

# Shape and Weight Rendering for Haptic Augmented Reality

Stefano Scheggi<sup>1</sup>, Gionata Salvietti<sup>1</sup> and Domenico Prattichizzo<sup>1,2</sup>

<sup>1</sup> Department of Information Engineering, University of Siena, Italy

<sup>2</sup> Italian Institute of Technology, Genova, Italy

{scheggi, salviettigio, prattichizzo}@dii.unisi.it

**Abstract**—Haptic interfaces in Augmented Reality scenarios are an efficient way to enhance the interaction with a virtual world. We introduce a new system that involves both hands in virtual object experience. We render contemporary shape and weight of a virtual object, merged in a real world environment, using a commercial haptic interface and a new kind of finger-worn device. We test the system with an experimental setup where a virtual cube is placed over a real marker. The system developed can be used in several applications of augmented reality where rendering only the object shape is not enough to perceive the virtual world. It can also lead towards the introduction of more wearable devices in the haptics augmented reality context.

## I. INTRODUCTION

Augmented Reality (AR) interfaces involve the overlay of virtual imagery onto the real world. The goal is to blend reality and virtuality in a seamless manner, enhancing the user's sense of immersion inside the virtual world by augmenting a completely synthetic world with real world scenes. As consequence, Augmented Reality interaction techniques need to be as intuitive as possible to be accepted by end users.

In the past a variety of interaction methods have been presented including the use of free hand cooperation with virtual objects. However, while great advances have been made in AR display technologies, there has been little research on the use of Augmented Reality and haptic interfaces for weight and shape recognition.

In one of the first contributions [13], the authors describe an augmented reality system that merges synthetic haptic input into user perceptions of the real environment. The interconnection with the augmented environment is obtained using a Phantom haptic interface. This system allows user interaction such as moving or lifting a virtual object. However, object shape and weight can not be perceived at the same time.

Over the years, two main fields of research have been explored to improve Augmented Reality perception: virtual environment enhancement and more sophisticated haptic interface integration. The work presented in [2] is an example of the former case. The authors carried out a technique for fingertip-based interaction. The proposed setup implements a simple set of gestures which enable the user to directly manipulate virtual objects. Although a natural and intuitive manipulation of the virtual objects is presented, the haptic feedback is only addressed to electronic vibrating buzzers.

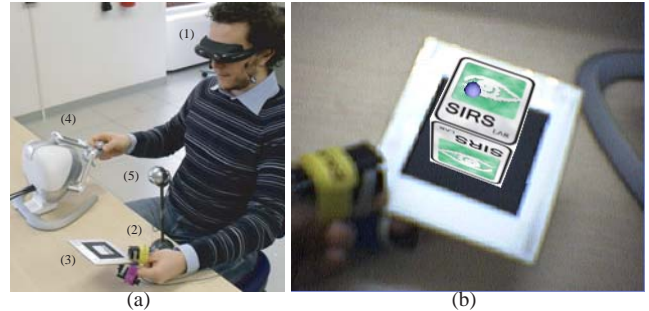


Fig. 1. *General framework*: (a) in the user left hand the finger-worn devices (2) allow to feel the virtual object weight, while a FALCON (4) is devoted to the rendering of object shape. The real scene is captured by a fixed webcam (5) while a video eyewear system (1) displays the Augmented Reality environment built on the marker (3); (b) fusion of the real and virtual world and Falcon avatar (blue sphere).

Consequently the haptic feedback does not allow to accurately perceive the virtual objects.

In [1], a recent work on AR technologies for medical training is presented. In particular, authors focused on augmenting a dummy leg with virtual soft tissue. They proposed a multimodal AR setup, including calibration methods, hybrid tracking, synchronization and a distributed computational framework. Force feedback during manipulation is generated via a haptic device. Since high stability and accuracy of the overall system are required, additional devices, such as an infrared optical 6-DoF tracking device, are necessary. Furthermore a calibration and synchronization process between all components are needed.

