Visuo-Haptic Mixed Reality with Unobstructed Tool-Hand Integration

Francesco Cosco, Carlos Garre, Fabio Bruno, *Member*, *IEEE*,
Maurizio Muzzupappa, and Miguel A. Otaduy, *Member*, *IEEE*

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. Unfortunately, the use of commodity haptic devices poses obstruction and misalignment issues that complicate the correct integration of a virtual tool and the user's real hand in the mixed reality scene. 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, using commodity haptic devices, and with a visually consistent integration of the user's hand and the virtual tool. We discuss the visual obstruction and misalignment issues introduced by commodity haptic devices, and then propose a solution that relies on four simple technical steps: color-based segmentation of the hand, tracking-based segmentation of the haptic device, background repainting using image-based models, and misalignment-free compositing of the user's hand. We have developed a successful proof-of-concept implementation, where a user can touch virtual objects and interact with them in the context of a real scene, and we have evaluated the impact on user performance of obstruction and misalignment correction.

Index Terms—Mixed reality, visuo-haptic mixed reality, occlusion handling, haptic interfaces, image-based rendering

1 Introduction

Mixed reality (MR) has typically dealt with the visual addition of virtual objects to a real scene. But, in order to fully experience the mixed environment, the integration of virtual and real objects must be extended to the rest of the sensory modalities. In this work, we address challenges related to Visuo-Haptic Mixed Reality (VHMR), where a user can *see and touch* dynamic virtual objects in combination with static real objects in the scene. Among others, VHMR has been introduced in medical applications [1], virtual prototyping, e.g., for the automotive industry [2], or digital entertainment [3].

In the typical desktop VR setup, the user looks at a screen, and visual and haptic stimuli are presented in a delocated manner. However, a MR setup allows the user to perceive visual and kinesthetic stimuli in a colocated manner, i.e., the user can see and touch virtual objects at the same spatial location. Visuo-haptic colocation of the user's hand and a virtual tool improves the sensory integration of multimodal cues and makes the interaction more natural, but it also comes with technological challenges. In this work, we identify, evaluate, and address two of these challenges, with the common goal of improving the naturalness of the perceptual experience and the sense of presence.

First, the inclusion of haptic interaction in a MR 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 colocated VHMR setup, the haptic device becomes an obstructive visual element. The importance of unobstructive 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 [4], or optical solutions based on retro-reflective paint and a head-mounted projector [5]. We propose instead a computational camouflage solution that increases the versatility of visuo-haptic interaction setups. It is based on image-space removal of the haptic device from the context scene, together with an Image-Based Rendering (IBR) strategy for background completion.

Second, visual display of virtual tool manipulation requires visual compositing of the virtual tool and the user's hand. One issue to be addressed is correct handling of occlusion between virtual and real objects, which has already received important attention in MR research, as it largely contributes to the feeling that virtual objects truly exist in the real world [6]. But commodity haptic devices pose an additional and so far unexplored issue to visual composition. They suffer from mechanical limitations that may restrain the ability to render the virtual objects up to their full dynamic range. In particular, when a virtual tool is constrained by a wall, a commodity haptic device cannot prevent the user from pushing inside the wall, producing an undesired misalignment of the virtual tool and the user's hand. We explore the influence of virtual tool misalignment on user

- F. Cosco, F. Bruno, and M. Muzzupappa are with the Department of Mechanical Engineering, University of Calabria, Ponte P. Bucci, 46C, 87036 Rende(CS), Italy.
E-mail: {francesco.cosco, f.bruno, muzzupappa}@unical.it.
- C. Garre and M.A. Otaduy are with the Department of Computer Science, Universidad Rey Juan Carlos, Edf. Ampliacio´n Rectorado, D-0052 c/ Tulipán, s/n, E-28933 Mo´stoles, Spain.
E-mail: {carlos.garre, miguel.otaduy}@urjc.es.

Manuscript received 12 May 2011; revised 30 Dec. 2011; accepted 22 Mar. 2012; published online 10 Apr. 2012.
Recommended for acceptance by D. Schmalstieg.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2011-05-0107.
Digital Object Identifier no. 10.1109/TVCG.2012.107.

1077-2626/13/$31.00 © 2013 IEEE Published by the IEEE Computer Society

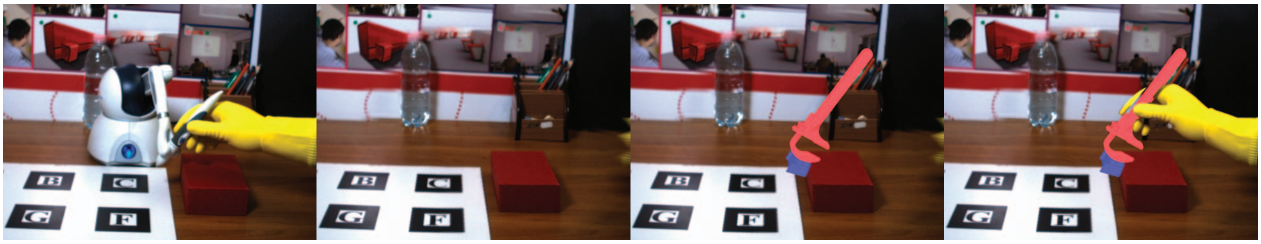

Fig. 1. A visual overview of the pipeline of our proposed VHMR display. From left to right: (a) View of the scene as seen in the real world; (b) Virtual view after image-based removal of the haptic device and the user's hand, followed by IBR-based background completion; (c) Composition of virtual objects and virtual tool in contact; (d) Final view, after image-based composition of the user's hand, free of tool-hand misalignments. Notice that the final position of the hand is not the same as in the real-world view.

performance, and propose a simple technique that compensates for misalignment by redrawing the user's hand with an artificial displacement.

After a discussion of related work in Section 2, we overview our VHMR display paradigm (depicted in Fig. 1), discuss hardware components, and outline the computational pipeline in Section 3. One interesting side effect of the MR display is that it can be leveraged on simple interactive procedures for haptic device calibration and scene preprocessing, as described in Section 4.

In the following sections, we describe our methods for unobstructed tool-hand integration in the VHMR display. In Section 5, we present image-space labeling of semantically distinct visual components. This step entails haptic-device segmentation based on tracking, and a color detection method for hand segmentation. Then, in Section 6, we describe how we avoid visual obstruction due to the haptic device, thanks to an IBR-based method for background repainting and computational camouflage. And in Section 7, we describe how we fix tool-hand misalignments due to mechanical limitations of commodity haptic devices, thanks to consistent visual compositing of the user's hand.

In Section 8, we discuss performance and accuracy of various components of the VHMR display algorithm, but, most importantly, we evaluate the impact of device camouflage and misalignment correction on the overall VHMR experience. We have designed VHMR experiments to measure user performance in terms of task completion time and stiffness discrimination, and we have found that both visual obstruction and tool-hand misalignment have a negative effect on user performance. Our computational solutions improve user performance significantly, therefore enhancing the overall VHMR experience.

2 Related Work

Our display paradigm falls in the category of MR, as its goal is to provide seamless display of mixed real and virtual objects. However, many of the technical issues are shared with augmented reality (AR), where properties of real objects are modified. For a taxonomy of mixed reality visual displays, please see the work of Milgram and Kishino [7], and for the differences between haptic MR and haptic AR, please see the works of Bayart and Kheddar [8] or Jeon and Choi [9]. We classify here the discussion of related work into three categories: occlusion handling in visual MR, colocated visuo-haptic displays, and solutions to visual obstruction due to haptic devices.

2.1 Occlusion Handling in Visual MR

Occlusion handling is perhaps the foremost technical issue in visual MR. As noted early on by Sekuler and Palmer [6], realistic occlusion between real and virtual objects enhances a user's perception that virtual objects truly exist in the real world, whereas an incorrect occlusion handling confuses the viewer. There are two large classes of methods for handling occlusion in the visual compositing of a MR scene: those that exploit prior geometric models of the real objects, and those that do not assume previous knowledge about the geometry. In model-based techniques, model registration becomes the main computational task, while non-model-based techniques must rely on image processing and depth estimation techniques [10], [11].

Furhmann et al. [12] extended model-based occlusion handling to manage occlusions caused by the user's body parts. They utilized an avatar of the user, modeled as a kinematic chain of articulated solids. The avatar was continuously tracked to mimic user motions, and then used to compute occlusions. Unfortunately, its registration is a time-consuming task, not suitable for real-time applications. Lok [13] developed a method relying on image-based modeling techniques, capable of incorporating real objects into a virtual environment, in order to avoid the predefinition of the geometric model of the real scene.

Non-model-based techniques typically aim at managing occlusion between real and virtual objects in dynamic scenes. The most investigated solution involves depth estimation of the scene and, due to the high computational cost of depth estimation, most approaches have targeted the optimization of performance with real-time applicability in mind. In [14], the processing time was optimized by restricting an edge-based stereo matching algorithm to the screen-region covered by virtual artifacts. Schmidt et al. [15] presented an efficient method for computing dense stereo matching. However, recently very promising solutions have been obtained by adopting special hardware and/or by designing highly parallelizable algorithms on graphics processors. Fischer et al. [11] proposed a prototype system that integrates time-of-flight range data to compute depth maps. In [16], the authors improved computational performance adopting a dense-stereo matching approach, executed on dedicated hardware. Lu and Smith [17] developed a GPU-based dense stereo matching algorithm 20 times faster than the equivalent CPU-based optimized algorithm.

