the output. Given the camera blend weights for the three vertices of the triangle, we again exploit hardware interpolation to define blend weights at arbitrary pixels inside the triangle, and blend the warped input images defined by the three vertices.

We have implemented view-dependent texture mapping on the GPU, with simple shaders for blending. The shaders have been implemented on NVIDIA's Cg shader language.

4 Results

We have evaluated the mixed reality paradigm on scenes with combined visuo-haptic feedback. The algorithm for visually removing the haptic device from the mixed scene is combined with state-of-the-art methods for haptic rendering of the interaction between a virtual tool and other deformable and rigid objects, both virtual and real. Please see the accompanying video for the dynamic results of the examples discussed here. We have used the following framework for our tests: A PHANToM Omni haptic device from SensAble Technologies, a head-mounted display composed of a Z800 3D visor from eMagin with two external flea2-08S2C cameras from Point Grey, and a dual 2.13-GHz processor PC with 2-Gb of RAM. We recorded 121 images for the background model, and sampled its geometric proxy with 597 vertices. Under these settings, the execution of the visual subtraction of the haptic device does not add a remarkable cost to the system. The rendering frame rate is of 30 fps for the standard AR system, and of 25 fps after the addition of our algorithm. The physically-based simulation takes the frame rate down to 12 fps in complex contact configurations, but we have used the multirate haptic rendering algorithm of [8] to maintain stable interaction.

Our first experiments involved only visual deletion of the device from the scene, as shown in Fig. 2. Possible run-time tracking and device calibration errors required a slight scaling of the virtual model of the haptic device. However, this scaling was minimal, as shown in Fig. 2. The quality of the IBR implementation was not optimal because the books were not part of the geometric proxies, leading to possible ghosting effects, and the illumination varied from the capture session to the time when some of the examples were computed.

Next, we incorporated dynamic virtual objects to the scene. In the examples we have used as virtual tool a wrench, shown in Fig. 3, which visually replaces the handle of the haptic device. Note that the wrench and the handle may appear slightly separated because our haptic rendering algorithm places a viscoelastic coupling between their dynamic models, and the wrench is constrained by contact with the virtual objects, while the physical device is not. The physically-based model of the virtual environment also incorporates virtual models (not depicted in the figure) of the desk and some of the books, which allows the computation of contact with real objects. Although not included in our proof-of-concept implementation, these virtual models of real objects would also allow casting shadows of real objects on virtual objects and vice versa. With the mixed reality paradigm based on image-based models, it would also be possible to include in the mixed reality scene a different background than the one used during the interaction.

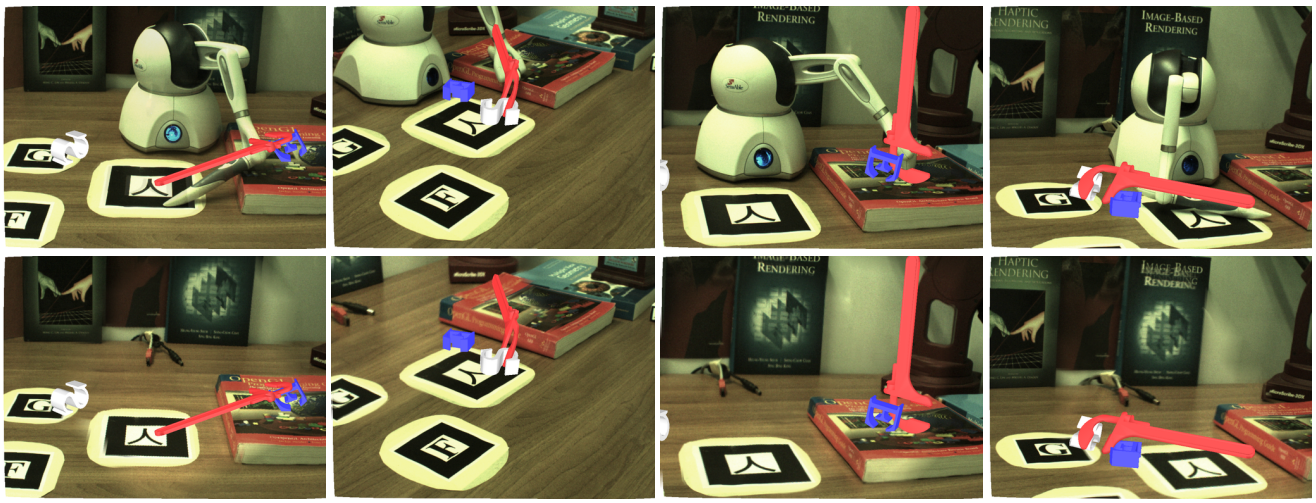

Figure 3: Images comparing visuo-haptic mixed reality with visual obtrusion Vs. our mixed reality paradigm where the haptic device is *removed* from the scene. The examples show contact of a virtual wrench with deformable virtual letters and real objects such as the desk and books.