Regarding the haptic interface integration, an articulated device has been presented in [14]. Authors developed a transparent lightweight string-based haptic interface suitable for multiple fingers. This solution requires a heavy computational power especially for what concerns the estimation of the virtual hands poses. Moreover, the calibration process of the haptic interface is manually performed with a consequent low performances.

Recently, [6] proposed an ungrounded portable device that provides kinesthetic sense to the fingers. Although the user can manipulate the device without any movement restriction, the authors tested only the force that is applied parallel to the interface axis. So further efforts are necessary to show device capabilities in representing realistic sensations.

The target of our work is to allow the user to perceive

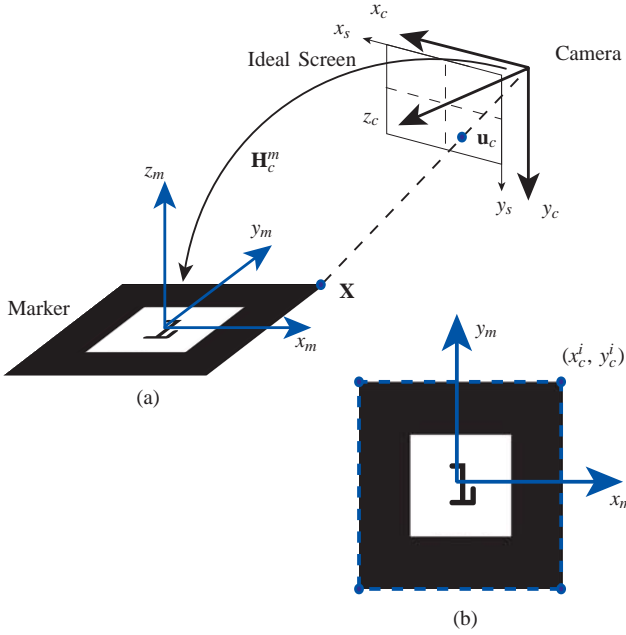


Fig. 2. *ARToolkit* coordinates frames: (a) camera frame  $\langle x_c, y_c, z_c \rangle$ , marker frame  $\langle x_m, y_m, z_m \rangle$  and ideal screen frame  $\langle x_s, y_s, z_s \rangle$ ; (b) marker line contour (dotted lines) and corners  $(x_c^i, y_c^i)$ ,  $i = 1, 2, 3, 4$ .

at the same time the shape and the weight of a virtual object by using both hands<sup>1</sup>. Thus, two haptic interfaces are needed. To the best of our knowledge research involving integration of multiple haptic devices with AR scenarios is new. The proposed system consists in using a finger-worn haptic display, introduced by Minamizawa et al. in [11], to represent the mass perception of a virtual object and a NOVINT Falcon [5] for the shape rendering. There are two main reasons for these choices. First of all, the use of both hands enhances the interaction with the AR helping the user to better feel the augmented environment. Secondly, the finger-worn display allows to extend the overall workspace since this is not a grounded mechanism. This permits to move freely the marker where the virtual object is overlaid.

The paper is organized as follow. In section II an overview of the proposed system is given. A description of graphic and haptic rendering is presented. In section III a possible application is discussed. Finally, conclusions and future work are outlined.

## II. SYSTEM OVERVIEW

The basic idea of the AR setup is to capture a real scene with a fixed camera, superimpose virtual objects and display the scene with a head-mounted display (see Fig. 1). In this context, two different haptic devices have been integrated: a Falcon haptic interface for virtual object shape rendering and three finger-worn devices to ensure weight perception.

<sup>1</sup>Videos of the application can be downloaded at: <http://sirslab.dii.unisi.it/research/haptic/projects/haptikar/>

```

1: for all video frames in input  $I$  do
2:   detect-marker ( $I$ )
3:   if a marker is found then
4:     Estimate  $\mathbf{H}_c^m$  using eq. (4)
5:     Draw the computer graphics model on the real
       world from the same position of the marker
6:   end if
7: end for
8:
9: function detect-marker ( $I$ )
10:  Detect image contour
11:  Estimate line contour and extract corners
12:  Select square shapes
13:  Check if retrieved square shapes match the template
14: end function

```

Table 1: *ARToolkit* fundamental steps.