Ventura and Hollerer [18] designed a technique aimed at the particular but very common case where a real occluding object, e.g., the user's hand, lies between the user's viewpoint (the camera) and the virtual objects. Given this assumption, they described a method that combines stereo-matching depth reconstruction with a color-based statistical refinement for noise reduction. In our implementation, we adopt a similar strategy, presenting a color-detection method that correctly manages occlusions of virtual objects and the user's hands. However, we manage to avoid depth reconstruction, with the subsequent benefits for real-time performance. The approach of Berger [19] also avoided depth reconstruction, as it was based on image-space contour detection and labeling of the relative depth of contours with respect to the virtual objects.

In [20], Lee and Park presented a method for mixed prototyping that uses physical props of virtual objects to determine depth relationships. They painted the physical prop with a special color, applied a chroma-key filter to the image, and composed the virtual object in chroma-key regions. Our approach does not rely on physical props.

2.2 Colocated Visuo-Haptic Displays

The idea of seeing and touching virtual objects as in real life was pioneered by Yokokohji et al. [21]. They addressed the importance of colocating haptic and visual feedback, and they referred to this concept as What-you-see-is-what-you-feel (WYSIWYF). Their solution was based on the extraction of an image of the user's real hand, and the composition of this image in the virtual world, which was displayed both visually and haptically. We adapt their approach of segmenting the image of the user's hand based on a chroma-key technique.

Several researchers have assessed the importance of colocating visual and haptic stimuli, as this allows virtual tasks to be carried out from a first-person point of view [22]. Visual and haptic delocation is however quite common, because the construction of a delocated setup is far simpler. Congedo et al. [23] emphasize that, in tasks where the contribution of touch is important, great effort should be undertaken to colocate vision and touch, so that the weight of the nondominant modality, i.e., touch, is not penalized. Spence et al. [24] summarize crossmodal congruency effects involving vision and haptics.

Visuo-haptic colocation can be achieved in several ways, and the most popular ones include workbenches with stereo projection systems [4], [25], mirror-based projection systems where the virtual image occludes the real scene [26], or head-mounted displays (HMDs) with head and device tracking [27]. We use a see-through HMD, with stereo tracking based on the cameras of the HMD.

The recent work of Knoerlein et al. [28] addresses several aspects of visual consistency in the context of a colocated VHMR display, such as occlusion handling, light tracking, and shadowing. As a step of their occlusion handling algorithm, they construct a depth map of the scene exploiting the same cameras used in the see-through display.

2.3 Visual Obstruction with a Haptic Device

As stated in the introduction, the addition of haptic devices introduces bulky obstructive objects in a MR scene. One possible solution to visual obstruction is to use stringed haptic devices, such as the SPIDAR [29]. 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 [2] 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 [4].

Another possible solution to visual obstruction is optical camouflage [30], which consists of covering the obstructive 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. [5] for solving the visual obstruction produced by haptic devices in MR scenes. Our MR paradigm can be perhaps interpreted as a computational camouflage approach. It can also be regarded as an example of diminished reality [31], which was already followed by Bayart et al. [32] to visually remove the haptic device from a VHMR display. Bayart used only one fixed camera, which simplifies the visual removal of the device, but does not allow colocation of haptic and visual stimuli.

In [33], we published our first approximation to VHMR, with a solution to the visual obstruction produced by the haptic device, based on IBR of precaptured background images. In particular, our approach followed Buehler's unstructured lumigraph rendering [34] and Debevec's view-dependent texture mapping [35]. In this paper, we extend the VHMR algorithm by adding correct visual compositing of the virtual tool and objects, the real background, and the user's hand. We handle occlusion problems using a color-detection approach, and we also address misalignment problems suffered with commodity haptic devices.

3 Visuo-Haptic Display Overview

Our aim is to design a VHMR display that allows seamless manipulation of both real and virtual objects. We focus on manipulation metaphors where the user holds a virtual tool to interact with the MR content. For a natural interaction, the user should be able to see his/her own hand holding the virtual tool, and all objects, both virtual and real, should satisfy physical laws, both visually and haptically.

In our work, we employ state-of-the-art simulation techniques to compute the physical interaction between virtual objects and to haptically render the manipulation. In particular, we adopt a constrained-dynamics algorithm for solving deformation and contact of rigid and deformable objects [36], and a multirate haptic rendering algorithm for manipulating rigid handles [37]. To enable contact between real and virtual objects, for each real object we simply model a virtual counterpart in our simulation algorithm. Currently, our solution is limited to the interaction with static real objects. Aleotti et al. [38] discuss other VHMR solutions with dynamic virtual objects and static real ones.

For the visual rendering, we follow a vision-based tracking approach of the user's head, together with video see-through display on an HMD. This solution allows view-dependent colocated display of visual and haptic feedback, and allows the user to see his/her own hand. These two features make manipulation and interaction more natural than looking at a monitor or not seeing the hand.

3.1 The Pipeline

The core of our contribution addresses problems induced by the combination of visual and haptic displays in the MR context, as VHMR introduces problems that are not present when visual or haptic displays are used alone. For completeness, we describe the full display pipeline and discuss design decisions, but many components of the pipeline could be replaced with other solutions. As outlined in the introduction, our contributions target two specific problems of the VHMR display: 1) visual obstruction of the mixed environment produced by the haptic device, and 2) hand-tool misalignment produced by mechanical limitations of commodity haptic devices.

Our visual display algorithm identifies semantically distinct visual elements, and then composes them in a consistent manner, solving occlusion and misalignment issues. Those visual elements consist of the background, the haptic device, the virtual tool, the user's own hand, touchable real objects, and virtual objects.

The whole display pipeline can be outlined as follows:

1. Preprocessing tasks for a given workspace.
   a. We capture an image-based model of the scene's background, using an interactive approach (Section 4.1).
   b. We calibrate the haptic device w.r.t. the scene using an interactive approach (Section 4.2).
2. Common MR precompositing tasks.
   a. An image of the scene (Fig. 1a) is captured.
   b. We compute the camera pose, following a state-of-the-art marker-based tracking approach.
3. Semantic labeling in screen space.
   a. To address the problem of visual obstruction, we identify and segment the region of the image occupied by the haptic device. We track the haptic device and the camera, and transform a kinematic model of the device onto screen space (Section 5.1).
   b. For correct visualization of the user's hand, we use a color detection approach to identify and segment the hand's screen-space projection (Section 5.3).
   c. We also define an interpolation region to gradually blend between the corrected misalignment-free hand configuration and the user's actual arm (Section 5.2).
4. Consistent compositing.
   a. The captured image is processed to remove the haptic device and the user's hand. Then (Fig. 1b), the emptied region in the image is filled with a view-dependent IBR of the background (Section 6).
   b. Virtual objects, including the hand-held tool, are composed in the scene, as shown in Fig. 1c.
   c. And, finally, as shown in Fig. 1d, the user's hand is repainted in the foreground with a virtual displacement that corrects tool-hand misalignments (Section 7).

3.2 Input Data

Our algorithm takes two types of data as input: *static* and *dynamic*. The static data are captured or computed only once during the system calibration stage, before entering the runtime pipeline. The static data consist of:

- The intrinsic camera parameters.
- The arrangement of marker geometry in the world, including the definition of the global reference system.
- A set of images of the background scene, acquired from an unstructured set of viewpoints, sampling the space where the user is expected to move.
- For each image, its extrinsic camera parameters (i.e., camera pose), computed w.r.t. the global reference system.
- A geometric proxy of the background, i.e., several world points with an approximately known position.
- The transformation between the local reference system of the haptic device and the global reference system.
- Geometric and kinematic models of the device.
- The color histogram of the user's hand (with a glove).

At runtime, the algorithm needs only the following *dynamic data*:

- For each eye, an image of the scene coupled with its associated camera pose.
- The configuration of the haptic device in its local reference frame.

Note that the hardware required by our VHMR display algorithm consists only of a camera-based see-through HMD and a commodity haptic device. These requirements are the basic ones for obtaining, separately, visually colocated MR and haptic feedback, and our pipeline does not impose additional needs. Both the static and dynamic data are acquired with the same hardware, and then our VHMR algorithm computes in a fully automated manner the visual and haptic representations to be displayed to the user. From the tracking point of view, our system also employs standard technology, as camera tracking can be performed with regular approaches for MR, and the haptic device configuration is provided by the device's own driver.

3.3 Hardware

In our test implementation, we have used the following hardware: A PHANToM Omni haptic device by SensAble Technologies, an HMD with an eMagin Z800 3D visor, and two external Point Gray flea2-08S2C cameras, and a 2.83-GHz quad-core PC with 4-Gb of RAM equipped with a Quadro FX4600 graphics card.

4 Calibration, Tracking, Acquisition

Setting up our VHMR display involves several preprocessing tasks to acquire the static data necessary for the runtime display algorithm. The first component to be configured is camera calibration and tracking. We calibrate the intrinsic camera parameters using Matlab's Calibration Toolbox [39]. For camera tracking, we have tested both the ARToolkit and ARToolkit PLUS marker-based tracking systems. We place in the scene a set of quadrangular markers, making sure that at least one marker is visible from all points in the user's workspace.

Once the working environment is selected and the haptic device is placed in this environment, the user must acquire background images and calibrate the location of the haptic device in the world. To ease these two tasks, we have designed interactive approaches that leverage the MR display.