5 Discussion and Future Work

In this paper, we have presented a novel mixed reality paradigm that enables compelling visuo-haptic augmented reality without the visual obtrusion introduced by haptic devices. The key element of the mixed reality paradigm is a computational approach for camouflage of the haptic device, using an image-based model of the context scene. We believe that this mixed reality paradigm can increase the versatility and outreach of visuo-haptic mixed reality.

Our proof-of-concept implementation lacks several advances in mixed reality and image-based rendering that, although orthogonal to our work, could increase the quality and versatility of the results. Some of these include more robust and general tracking, image-based rendering with refined depth estimation and geometric proxies, denser input acquisition together with advanced closest-ray-search data structures, full handling of occlusion between virtual and real objects using, e.g., foreground detection, and consistent illumination and shadowing of the virtual objects.

The mixed reality paradigm could also be enhanced to support dynamic backgrounds. This would require run-time acquisition of the background, e.g., with additional cameras. In principle the inclusion of a passive dynamic background appears feasible, as long as the capture includes a rough depth estimation. It is more difficult, however, to include an active dynamic background that interacts with the virtual objects, as this requires virtual replicas of the dynamic background objects in the physically-based model of the environment.

Together with the handling of dynamic backgrounds, the foremost limitation of the mixed reality paradigm is the correct display of the user's own hand. In our proof-of-concept implementation we have simply opted for not displaying the hand, but full first-person interaction would require its inclusion in the final composite scene. At the moment we are investigating a fully image-based solution, based on the detection of the hand in the input image and its composition with the virtual tool.

Acknowledgements

We would like to thank the anonymous reviewers, David Miraut and the rest of the GMRV group at URJC. This work was funded in part by URJC - Comunidad de Madrid (project CCG08-URJC/DPI-3647) and the Spanish Ministry of Science and Innovation (project TIN2009-07942).

References

[1] G. Bianchi, C. Jung, B. Knoerlein, G. Szekely, and M. Harders. High-fidelity visuo-haptic interaction with virtual objects in multi-modal AR systems. *Proc. of ISMAR*, 2006.

[2] J. D. Brederson, M. Iktis, C. R. Johnson, and C. D. Hansen. The visual haptic workbench. *Proc. of PHANToM User Group Workshop*, 2008.

[3] C. Buehler, M. Bosse, L. McMillan, S. J. Gortler, and M. F. Cohen. Unstructured lumigraph rendering. *Proc. of ACM SIGGRAPH*, 2001.

[4] M. Congedo, A. Lécuyer, and E. Gentaz. The influence of spatial de-location on perceptual integration of vision and touch. *Presence: Teleoperators and Virtual Environments*, 15(3), 2006.

[5] S. Coquillart. A first-person visuo-haptic environment. *Proc. of HCII*, 2007.

[6] P. Debevec, C. Taylor, and J. Malik. Modeling and rendering architecture from photographs. *Proc. of ACM SIGGRAPH*, 1996.

[7] J. Fornaro, M. Harders, M. Keel, B. Marincek, O. Trentz, G. Szekely, and T. Frauenfelder. Interactive visuo-haptic surgical planning tool for pelvic and acetabular fractures. *Proc. of MMVR*, 2008.

[8] C. Garre and M. A. Otaduy. Haptic rendering of complex deformations through handle-space force linearization. In *Proc. of World Haptics Conference*, mar 2009.

[9] M. Inami, N. Kawakami, D. Sekiguchi, Y. Yanagida, T. Maeda, and S. Tachi. Visuo-haptic display using head-mounted projector. *Proc. of IEEE Virtual Reality Conference*, 2000.

[10] M. Inami, N. Kawakami, and S. Tachi. Optical camouflage using retro-reflective projection technology. *Proc. of ISMAR*, 2003.

[11] M. Ishii and M. Sato. A 3D spatial interface device using tensed strings. *Presence*, 3(1), 1994.

[12] B. Knoerlein, G. Szekely, and M. Harders. Visuo-haptic collaborative augmented reality ping-pong. *Proc. of Conference on Advances in Computer Entertainment Technology*, 2007.

[13] M. Ortega and S. Coquillart. Prop-based haptic interaction with co-location and immersion: An automotive application. *Workshop HAVE*, 2005.

[14] C. Spence, F. Pavani, A. Maravita, and N. P. Holmes. Multi-sensory interactions. In M. C. Lin and M. A. Otaduy, editors, *Haptic Rendering: Foundations, Algorithms and Applications*, chapter 2. AK Peters, 2008.

[15] D. Stevenson, K. Smith, J. Mclaughlin, C. Gunn, J. Veldkamp, and M. Dixon. Haptic workbench: A multisensory virtual environment. *Proc. of SPIE*, 1999.

[16] N. Tarrin, S. Coquillart, S. Hasegawa, L. Bouguila, and M. Sato. The stringed haptic workbench: A new haptic workbench solution. *Proc. of Eurographics*, 2003.

[17] S. Zokai, J. Esteve, Y. Genc, and N. Navab. Multiview paraperspective projection model for diminished reality. *Proc. of ISMAR*, 2003.