### A. Graphics rendering

We used the *ARToolKit* library [7], [8] to compute the camera position and orientation with respect to the marker reference system. *ARToolKit* is a C and C++ language software library that lets programmers easily develop Augmented Reality applications. It uses computer vision techniques to compute the real camera position and orientation relative to marked cards, allowing the programmer to overlay virtual objects onto these cards. Given square markers are used as a base of the coordinates frame where virtual objects are represented.

Let us consider the setup reported in Fig. 2 where a perspective camera  $\langle c \rangle$  is observing a marker  $\langle m \rangle$ . If we call the 3-D coordinates of a point  $\mathbf{X}$  relative to the camera and marker reference frame  $\mathbf{P}_c \in \mathbf{R}^3$  and  $\mathbf{P}_m \in \mathbf{R}^3$ , they are related by a rigid-body transformation:

$$\mathbf{P}_c = \mathbf{R}_c^m \mathbf{P}_m + \mathbf{t}_c^m \quad (1)$$

where  $\mathbf{R}_c^m \in SO(3)$  is the rotation matrix which relates the camera and marker reference frame and  $\mathbf{t}_c^m$  the corresponding translation vector. Considering  $\tilde{\mathbf{P}}_c, \tilde{\mathbf{P}}_m$  the relative extension in homogeneous coordinates, eq. (1) can be written as

$$\tilde{\mathbf{P}}_c = \mathbf{H}_c^m \tilde{\mathbf{P}}_m \quad (2)$$

where

$$\mathbf{H}_c^m = \begin{bmatrix} \mathbf{R}_c^m & \mathbf{t}_c^m \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}.$$

Let us assume that the intrinsic camera calibration matrix is given by,

$$\mathbf{K} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where  $f_x, f_y$  (pixels) denote the focal lengths of the camera along the x and y directions,  $s$  is the skew factor and  $(u_0, v_0)$  (pixels) is the principal point of the CCD. According to

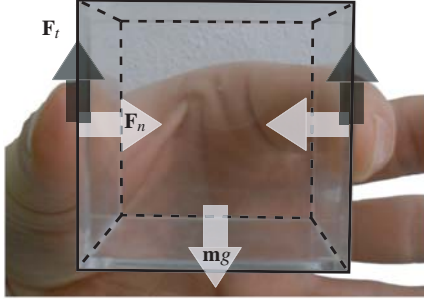


Fig. 3. Vertical stress  $\mathbf{F}_n$  and shearing stress  $\mathbf{F}_t$  between fingers and object in grasping. In this case gravity generates  $\mathbf{F}_n$  and  $\mathbf{F}_t$  represents the grip force.

perspective projection models, the projection  $\tilde{\mathbf{u}}_c \triangleq [x_c, y_c, 1]^T$  of  $\mathbf{P}_c$  on the camera is given by

$$\tilde{\mathbf{u}}_c = \mathbf{K}[\mathbf{I} \ \mathbf{0}] \tilde{\mathbf{P}}_c, \quad (3)$$

where  $\mathbf{I} \in \mathbf{R}^{3 \times 3}$  represents the identity matrix. By putting together eq. (2) and eq. (3) we obtain,

$$\tilde{\mathbf{u}}_c = \mathbf{K}[\mathbf{I} \ \mathbf{0}] \mathbf{H}_c^m \tilde{\mathbf{P}}_m$$

which describes the projection on the camera image plane of a 3-D point expressed in the marker reference frame  $\langle m \rangle$ . The marker pose and position ( $\mathbf{H}_c^m$ ) can be obtained by minimizing the reprojection error

$$err = \frac{1}{4} \sum_{i=1,2,3,4} (\hat{x}_c^i - x_c^i)^2 + (\hat{y}_c^i - y_c^i)^2 \quad (4)$$

where  $\hat{x}_c^i, \hat{y}_c^i$  are noisy measurements of the  $i$ -th marker corner and  $x_c^i, y_c^i$  are the ideal corresponding points (see Fig. 2), [9, ch.6], [4, ch.4]. Table 1 briefly gives an overview of the ARToolkit behaviour.