4.1 Background Scene Acquisition

The IBR-based background rendering algorithm described later in Section 6 uses as input a set of images of the scene and their corresponding camera transformations. An IBR method can be regarded as a method that interpolates the plenoptic function [40], therefore the input images can be regarded as points scattered on the multidimensional space of the plenoptic function. A well-sampled data set implies a sufficient sampling density over the domain of possible viewpoints. Obtaining a well sampled set of background images is a complicated task because of the difficulty to evaluate reconstruction quality during acquisition.

We have designed an interactive procedure that exploits a basic MR display to build up a well-sampled data set of the scene. The user must first set up the working environment as desired, including tracking markers, but removing the haptic device. Then, as the user moves a tracked camera through the scene, the application presents a preview of the synthesized background using our IBR algorithm. The user may view at the same time the real image from the camera's point-of-view and the synthetic IBR image, and judge whether the quality is sufficient.

Whenever the user considers that the IBR quality is not sufficient from a certain viewpoint, he/she may add the current image to the input data set, simply by issuing a command through the keyboard. The data set is interactively augmented, and the MR preview application uses at all times the currently captured data set. The user must know in advance the region in space and the viewing orientations that will be visited at runtime, to check IBR image quality at the appropriate locations.

4.2 Haptic Device Calibration

The integration of haptic devices in the MR context requires a calibration step to align haptic and world coordinate systems, because colocation errors could diminish usability and compromise user attention [27], [41], [42].

Given a set of points $\{P_i\}$, expressed both in the reference frame of the haptic device, $\{P_{h,i}\}$, and in the global reference frame of the world, $\{P_{w,i}\}$, calibration consists in our case of estimating the rigid transformation (T, R) between the two reference frames. In this work, we adapt the solution proposed by Harders et al. [42]. Note that we assume that the haptic device controller provides us with the position of the end effector w.r.t. the reference frame of the device itself, therefore, there is no need to know the kinematic model of the device explicitly. We first extract the translation T based on the centroids of the point-sets, $T = \frac{1}{n}\sum_i^n P_{h,i} - \frac{1}{n}\sum_i^n P_{w,i}$. Then, we estimate the remaining affine transformation between the point-sets by solving a least-squares problem:

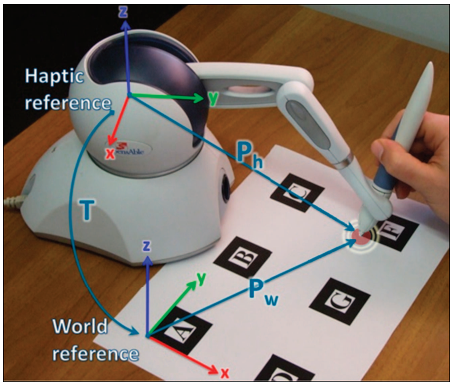

Fig. 2. Interactive MR calibration of a haptic device. To generate a point for the calibration procedure, the MR display shows a flashing ring with known 3D position, and the user places the tip of the handle on the ring, watching the MR display itself for feedback.

$$A = \arg\min \sum_i^n \|P_{h,i} - A \cdot P_{w,i} - T\|^2. \tag{1}$$

Finally, we do a polar decomposition of A to extract a rotation R.

The difficulty of calibration is not the computation of the rigid transformation (T, R), but the sampling and measurement of the input points $(P_{h,i}, P_{w,i})$. We have implemented a simple and interactive calibration procedure that obtains an acceptable level of accuracy without requiring additional hardware infrastructure. Leveraging our basic MR display, the user is interactively assisted and guided to provide sufficient and accurate data for the calibration process. Once the device is placed in the scene, the calibration process can be executed in just a couple minutes.

Our system generates points $P_{w,i}$ in the reference frame of the world, one at a time, and renders them in the MR display (using a flashing ring, as shown in Fig. 2 just under the device tip). The user is then requested to place the tip of the haptic device at the calibration point. Once the user considers, by watching the MR display, that the tip is at the right location, he/she presses a button, the system reads through the device API the coordinates $P_{h,i}$ of the tip in the reference frame of the device, and the procedure moves on to the next point. To improve calibration accuracy, the system requests the user to pick the same points several times, and we use points that are easy to locate, e.g., corners of tracking markers.

With our calibration procedure, we have measured an average error between reference calibration points and collected points of 1.35 mm, with a standard deviation of 0.58 mm. This error is expected given the $\pm 5\%$ nonlinearity of the gimbal encoders of the device.

5 Semantic Labeling in Image-Space

As a prerequisite for the correct visual composition of the various elements in the VHMR scene, we first identify three distinct regions in screen space. For each region, we compute a mask that is used in the subsequent rendering steps. The first region corresponds to the haptic device, which we identify by tracking the actual device w.r.t. the camera and rendering an approximate geometric model. This region in the image will be deleted and repainted using the IBR-based background completion approach. The

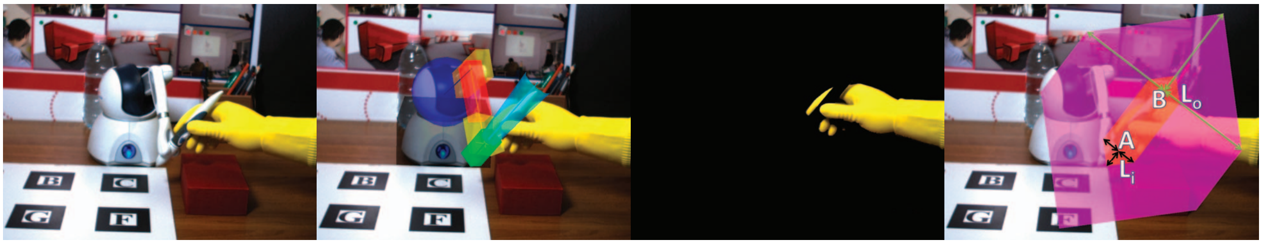

Fig. 3. Visual overview of the image-space semantic labeling. (a) Current camera view of the scene; (b) Color-labeling of the haptic device, obtained by rendering an approximate kinematic model; (c) User's hand segmented following a color detection approach (enhanced by wearing a monochrome glove); (d) Projected grasping volume, labeled by drawing two separate hexagons for the inner and outer grasping areas.

second region corresponds to the user's hand, which we identify using a color detection method. This region will be transformed in image-space to correct tool-hand misalignments, and the resulting void will also be repainted with our IBR approach. By letting the user wear a monochrome glove, we increase the quality of hand segmentation over pure skin-color detection. Finally, the third region identifies the projection of a grasping volume surrounding the device handle, which is used for smoothly blending the user's arm and the transformed image of the hand. This section describes the labeling of regions in the following order: haptic device, grasping volume, and user's hand.

5.1 Masking the Haptic Device in Screen Space

To label the image region obstructed by the haptic device, we exploit knowledge of camera and device transformations. As a preprocess, we define approximate geometric models for the links of the device. We bound each link with a best-fit prism, enlarged by 5 percent to account for tracking and calibration errors. We bound the handle of the device using a truncated cone, however, because of the slight nonlinearity of gimbal encoders. We first define link transformations in the local reference frame of the device using the joint angles read through the device API, and we then transform them to the global reference frame of the scene based on the calibration results described in Section 4.2.

Next, we project the 3D geometric model of the device to screen space, to label the covered image region. The extrinsic camera parameters for the current frame complete the definition of modelview and projection matrices, and we use a rendering procedure on graphics hardware to automatically label the device region in image space. When performing this rendering step, we activate a mask in the stencil buffer for the rendered pixels. Fig. 3b shows the device mask, with the links drawn in different colors.

We have considered other possibilities for labeling the haptic device in image space, in particular painting the device with a specific color and using chroma detection techniques. This approach would be more robust if calibration problems might occur, but it could also suffer from color and/or lighting issues. In our experiments, the robustness of the device and camera tracking appeared to be sufficient, and we found that our approach based on kinematic mask rendering was sufficiently accurate.

5.2 Definition of the Projected Grasping Volume

We define the grasping volume as a 3D region around the device handle, which, projected to image space, can be regarded as a grasping area. This area is itself divided into inner and outer regions, to define blending weights for the consistent compositing of the user's hand described later in Section 7. We label both the inner and outer grasping areas by drawing 2D hexagons centered on the projected axis of the device handle, as depicted in Fig. 3d. The hexagons approximate projected swept-sphere volumes, but they are much simpler to render.

Following the kinematic procedure described in the previous section, we obtain the 3D positions of the end points of the device handle. We then transform these points to image space, using the camera transformation and projection matrices, and we obtain the 2D points A and B shown in Fig. 3d. We define as $L_{AB} = \|A - B\|$ the projected length of the device handle. Then, we compute the width of the outer grasping area, L_o, as 75 percent of L_{AB}. To prevent problems when the device handle is close to parallel to the viewing direction, we set a lower limit for L_o as 10 percent of the diagonal of the viewport (D). The width of the inner grasping area, L_i, is set simply as 20 percent of L_o. Specifically, we compute L_o and L_i based on the following expressions:

$$L_o = \max(0.75 \cdot L_{ab}, 0.1 \cdot D) \qquad \text{and} \qquad L_i = 0.2 \cdot L_o. \quad (2)$$

5.3 Segmentation of the User's Hand

To label the hand in screen space, we use a color-based segmentation approach. Our solution is designed to support direct skin-color detection, but results are enhanced by wearing a monochrome glove. Tens of works have investigated this issue by analyzing camera images in different color spaces, and there are comparisons that discuss pros and cons of each algorithm [43], [44], [45], [46]. We execute the segmentation in the Hue, Saturation, Intensity (HSI) color space [47] to overcome illumination problems (e.g., high intensity under white light), the influence of ambient lights, or the orientation w.r.t. the light source [45].