### B. Haptic rendering

Most of the previous works on using haptic devices to better interact with augmented reality, are focused either on the object shape or on the object weight rendering. Our target is to enhance the interaction with the virtual environment simulating both perceptions simultaneously using two different interfaces. A wearable haptic device has been used to represent gravity force of the AR object. Conventionally, it is believed that it is necessary to reproduce the proprioceptive sensation in order to render the weight of an object, thus a grounded device is required. However, it has been demonstrated that the vertical and shearing forces generated by the deformation of the fingerpads can reproduce reliable weight sensations even when the proprioceptive sensations on the wrist and the arm are absent [10]. Forces perceived at the fingerpads can be divided into vertical stress and shearing stress. A simple task like grasping a box with the thumb and the index as showed in Fig. 3, can be taken as example. The normal stress ( $\mathbf{F}_n$ ) is the grip force exerted by the two fingers and the tangential stress ( $\mathbf{F}_t$ ) is the gravity force of

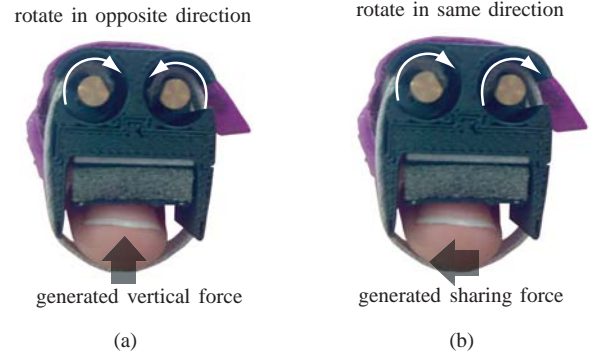


Fig. 4. Methods to generate vertical stress (left) and shearing stress (right) on fingerpad.

the object. According to these observations Minamizawa et al. have designed a mechanism for haptic display, based on the principle showed in Fig. 4. This device consists of two DC motors and a belt as in Fig. 5. To render the grip sensation (normal force), the dual motors are driven in opposite directions of rotation so they roll up the belt. The vertical stress is then generated on the fingerpad of the operator and to get the tangential force, the motors are driven in the same direction of rotation as in Fig. 4(b).

Finger-worn interfaces are light, wearable and quite transparent to the user but they present some drawbacks. One of this is the possibility to send back forces only in two directions even when three directions are needed (for example in a contact point simulation in a 3-D environment).

To better understand which forces have to be rendered to the user through the finger-worn device, we now describe the Haptic AR scenario. Considering that the operator sizes the marker in a corner using thumb opposite to index and medium finger (Fig. 1(b)), three contact points arise that we assume as soft finger [12]. Let us refer to the vector  $\mathbf{w}_g \in \mathbf{R}^3$  as the weight force applied at the object center of mass. Furthermore, we define the vector  $\mathbf{w}_m$  as the weight

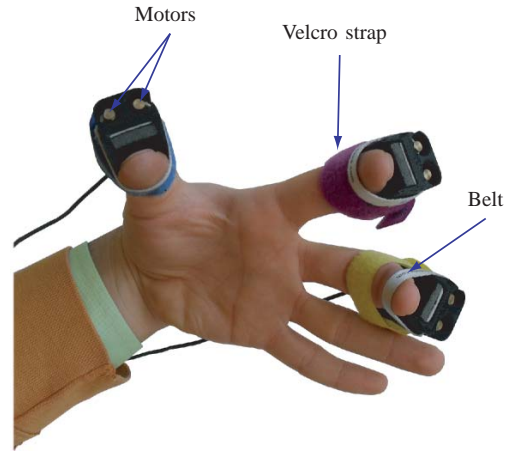


Fig. 5. Finger-worn haptic interface: two motors allow to generate both vertical and shearing stress by rolling up the belt. The Velcro strap fixes the device at the finger.

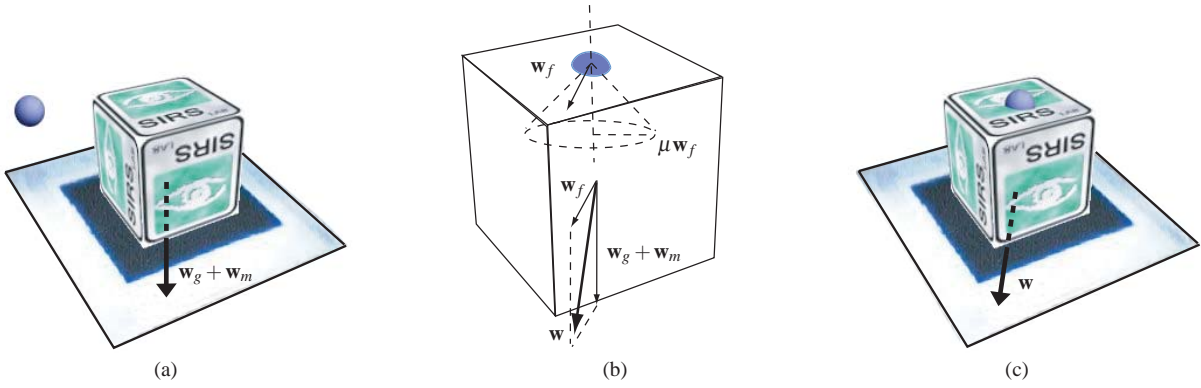


Fig. 6. *Virtual object interaction*: (a) when no interaction exists between the Falcon and virtual objects, the finger-worn devices only render the object mass ( $\mathbf{w}_g$ ) since the marker mass ( $\mathbf{w}_m$ ) is already exerted by the hand; (b) when the Falcon interacts with objects, also the forces exerted by the user ( $\mathbf{w}_f$ ) contribute in the resulting force computation, while  $\mu\mathbf{w}_f$  represents the radius of the friction cone; (c) the resulting force ( $\mathbf{w}$ ) to be rendered through finger-worn devices.

force due to marker mass. We assign to the virtual object a fixed mass and we compute the orientation of the gravity force according to marker rotations referred to the camera frame. In addition, a static friction coefficient  $\mu$  is assigned to the whole surface of the virtual object. When the object is touched by the Falcon avatar, the user applies a force  $\mathbf{w}_f$  that can be computed through the information coming from the haptic interface. We define the resultant  $\mathbf{w}$  of the forces exerted at the fingertip as

$$\mathbf{w} = \mathbf{w}_g + \mathbf{w}_m + \mathbf{w}_f. \quad (5)$$

Envisaging the grasp analysis of the system, the resultant vector can be rewritten as

$$\mathbf{w} = \mathbf{G}\mathbf{p} + \alpha N(\mathbf{G}) \quad (6)$$

where  $\mathbf{G} \in \mathbf{R}^{6 \times 12}$  is the Grasp matrix defined as in [12],  $\mathbf{p}$  is the vector of forces applied at the contact point defined as

$$\mathbf{p} = [p_{1x}, p_{1y}, p_{1z}, p_{1\omega}, p_{2x}, p_{2y}, p_{2z}, p_{2\omega}, p_{3x}, p_{3y}, p_{3z}, p_{3\omega}]^T,$$

$N()$  denote a matrix whose columns form a basis for the nullspace of  $\mathbf{G}$  while  $\alpha$  is a generic vector. The forces scenario is shown in Fig. 6.

Note that for some marker position or for objects with shapes different from the cube, the weight force should be rendered also from the haptic interface. Anyway, we consider for simplicity that only the finger-worn device are devoted to weight rendering, so we concentrate the grasp analysis on the three fingers.