According to [48], skin-color distribution can be modeled through an elliptical Gaussian joint probability distribution function, defined as

$$p(P_C) = \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2}(P_C - K_C)^T \Sigma^{-1} (P_C - K_C)}, \quad (3)$$

where $P_C = (H, S, I)$ is the pixel color and $K_C = (H_k, S_k, I_k)$ is the key color corresponding to the skin or glove, both expressed in HSI space. Σ is the diagonal covariance matrix,

$$\Sigma = \begin{pmatrix} \sigma_h & 0 & 0 \\ 0 & \sigma_s & 0 \\ 0 & 0 & \sigma_i \end{pmatrix}, \quad (4)$$

where $(\sigma_h, \sigma_s, \sigma_i)$ adjust each component's sensitivity.

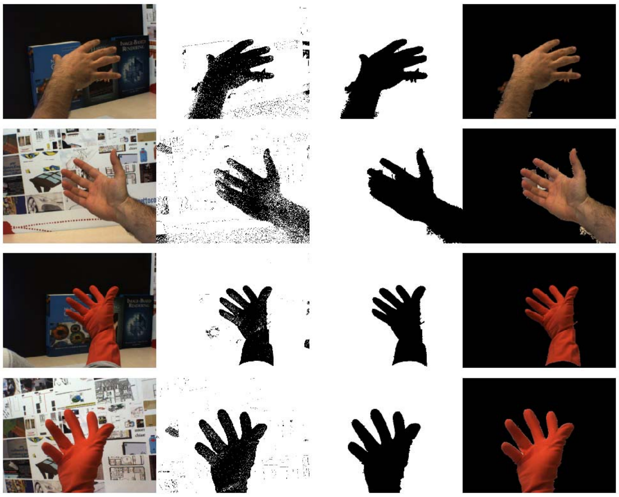

Fig. 4. Comparison of the quality of hand segmentation using the bare skin (top two rows) or a monochrome glove (bottom two rows). The four columns show, from left to right, the captured image, segmentation after applying the Gaussian probability model, segmentation after noise reduction and hole filling, and the final segmented hand.

Then, the segmentation procedure computes the probability $p(P_c)$ for each pixel, and labels it as belonging to the key region if the probability is larger than a threshold value τ. As a preprocess, we calibrate the threshold value τ, the key-color K_C, and the coefficients of the covariance matrix $(\sigma_h, \sigma_s, \sigma_i)$. For calibration, we use an interactive application that displays the current segmentation results, while the user is requested to manually tune the parameters. This interactive calibration approach allows us to personalize the environment to each user's skin or glove, different lighting conditions, and different background scenarios.

The color-based segmentation described above suffers from noise. We improve quality by applying the approach in [49], which consists of a hole-filling step based on morphological reconstruction, followed by an erosion filter for noise reduction, and a final opening by reconstruction.

We further improve the quality of the hand segmentation by increasing the saturation of the video camera (close to a level of 100 percent), which alters color perception, and by wearing a monochrome glove. Fig. 4 offers a qualitative comparison between hand segmentation of the bare skin or wearing a monochrome glove, with two different backgrounds. The four columns show, from left to right, the captured image, the result of segmentation after probability thresholding, the result after hole filling and noise reduction, and the final segmented hand. We used a glove in all the examples reported in the paper.

6 IBR-Based Background Completion

As outlined in Section 3.1, our VHMR pipeline extracts the device and the hand from the image and fills the resulting emptied region using an IBR-based background rendering algorithm. Our approach is strongly inspired on the unstructured lumigraph approach [34]. Based on the acquired background data described in Section 4.1, the scene light field is considered to be known for a set of irregularly distributed rays, and we use view-dependent texture mapping (VDTM) [35] to interpolate light field data from those rays. More precisely, we compute camera blend weights for a discrete set of vertices on the emptied region, we mesh the vertices and define a camera blend field over the emptied region, and compute the final image by blending the results of VDTM. We use two types of input data in a combined manner: a set of prerecorded images of the background scene with known camera positions, and a very rough geometric approximation of the background.

We describe next the full IBR algorithm in three steps: 1) the geometric proxy, 2) meshing and camera blend field computation, and 3) VDTM.

6.1 Geometric Proxy of the Background

As a preprocess, we construct a very rough geometric proxy of the background, following the approach in [34]. The geometric proxy provides depth estimates for the homography transformation of VDTM.

For the office-like settings used in our experiments, the proxy consists of planar desks and planar walls. For example, in our test settings, we use one plane for the desk, and two other planes for the vertical walls forming the background corner. Other smaller objects lying on desks or hanging from walls, such as books, bottles, or pencil holders were not included in the proxy.

6.2 Meshing and Camera Blend Field

As a preprocess, we triangulate the geometric proxy. At runtime, we project its triangles onto the image plane, and thus we obtain a meshing of the emptied region in screen space. The projected vertices of the geometric proxy constitute *geometry vertices* in the context of the unstructured lumigraph algorithm. Note that the original unstructured lumigraph approach computes a Delaunay triangulation of geometry and camera vertices, whereas we use a precomputed triangulation of geometry vertices only.

For each geometry vertex, we define camera weights, and these weights are interpolated inside the mesh triangles, defining a camera blend field. In our implementation, we have simplified the unstructured lumigraph approach, and we select only one source camera, with weight 1, for each geometry vertex, leaving all other cameras with weight 0. This approach largely simplifies the implementation of IBR as a shader. The simplification could also produce a degradation of visual quality due to increased distortion. However, we found that it reduces blur, which appeared to be beneficial. Moreover, we observed that users are mainly focused on the virtual objects that are added into the scene, and a quasi-realistic background appears to be sufficient.

At runtime, to pick the source camera for a geometry vertex V, we compute the viewing direction v of the vertex in the current camera, and compare it against the viewing directions $\{v_i\}$ of the same vertex in all prerecorded background images. We pick the source camera for which the angle between viewing directions is minimal, i.e., $\arg\max v^T \cdot v_i$. We define the viewing direction of a vertex as the unit vector pointing from the camera center to the vertex, expressed in the camera's reference system.

Fig. 5(left) shows the triangulation of the emptied region in magenta. It also shows in yellow an extra ring where we blend the results of VDTM and the captured image, to avoid

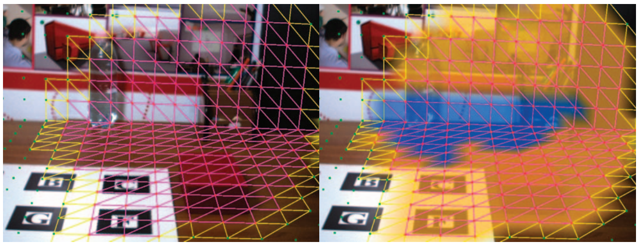

Fig. 5. Left: In magenta, triangulation of the emptied region formed by labeling the haptic device and the user's hand in image space. In yellow, blending ring to avoid discontinuities between IBR-based background completion and the captured image. Right: An example camera blend field.

image discontinuities. In practice, the emptied region is defined using the stencil buffer, which is set during the haptic-device and user-hand labeling steps described in Section 5. Fig. 5(right) shows an example camera blend field.

6.3 View-Dependent Texture Mapping

Given a triangle of the mesh, with image-space vertices $\{a, b, c\}$, we synthesize the background image inside the triangle following VDTM. In other words, we warp and blend the prerecorded background images $\{B_a, B_b, B_c\}$ of the source cameras corresponding to the vertices.

For each vertex, we compute three homography transformations, $\{T_a, T_b, T_c\}$, based on the vertex position, the current camera pose, and the poses of the three source cameras. Then, we map each vertex $V \in \{a, b, c\}$ to each background image $B \in \{B_a, B_b, B_c\}$, to obtain nine pairs of texture coordinates $\{t_{a,B_a}, t_{a,B_b}, t_{a,B_c}, t_{b,B_a}, t_{b,B_b}, t_{b,B_c}, t_{c,B_a}, t_{c,B_b}, t_{c,B_c}\}$. We use these texture coordinates to warp the appropriate portion of each source background image onto the current output triangle. See [50] for mathematical details of VDTM.

We define blending weights for the three source cameras on the three vertices according to vectors $W_a = (1, 0, 0), W_b = (0, 1, 0), W_c = (0, 0, 1)$, with a weight of 1 for corresponding vertex-camera pairs. We have implemented VDTM exploiting regular multitexture rendering in OpenGL, and using the following fragment shader program (written using NVidia's Cg shading language).

```
void vdtm(float3 tBa : TEXCOORD1,
          uniform sampler2D Ba,
          float3 tBb : TEXCOORD2,
          uniform sampler2D Bb,
          float3 tBc : TEXCOORD3,
          uniform sampler2D Bc,
          float4 W : COLOR,
          out float4 color : COLOR)
float4 ca = tex2D(Ba, tBa.xy);
float4 cb = tex2D(Bb, tBb.xy);
float4 cc = tex2D(Bc, tBc.xy);
color = ca * W.x + cb * W.y + cc * W.z;
color.w = W.w;
```

The multitexture feature is used to bind the three background images as three textures. The texture coordinates in the three textures and the blending weights are passed as per-vertex values, and they are hardware-interpolated in the rasterization step to produce per-fragment texture coordinates and weights. Note that we pass the blending weights through the RGB color attribute. We use the alpha channel to define the smooth transition between the synthesized background image and the real captured image.