Considering eq. (5) and (6), vector  $\mathbf{p}$  can be computed as

$$\mathbf{p} = \mathbf{G}^\dagger \mathbf{w}_g + \mathbf{G}^\dagger \mathbf{w}_m + \mathbf{G}^\dagger \mathbf{w}_f + \alpha N(\mathbf{G}) \quad (7)$$

Anyway, using wearable devices, we must select which forces have to be displayed by the finger-worn haptic interfaces. In fact, the components  $\mathbf{G}^\dagger \mathbf{w}_m$  and all the internal forces belonging to the nullspace of  $\mathbf{G}$  are directly exerted by the hand once the marker is grasped. Thus, we can assume

$$\mathbf{p} = \mathbf{G}^\dagger \mathbf{w}_g + \mathbf{G}^\dagger \mathbf{w}_f. \quad (8)$$

At this point another issue has to be taken in account. As we have seen above, only two forces, normal and tangential, can be reproduced by the finger-worn device. So that, a matrix  $\mathbf{M}_a$  (Actuation Matrix) that allows to map vector  $\mathbf{p}$  in corresponding forces  $\mathbf{p}_a \in \mathbf{R}^{6 \times 1}$ , representing normal and tangential forces in each contact point, is needed. Vector  $\mathbf{p}_a$  has to be rendered by the devices.

This definition is contact points position dependent. If contacts take up different configurations, matrix  $\mathbf{M}_a$  has to be recomputed. At the end, vector representing the force values produced by haptic interfaces is computed by using

$$\mathbf{p}_a = \mathbf{G}_a^\dagger \mathbf{w} \quad (9)$$

where  $\mathbf{G}_a = \mathbf{M}_a \mathbf{G}$  is called Grasp Actuation matrix.

Concerning the shape rendering, a NOVINT Falcon is used. This device provides 3-DoF displacement sensing and force rendering, thus it allows point interaction in a 3-D virtual environment. The workspace of  $10.16 \times 10.16 \times 10.16$  cm, the force capability bigger than 1 kg and the position resolution more of 400 dpi, makes the Falcon suitable for our application. Moreover, this is one of the cheapest commercial device available.

In order to obtain correct displacement of the Falcon avatar in the virtual reality and to render the forces to the user in the opportune direction, orientation of the device reference frame has to be known. Fig. 7 shows the reference frame  $\langle x_f, y_f, z_f \rangle$  in which the position of the haptic end-effector is expressed. Furthermore, the god-object algorithm [15] has been used for the computation of shape rendering.

### C. Setup description

Central to AR setup are the reference frame definitions (see Fig. 7). In our work we have considered both camera and Falcon reference frames fixed. It arises that the relative pose is known during the experiment. This constraint allows to simplify the graphical rendering of the Falcon avatar and its interaction with virtually augmented objects. On the contrary, no particular position constraints are imposed on the marker other than camera visibility ones.

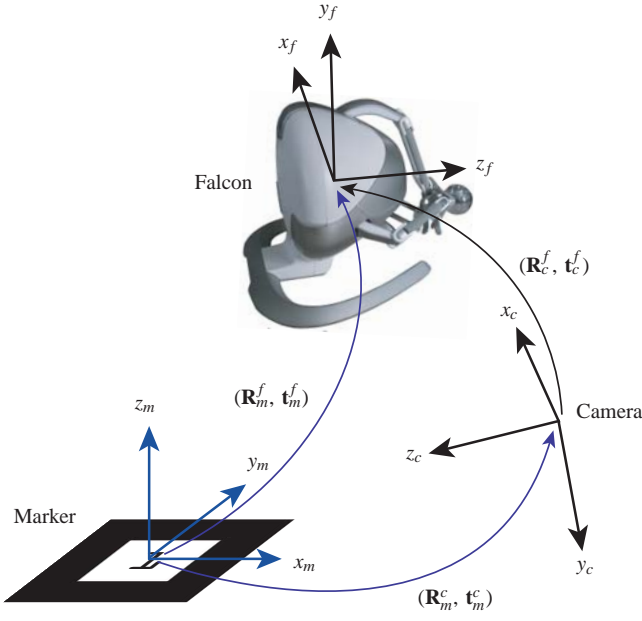


Fig. 7. The Falcon haptic interface and relative reference frame  $\langle x_f, y_f, z_f \rangle$ . All reference frames are related by a rigid motion transformation.