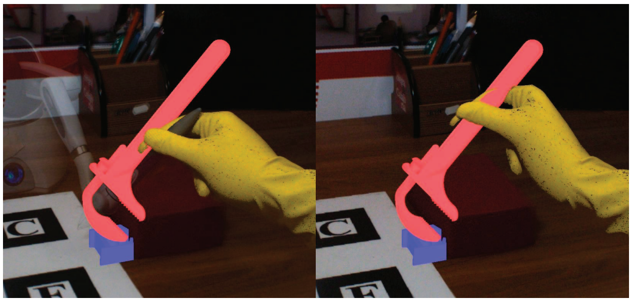

Fig. 6. Left: Naïve composition of virtual objects, resulting in a misalignment of the user's hand and the virtual tool (notice the configuration of the device, rendered semi-transparent). Right: Our consistent composition of the hand, obtained by repainting the hand mask using a correction displacement.

In Section 8, we evaluate the performance of our IBR-based background rendering approach.

7 Consistent Hand Compositing

Naïve visual mixing of real and virtual objects, in particular the user's hand with a virtual tool, produces visually inconsistent scenes. One of the sources of inconsistencies, largely explored in MR research in the past, is incorrect occlusion handling. Another source of inconsistencies, far more unexplored, is the misalignment between hand and tool induced by mechanical limitations in commodity haptic devices, as shown in Fig. 6(left). In this section, we describe our image-space solution for visually consistent composition of the user's hand and a virtual tool, putting special emphasis on misalignment correction.

Occlusion handling is not the focus of our innovation, and we adopt a simple, though not general, approach that assumes that the part of the hand that is visible in the real world remains in the foreground in the mixed scene. We first segment an image mask for the hand, as described in Section 5.3, and once the background is repainted and virtual objects are rendered, we overlay this mask as a texture onto the image. To blend the hand with the user's arm, we define smoothly decaying blending weights in a region of the image corresponding to the grasping-area mask (see Section 5.2). More elaborate approaches based on depth reconstruction, as discussed in Section 2, could be adopted to correctly render virtual objects in front of the hand.

Tool-hand misalignments are produced by stiffness limitations of haptic devices. In the contact configuration shown in Fig. 6(left), the tool is modeled as a rigid body, and it collides with a deformable body. In our approach, contact is solved using constraints [36], and then the visually perceived stiffness corresponds to the stiffness of the deformable body being touched. To ensure stability of haptic rendering, we use a virtual coupling approach [51], which simulates a spring-damper system between the tool and a rigid frame corresponding to the configuration of the haptic handle. Forces and positions are transmitted between the simulated tool and the haptic handle thanks to the deformation of the spring-damper system. With commodity haptic devices, the stiffness of this spring is strongly limited (down to about 200 N/m in our implementation), and then the haptically perceived stiffness and the motion of the user's hand are dominated by the spring's stiffness. Interestingly, visual-over-haptic dominance has often been exploited to convey stiff contact despite haptic device limitations, but in a VHMR setting the disparity between visual and haptic stiffness leads to a noticeable misalignment between the user's hand and the tool.

To correct the misalignment, we start by leveraging the haptic rendering computations, as the elongation of the spring-damper system directly defines the 3D displacement vector $\vec{v}_{3D}$ between the device handle and the virtual tool. Then, simply by concatenating modelview, projection, and viewport transformations, we obtain the equivalent image-space 2D displacement, $\vec{v}_{2D} = Viewport \cdot Projection \cdot Modelview \cdot \vec{v}_{3D}$. Finally, we apply this 2D displacement when we render the image mask of the hand (and the grasping-area mask) as described before.

Fig. 6 compares the composition of the hand and the virtual tool with and without corrective displacement. Despite the improved visual integration, note that in the corrected display, visuo-haptic colocation is not fully satisfied. Fortunately, the magnitude of the inconsistency is imperceivable thanks to visual dominance over proprioception [52]. As the experiments described in Section 8 demonstrate, our misalignment correction can significantly improve user performance in virtual visuo-haptic interaction.

8 Results

As discussed in Section 3, our VHMR display combines the techniques for visual display described throughout the paper with state-of-the-art haptic rendering of deformable and rigid objects. Please see the accompanying video, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2012.107 and at http://www.gmrv.es/Publications/2012/CGBMO12/, for dynamic results of our VHMR display. In this section, we evaluate timings of the overall pipeline and quality of the IBR background completion. But, most importantly, we evaluate the effect on user performance of our computational camouflage and misalignment correction solutions. To this end, we have designed and executed user studies, and our results show that both visual obstruction and tool-hand misalignment may have a negative impact on user performance, which can be improved with our solutions.

8.1 Timings

We execute several tasks in parallel, namely camera tracking, rendering (including labeling), simulation of the virtual world, and haptic rendering. Tracking runs at about 300 fps, and haptic rendering is executed on a hard-real-time process running at 1 kHz. We use SensAble's low-level OpenHaptics library to read the configuration of the PHANToM Omni and write output forces.

The following timings correspond to the example shown in Figs. 1, 3, and 6, using a $1{,}024 \times 768$ viewport. For this example, we recorded 121 images for the background model, and sampled its geometric proxy with 597 vertices. In the rendering process, the major tasks are: labeling of the haptic device (1ms for mask computation), labeling of the user's hand (under 1 ms), and IBR-based background rendering (8.7 ms of CPU time and 3.6 ms of GPU time for the full viewport). Currently, the rendering update in our system is actually limited by the refresh rate of the cameras (30 fps, issued as a nonblocking call).

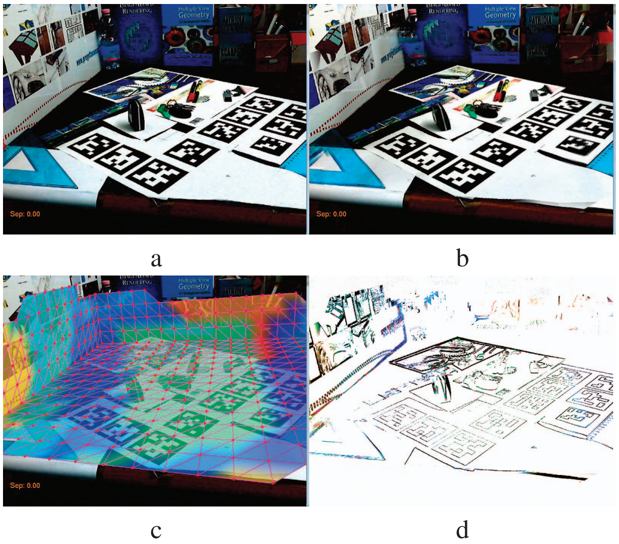

Fig. 7. Comparison of (a) a real image and (b) a synthetic image computed with our IBR approach; (c) Meshing of the geometric proxy overlaid with the camera blend field; (d) Image difference with additional contrast-enhancement.

The simulation of the virtual world may slow down to about 20 fps on complex contact configurations, i.e., when the virtual tool squeezes some deformable object. However, we use a multirate haptic rendering algorithm [37] to provide force feedback at 1 kHz.

Putting all pieces together, the overall latency of the system is governed by the camera refresh rate (30 fps) or by the simulation update rate (20 fps or higher), depending on the contact configuration. Either way, we found that latency was not an issue in our examples.

8.2 Evaluation of IBR

We have evaluated the quality of our IBR background rendering approach by comparing synthetic images to real captured images. Fig. 7a shows the real image and Fig. 7b the synthetic image for a representative scenario. In this example, we used a geometric proxy formed by three orthogonal planes meeting at a corner. Fig. 7c shows the meshing of the geometric proxy, as well as the camera blend field for this particular frame.

Fig. 7d depicts the rendering error, computed as per-pixel difference between the real and synthesized images, with additional contrast enhancement. It appears that error is mostly present in the form of image disparity (due to distortion or parallax error). Such image-disparity error is not perceptually significant in regions dominated by one source camera, because it does not reduce image quality much. In transitions between repainted areas and the actual camera view, we alleviate discontinuities produced by image-disparity by adding a narrow blending band.

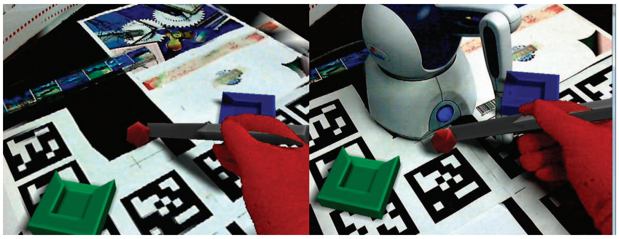

Fig. 8. Scenario for evaluating the impact of our unobstructed display on user performance, with our solution (on the left) and without (on the right). Subjects were supposed to move the virtual red object between the blue and green pockets, using a virtual stick.

In regions where camera blending dominates, image errors are present in the form of ghosting (see, for example, the bottle at the corner). There are two main sources of error: 1) inaccuracies in pose estimation due to tracking limitations, and 2) wrong depth estimation. As discussed in Section 4, we use a limited tracking approach based on the ARToolkit marker-based system. This system is sensitive, for example, to the occlusion of markers. More robust approaches, based, for example, on the tracking of natural features [53], could improve the quality of IBR. A more realistic geometric proxy of the background, obtained through real-time 3D reconstruction (e.g., as recently proposed in [54]), could also improve the quality of IBR.

As mentioned earlier, for the example in Figs. 1, 3, and 6, shown also in the video, available in the online supplemental material, we recorded 121 background images. Camera positions cover a volume with a radius of about 50 cm, and camera orientations cover a vision cone of about 90 degree. There is no absolute guideline in terms of the required camera sampling resolution, as this depends strongly on the size and complexity of the scene. However, our interactive background capture process, described in Section 4.1, allows one to get immediate feedback about reconstruction quality and thus adapt the sampling accordingly.

8.3 Evaluation of the Unobstructed Display

We hypothesize that, if the haptic device is visible in a VHMR display, users will naturally consider the device as part of the MR scene, even though it is an accessory element. If the device occupies part of the MR workspace and interferes with trajectories of the virtual tool, users will not ignore the existence of the haptic device, and this will have a negative impact on task performance. We also hypothesize that, on the other hand, with our computational camouflage algorithm, users will manipulate the virtual tool ignoring the existence of the haptic device, therefore reducing task completion times.

To test our hypothesis, we have designed a task where users must push a virtual ball between two pockets using a virtual stick (as shown in Fig. 8 and the video, available in the online supplemental material). Following a straight path, the device does not reach its mechanical limits, and the handle does not collide with the body of the device. The virtual stick, however, is longer than the real handle, and *visually* collides with the body of the device. If our hypothesis is correct, task completion times should be longer when the haptic device obstructs the MR view, as users will follow trajectories that avoid the device.

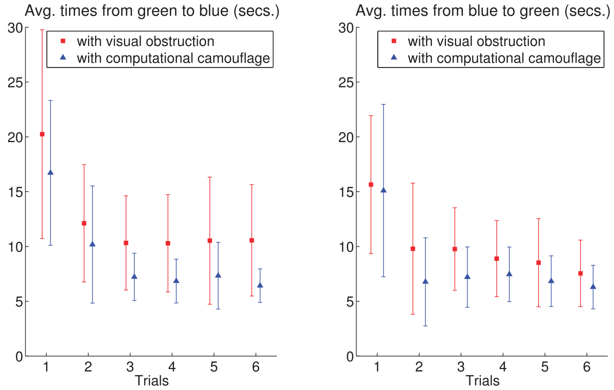

Fig. 9. Comparison of task completion times (in secs.) in the experiment in Fig. 8, with computational camouflage of the haptic device (in blue) and without (in red). The plot shows average times and standard deviation for subsequent trials of the experiment.

Based on this task, we have designed a user study that compares average task completion times with and without our computational camouflage algorithm. 30 subjects, 20 male and 10 female, with ages ranging from 21 to 35, all right-handed, with no previous haptics or MR experience, have participated in the study. Half of the participants have carried out the study with no computational camouflage, i.e., with visual obstruction produced by the haptic device. The other half have carried out the study with our computational camouflage method. These participants wore the HMD before being presented with the MR scene, and were not aware of the existence of the haptic device.

Each participant was asked to move the ball back and forth between the blue and green pockets, as quickly as possible, for a total of six times. Once the ball is detected to be inside a pocket, the system emits a sound, and the user may move the ball to the other pocket. During pilot studies, we observed that some users lost control of the ball and it drifted outside the workspace. This would produce outliers in the experiment, and we decided to place a transparent virtual wall to constrain the ball to the workspace of the device. Prior to the experiment, all participants spent 5 minutes interacting with a similar VHMR scenario to become familiar with the interaction metaphor.

Fig. 9 shows the average completion times for the six subsequent trials of the experiment. We show separately the results for the group that used computational camouflage (in blue) and the group that suffered visual obstruction (in red). In the first two trials there is a clear training effect, and the completion times do not differ significantly across groups. In the last four trials, once the training effect disappeared, the average completion time of green-to-blue-pocket motions (G2B) was 10.34 ± 3.57 s with visual obstrucion, and 6.96 ± 1.43 s with computational camouflage. On the other hand, the average completion time of blue-to-green-pocket motions (B2G) was 8.68 ± 3.3 s with visual obstruction, and 6.94 ± 1.65 s with computational camouflage. Completion times on the last four trials are significantly shorter with computational camouflage, as shown by results of independent samples Welch's t-tests, with $t = 3.49$ for G2B (with $p = 0.003$), and $t = 1.82$ for B2G (with $p = 0.084$). Posthoc power analysis confirms a sufficiently large test group for

G2B ($1-\beta = 0.96$), but not for B2G ($1-\beta = 0.55$). In the G2B motion, results clearly indicate that visual obstruction produced by a haptic device has a negative impact on task performance, and our computational camouflage approach succeeds at improving the performance. Users who see the device may unconsciously consider it as part of the scene, and may not be able to ignore it while carrying out their task. With computational camouflage, on the other hand, users naturally ignore the haptic device.

In addition to task completion times, we have also analyzed path length. The results show exactly the same trend as for completion times. A rough analysis of the trajectories clearly indicates that users with computational camouflage stay closer to a straight path, while users with visual obstruction take a curved path around the haptic device. As an additional piece of evidence, the average speeds for the two groups are extremely similar: 28.52 ± 11.29 mm/s with visual obstruction, and 29.04 ± 6.17 mm/s with computational camouflage. One could expect that, as trials evolved, the group with visual obstruction would grow awareness that the device is simply an accessory and ignore its visual presence, but the timings in Fig. 9 do not reflect such awareness. Another interesting aspect is that the results are somewhat less significant for the path from the blue to the green pocket. We conjecture that the reason could be a combination of the relative position of the objects with respect to the user, and possibly occlusion produced by the user's own hand. Last, some users from the group with computational camouflage mentioned that they could perceive the presence of some type of device, but they could not describe the nature or characteristics of the device.

8.4 Evaluation of Misalignment Correction

We hypothesize that, in a VHMR display with a commodity haptic device and naïve composition of the user's hand and the virtual tool, the stiffness perceived by a user while interacting with a stiff virtual object will be affected by the visible displacement of the user's hand. In other words, despite a small displacement of the stiff object, the mechanical limitations of the device will lead to a larger displacement of the hand, and this will impair stiffness discrimination. We also hypothesize that, on the other hand, our misalignment correction technique will reduce the visible displacement of the hand when interacting with a stiff virtual object, therefore improving stiffness discrimination. Note that the haptically perceived stiffness is the same in both situations, and our hypothesis relies on the dominance of vision over touch for stiffness perception.

To test our hypothesis, we have designed a stiffness discrimination experiment following a protocol similar to that in [55]. Participants are presented with two objects on multiple trials, and they are instructed to select the stiffer one on every trial. One of the objects, selected randomly every trial, has a fixed nominal stiffness, and the stiffness of the other object is automatically adjusted on each trial. Please refer to [55] for the exact stiffness adaptation algorithm, but in essence it reduces the stiffness difference when the user makes several correct selections in a row, and increases the difference when the user makes several wrong selections. Upon convergence, the stiffness difference between the two objects constitutes the just noticeable difference (JND). In our experiment, we have used two flexible prisms, and we have instructed users to push on them laterally, as shown in Fig. 10. The accompanying video, available in the online supplemental material, also shows the experimental setup. We have simulated the deformable prisms using a linear corotational elastic model discretized with finite elements [56], and we have selected the nominal Young modulus such that the prism has a nominal linear stiffness of 1,400 N/m at the top.

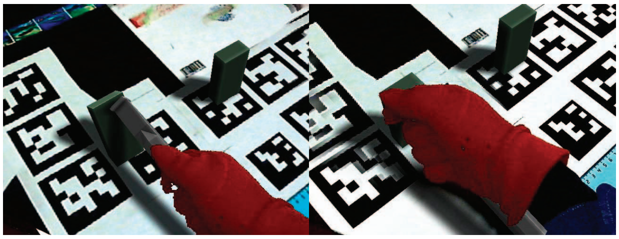

Fig. 10. VHMR task where a user pushes on two prisms to decide which one is stiffer. On the left, with our misalignment correction and, on the right, with tool-hand misalignment.

Based on this stiffness discrimination experiment, we have designed a user study that compares both the JND and the Weber fraction (WF) with and without misalignment correction. 10 subjects, eight male and two female, all right-handed, with ages ranging from 25 to 40, four of them with some previous haptics and/or MR experience, took part in the study. All participants performed the experiment both with and without misalignment correction; half of them first with correction, the other half first without. Prior to the experiment, all participants spent 5 minutes interacting with a simple VHMR scenario to become familiar with the interaction metaphor.

Without misalignment correction, the average JND of stiffness across all participants was 561 ± 114 N/m, and the average WF 0.42 ± 0.088. In contrast, with correction, the average JND was 418 ± 103 N/m, and the average WF 0.33 ± 0.098. A paired-samples t-test demonstrates a significant decrease of the WF, with $t = 4.7$ (and $p = 0.001$). Posthoc power analysis confirms a sufficiently large test group (with $1-\beta = 0.98$). Note that a smaller JND or WF indicates finer stiffness sensitivity, i.e., better stiffness discrimination. Our results clearly indicate that tool-hand misalignments may have a negative impact on stiffness discrimination, which is improved with our misalignment correction.

9 Conclusions and Future Work

In this paper, we have demonstrated a novel paradigm for VHMR. Its major contributions are a computational approach to eliminate visual obstruction introduced by haptic devices, and the visually and mechanically consistent compositing of the user's hand and virtual objects. To the best of our knowledge, our work is the first that presents computational solutions to the visual obstruction problem and to the visual misalignment produced by mechanical limitations in commodity haptic devices.

We have also designed and executed experiments to evaluate the perceptual impact of visual obstruction and tool-hand misalignment. Based on our results, we can affirm that both visual obstruction and tool-hand misalignment can have a negative impact on task performance. Specifically, we found that visual obstruction affects negatively task completion times in VHMR manipulation tasks, and tool-hand misalignment affects negatively stiffness discrimination during contact with stiff objects. Moreover, we have demonstrated that our solutions based on computational camouflage and image-space misalignment correction significantly improve task performance.

We also consider that our findings open up exciting directions for further work, in the study of perceptual implications of VHMR displays and their technical limitations, and in the improvement of technical solutions for VHMR. Our two user studies demonstrate the success of computational camouflage and misalignment correction on certain specific tasks, but the results cannot be extrapolated to general tasks. Many questions remain unanswered. Among them: In what tasks do visual obstruction and tool-hand misalignment compromise performance? What quality level (e.g., amount of noise, tracking error, etc.) is required for computational camouflage and misalignment correction to improve performance instead of hurting it further? In situations with visual obstruction, can users grow awareness of the device so that their performance is not hurt? Do users of a VHMR display with computational camouflage act differently if they are unaware of the existence of the device or if they know that camouflage produces a visual illusion to hide the device?

Despite its innovations, our VHMR pipeline presents multiple limitations, some of which could be improved with recent methods, and others that require further research. Our IBR solution, for example, is based on a proof-of-concept implementation, and its quality can be enhanced through better tracking and depth reconstruction methods. For tracking, natural-feature-based methods such as PTAM [53] could improve robustness. For depth reconstruction, recent methods such as those in [54] and [28] would enable better IBR background repainting, but also a more versatile composition of the user's hand and virtual objects.

Our particular solutions for computational camouflage and misalignment correction also suffer from potential limitations. Since the haptic device is visually removed from the MR scene, the user may inadvertently collide against it, and will have no information about the workspace limitations. These two issues may need to be solved through novel interaction metaphors. Tool-hand misalignment correction introduces a discrepancy between the visual and proprioceptive perception of the hand's position, and it is important to understand the effects of this discrepancy, even though it remained unnoticed in our experience. Similarly, the hand and the arm are blended to alleviate discontinuities, but under extreme misalignments the blended posture may appear unnatural and disturb the user.

The quality and versatility of the overall VHMR display could be improved along several lines as well. One line is the visual composition of real and virtual objects, adding consistent illumination, shadows and reflections [57], or simulating camera effects [58]. Another line is the support for diverse hand postures, which could be addressed by adding a physical prop to the handle of the haptic device, or perhaps by displaying a simulated hand instead of the user's actual hand. The latter introduces additional technical issues for the simulation of the hand, as well as perceptual questions regarding ownership of the virtual hand. Yet another possible improvement concerning the hand is to enable direct haptic interaction through the hand, instead of being limited to tool-based interaction. And the last notorious limitation of the current VHMR setup is that it supports only static (real) objects and a static background. Support for dynamic scenes would require on-the-fly capture of the background for IBR-based computational camouflage, and tracking of dynamic objects for simulating contact with virtual ones.

To conclude, we would like to start trying our VHMR display solutions on real MR applications. Those potential applications must admit commodity haptic devices, which implies that they cannot impose demanding requirements in terms of accuracy and mechanical performance. This category of applications possibly includes some aspects of digital design and prototyping, electronic commerce, medical training on platforms that combine mock-ups with simulated elements, and MR gaming.

Acknowledgments

The authors would like to thank the anonymous reviewers for their comments, and Sofía Bayona for helping with the experimental analysis. This work was funded in part by the Spanish Ministry of Science and Innovation (project TIN2009-07942).

References

[1] M. Harders, G. Bianchi, and B. Knoerlein, "Multimodal Augmented Reality in Medicine," *Proc. Int'l Conf. Human-Computer Interaction,* 2007.

[2] M. Ortega and S. Coquillart, "Prop-Based Haptic Interaction with Co-Location and Immersion: An Automotive Application," *Proc. IEEE Int'l Workshop Haptic Audio Visual Environments and Their Applications (HAVE),* 2005.

[3] B. Knoerlein, G. Szekely, and M. Harders, "Visuo-Haptic Collaborative Augmented Reality Ping-Pong," *Proc. Conf. Advances in Computer Entertainment Technology,* 2007.

[4] N. Tarrin, S. Coquillart, S. Hasegawa, L. Bouguila, and M. Sato, "The Stringed Haptic Workbench: A New Haptic Workbench Solution," *Proc. Eurographics,* 2003.

[5] M. Inami, N. Kawakami, D. Sekiguchi, Y. Yanagida, T. Maeda, and S. Tachi, "Visuo-Haptic Display Using Head-Mounted Projector," *Proc. IEEE Virtual Reality Conf.,* 2000.

[6] A.B. Sekuler and S.E. Palmer, "Perception of Partly Occluded Objects: A Microgenetic Analysis," *J. Experimental Psychology: General,* vol. 121, pp. 95-111, 1992.

[7] P. Milgram and F. Kishino, "A Taxonomy of Mixed Reality Visual Displays," *IEEE Trans. Information Systems,* vol. E77-D, no. 12, pp. 1321-1329, Dec. 1994.

[8] B. Bayart and A. Kheddar, "Haptic Augmented Reality Taxonomy: Haptic Enhancing and Enhanced Haptics," *Proc. Eurohaptics,* pp. 641-644, 2006.

[9] S. Jeon and S. Choi, "Haptic Augmented Reality: Taxonomy and an Example of Stiffness Modulation," *Presence,* vol. 18, no. 5, pp. 387-408, 2009.

[10] D.E. Breen, R.T. Whitaker, E. Rose, and M. Tuceryan, "Interactive Occlusion and Automatic Object Placement for Augmented Reality," *Computer Graphics Forum,* vol. 15, no. 3, pp. 11-22, 1996.

[11] J. Fischer, B. Huhle, and A. Schilling, "Using Time-of-Flight Range Data for Occlusion Handling in Augmented Reality," *Proc. Eurographics Symp. Virtual Environments (EGVE),* pp. 109-116, 2007.

[12] A. Fuhrmann, G. Hesina, F. Faure, and M. Gervautz, "Occlusion in Collaborative Augmented Environments," *Computers and Graphics,* vol. 23, no. 6, pp. 809-819, 1999.

[13] B. Lok, "Incorporating Dynamic Real Objects Into Immersive Virtual Environments," *Proc. Symp. Interactive 3D Graphics,* pp. 31-40, 2003.

[14] M. Kanbara, T. Okuma, H. Takemura, and N. Yokoya, "A Stereoscopic Video See-through Augmented Reality System Based on Real-Time Vision-Based Registration," *Proc. IEEE Virtual Reality,* pp. 255-262, 2000.

[15] J. Schmidt, H. Niemann, and S. Vogt, "Dense Disparity Maps in Real-Time With an Application to . . .," *Proc. IEEE Sixth Workshop Applications of Computer Vision (WACV '02),* pp. 225-230, 2002.

[16] G. Gordon, M. Billinghurst, M. Bell, J. Woodfill, B. Kowalik, A. Erendi, and J. Tilander, "The Use of Dense Stereo Range Data in Augmented Reality," *Proc. Int'l Symp. Mixed and Augmented Reality (ISMAR '02),* pp. 14-23, 2002.

[17] Y. Lu and S. Smith, "GPU-Based Real-Time Occlusion in an Immersive Augmented Reality Environment," *J. Computing and Information Science in Eng.,* vol. 9, no. 2, 2009.

[18] J. Ventura and T. Hollerer, "Depth Compositing for Augmented Reality," *Proc. ACM SIGGRAPH '08,* 2008.

[19] M.-O. Berger, "Resolving Occlusion in Augmented Reality: A Contour Based Approach without 3D Reconstruction," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 91-96, 1997.

[20] W. Lee and J. Park, "Augmented Foam: A Tangible Augmented Reality for Product Design," *Proc. IEEE/ACM Int'l Symp. Mixed and Augmented Reality,* pp. 106-109, 2005.

[21] Y. Yokokohji, R. Hollis, and T. Kanade, "Wysiwyf Display: A Visual/Haptic Interface to Virtual Environment," *Presence,* vol. 8, no. 4, pp. 412-434, 1999.

[22] S. Coquillart, "A First-Person Visuo-Haptic Environment," *Proc. Int'l Conf. Human-Computer Interaction Inst. (HCII),* 2007.

[23] 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,* vol. 15, no. 3, pp. 353-357, 2006.

[24] C. Spence, F. Pavani, A. Maravita, and N.P. Holmes, "Multisensory interactions," *Haptic Rendering: Foundations, Algorithms and Applications,* chapter 2, M.C. Lin and M.A. Otaduy, eds., A.K. Peters, 2008.

[25] J.D. Brederson, M. Iktis, C.R. Johnson, and C.D. Hansen, "The Visual Haptic Workbench," *Proc. PHANToM User Group Workshop,* 2008.

[26] D. Stevenson, K. Smith, J. Mclaughlin, C. Gunn, J. Veldkamp, and M. Dixon, "Haptic Workbench: A Multisensory Virtual Environment," *Proc. SPIE,* vol. 3639, pp. 356-366, 1999.

[27] 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. IEEE/ACM Fifth Int'l Symp. Mixed and Augmented Reality (ISMAR),* 2006.

[28] B. Knoerlein, G. Szekely, and M. Harders, "Enhancing Visual Fidelity in Multimodal Augmented Reality Enviornments," *Proc. Int'l Conf. Computer Graphics, Visualization and Computer Vision (WSCG),* pp. 197-204, 2010.

[29] M. Ishii and M. Sato, "A 3D Spatial Interface Device Using Tensed Strings," *Presence,* vol. 3, no. 1, pp. 81-86, 1994.

[30] M. Inami, N. Kawakami, and S. Tachi, "Optical Camouflage Using Retro-Reflective Projection Technology," *Proc. IEEE/ACM Second Int'l Symp. Mixed and Augmented Reality (ISMAR),* 2003.

[31] S. Zokai, J. Esteve, Y. Genc, and N. Navab, "Multiview Paraperspective Projection Model for Diminished Reality," *Proc. IEEE/ACM Second Int'l Symp. Mixed and Augmented Reality (ISMAR),* 2003.

[32] B. Bayart, J.Y. Didier, and A. Kheddar, "Force Feedback Virtual Painting on Real Objects: A Paradigm of Augmented Reality Haptics," *Eurohaptics '08: Proc. Sixth Int'l Conf. Haptics: Perception, Devices and Scenarios,* pp. 776-785, 2008.

[33] F. Cosco, C. Garre, F. Bruno, M. Muzzupappa, and M. Otaduy, "Augmented Touch without Visual Obtrusion," *Proc. IEEE/ACM Eighth Int'l Symp. Mixed and Augmented Reality (ISMAR),* pp. 99-102, Oct. 2009.

[34] C. Buehler, M. Bosse, L. McMillan, S.J. Gortler, and M.F. Cohen, "Unstructured Lumigraph Rendering," *Proc. ACM SIGGRAPH,* 2001.

[35] P. Debevec, C. Taylor, and J. Malik, "Modeling and Rendering Architecture from Photographs," *Proc. ACM SIGGRAPH,* 1996.

[36] M.A. Otaduy, R. Tamstorf, D. Steinemann, and M. Gross, "Implicit Contact Handling for Deformable Objects," *Computer Graphics Forum,* vol. 28, no. 2, pp. 559-568, Apr. 2009.

[37] C. Garre and M.A. Otaduy, "Haptic Rendering of Complex Deformations through Handle-Space Force Linearization," *Proc. World Haptics Conf.,* Mar. 2009.

[38] J. Aleotti, F. Denaro, and S. Caselli, "Object Manipulation in Visuo-Haptic Augmented Reality with Physics-Based Animation," *Proc. IEEE Int'l Symp. Robots and Human Interactive Comm. (RO-MAN),* pp. 38-43, 2010.

[39] J.-Y. Bouguet, "Camera Calibration Toolbox for Matlab," http://www.vision.caltech.edu/bouguetj/calib_doc/. June 2008.

[40] E. H and J. R, "The Plenoptic Function and the Elements of Early Vision," *Computational Models of Visual Processing,* chapter 1, M. Landy and J. Movshon, eds., MIT Press, 1991.

[41] V.A. Summers, K.S. Booth, T. Calvert, E. Graham, and C.L. MacKenzie, "Calibration for Augmented Reality Experimental Testbeds," *Proc. Symp. Interactive 3D Graphics (I3D '99),* pp. 155-162, 1999.

[42] M. Harders, G. Bianchi, B. Knoerlein, and G. Székely, "Calibration, Registration, and Synchronization for High Precision Augmented Reality Haptics," *IEEE Trans. Visualization and Computer Graphics,* vol. 15, no. 1, pp. 138-149, Jan./Feb. 2009.

[43] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A Survey on Pixel-Based Skin Color Detection Techniques," *Graphicon '03: Proc. Int'l Conf. Computer Graphics,* pp. 85-92, 2003.

[44] M. Shin, K. Chang, and L. Tsap, "Does Colorspace Transformation Make Any Difference on Skin Detection?," *Proc. IEEE Workshop Applications of Computer Vision,* pp. 275-279, 2002.

[45] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A Survey of Skin-Color Modeling and Detection Methods," *Pattern Recognition,* vol. 40, no. 3, pp. 1106-1122, 2007.

[46] S. Phung, A. Bouzerdoum, and D. Chai, "Skin Segmentation Using Color Pixel Classification: Analysis and Comparison," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 1, pp. 148-154, Jan. 2005.

[47] R.C. Gonzales and R.E. Woods, *Digital Image Processing.* Pearson Education, 2008.

[48] J. Yang and A. Waibel, "A Real-Time Face Tracker," *Proc. IEEE Third Workshop Applications of Computer Vision (WACV '96),* pp. 142-147, 1996.

[49] P. Karas, "Efficient Computation of Morphological Greyscale Reconstruction," *Proc. Sixth Doctoral Workshop Math. and Eng. Methods in Computer Science (MEMICS),* pp. 54-61, 2010.

[50] Y. Ma, *An Invitation to 3-d Vision: From Images to Geometric Models,* vol. 26. Springer Verlag, 2004.

[51] J.E. Colgate, M.C. Stanley, and J.M. Brown, "Issues in the Haptic Display of Tool Use," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems,* pp. 140-145, 1995.

[52] E. Burns, S. Razzaque, A.T. Panter, M.C. Whitton, M.R. McCallus, and F.P. Brooks Jr., "The Hand Is Slower than the Eye: A Quantitative Exploration of Visual Dominance over Proprioception," *Proc. IEEE Virtual Reality Conf.,* 2005.

[53] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," *Proc. IEEE/ACM Sixth Int'l Symp. Mixed and Augmented Reality (ISMAR '07),* pp. 225-234, 2008.

[54] R. Newcombe and A. Davison, "Live Dense Reconstruction with a Single Moving Camera," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 1498-1505, 2010.

[55] E. Karadogan, R. Williams, J. Howell, and R. Conatser Jr., "A Stiffness Discrimination Experiment Including Analysis of Palpation Forces and Velocities," *Simulation in Healthcare,* vol. 5, no. 5, pp. 279-288, 2010.

[56] M. Müller and M. Gross, "Interactive Virtual Materials," *Proc. Graphics Interface,* 2004.

[57] M. Aittala, "Inverse Lighting and Photorealistic Rendering for Augmented Reality," *The Visual Computer,* vol. 26, no. 6, pp. 669-678, 2010.

[58] G. Klein and D. Murray, "Simulating Low-Cost Cameras for Augmented Reality Compositing," *IEEE Trans. Visualization and Computer Graphics,* vol. 16, no. 3, pp. 369-380, May/June 2010.

Francesco Cosco received the BS, MS, and PhD degrees in 2003, 2005, and 2011, respectively, in mechanical engineering from the University of Calabria. He is currently a teaching assistant and an associate researcher in the Department of Mechanical Engineering of the University of Calabria. His main research areas are virtual and mixed prototyping; visuo-haptic, mixed and virtual reality focused on industrial applications (e.g., usability analysis, mock-up analysis, and early design evaluations). During the PhD program (2009), he was a visiting researcher at the Department of Computer Science's Modeling and Virtual Reality Group (GMRV) at Universidad Rey Juan Carlos (URJC Madrid), where he started part of the investigations related to this paper.

Carlos Garre received the degree in computer science from Universidad Politecnica de Madrid and the master's degree on computer graphics and virtual reality from Universidad Rey Juan Carlos (URJC Madrid) in 2008. He is currently a teaching assistant and working toward the PhD degree in the Department of Computer Science's Modeling and Virtual Reality Group (GMRC) at URJC Madrid. His main research areas are haptic rendering and physically based computer animation.

Fabio Bruno received the MS degree in 2001, in industrial engineering and the PhD degree in 2005, in mechanical engineering from Università della Calabria. He is an associate professor in the Department of Mechanical Engineering at Università della Calabria (Italy). He teaches the course of Technical Drawing and Virtual Prototyping at the Faculty of Engineering. His main research areas are mixed reality and computer vision and their application in virtual prototyping and digital heritage. He is a member of the IEEE.

Maurizio Muzzupappa received the PhD degree in 1992 at the Department of Mechanical Engineering of the Università degli Studi di Pisa. He is an associate professor of Computer Aided Design in the Department of Mechanical Engineering at Università della Calabria, Italy. His current research activities include: applications of virtual and augmented reality to the design process of industrial products (in particular the usability of the user interface). Other topics are: participatory design, CAD automation, and reverse engineering.

Miguel A. Otaduy received the BS degree in 2000 in electrical engineering from Mondragon University, and the MS and PhD degrees in 2003 and 2004, respectively, in computer science from the University of North Carolina at Chapel Hill. He is an associate professor in the Department of Computer Science's Modeling and Virtual Reality Group (GMRV) at Universidad Rey Juan Carlos (URJC Madrid). His main research areas are physically based computer animation, haptic rendering, contact modeling, virtual reality, and geometric algorithms. From 2005 to 2008, he was a research associate at ETH Zurich, and then he joined URJC Madrid. He has published more than 50 papers in computer graphics and haptics, and has recently cochaired the program committees for the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (2010) and the Spanish Computer Graphics Conference (2010). He is a member of the IEEE.