Since all the virtually augmented objects are referred to the marker coordinates system, a rigid transformation  $(\mathbf{R}_m^f, \mathbf{t}_m^f)$  should be applied to the Falcon avatar to correctly interact with the virtual scenario. In the proposed setup we have assumed that the translational vector is  $\mathbf{t}_m^f = \mathbf{0}$ . This assumption allows the avatar to be always able to touch the virtual object and allows to overcome the limited workspace of the haptic interface with respect to the camera field of view. As a consequence, in the reminder of this section, only rotational transformations between the different coordinates systems are considered. The rotation matrix between Falcon and marker frame can be represented as

$$\mathbf{R}_m^f = \mathbf{R}_m^c \mathbf{R}_c^f \quad (10)$$

where  $\mathbf{R}_m^c = (\mathbf{R}_c^m)^{-1} = (\mathbf{R}_c^m)^T$  is estimated by the ARToolkit (see Sect. II-A) and  $\mathbf{R}_c^f$  is known. Thereafter a generic point  $\mathbf{P}_f$  expressed in the Falcon coordinates frame can be expressed in the marker reference frame as

$$\mathbf{P}_m = \mathbf{R}_m^f \mathbf{P}_f.$$

However, the opposite transformation should be applied to the force vector in order to be correctly rendered by the haptic interface. Let  $\mathbf{w}_m$  be the vector representing the force in the marker frame. From eq. (10) one gets

$$\mathbf{w}_f = (\mathbf{R}_m^f)^{-1} \mathbf{w}_m.$$

### III. APPLICATION DESCRIPTION

A real scenario, such a desktop, is captured by a fixed webcam and displayed to the user via a video eyewear. A virtual cube with a given weight is superimposed in the scene using the ARToolkit. Finger-worn and Falcon interfaces are used for haptic rendering. A Logitech QuickCam camera with a frame rate of 30 fps at  $640 \times 480$  pixels is used. The

1:	Initialize Falcon haptic interface and ARToolkit in different threads
2:	<b>for all</b> video frames in input $I$ <b>do</b>
3:	detect-marker ( $I$ )
4:	<b>if</b> a marker is found <b>then</b>
5:	Estimate $\mathbf{H}_c^m$ between the marker and the camera
6:	Read the Falcon end-effector position $\mathbf{P}_f$
7:	Apply the transformation matrix to retrieve the Falcon avatar $\mathbf{P}_m = \mathbf{R}_m^f \mathbf{P}_f$
8:	Check for collision between Falcon avatar and virtual objects
9:	<b>if</b> a collision is found <b>then</b>
10:	Compute the force vector $\mathbf{w}_m$
11:	$\mathbf{w}_f = (\mathbf{R}_m^f)^{-1} \mathbf{w}_m$
12:	Send $\mathbf{w}_f$ to the Falcon haptic interface
13:	<b>end if</b>
14:	Compute the force $\mathbf{p}_a = \mathbf{G}_a^\dagger \mathbf{w}$
15:	Send $\mathbf{p}_a$ to the finger-worn haptic interface
16:	Display the virtual model and Falcon avatar
17:	<b>end if</b>
18:	<b>end for</b>

Table 2: Graphics and haptic rendering steps.

camera is rotated of  $-\pi/6$  about  $x$ -axis with respect to the horizon plane. We have chosen this configuration to improve visibility of the marker when different motions and rotations occur. Referring to Fig. 7, the rotational displacement  $\mathbf{R}_c^f$  between the Falcon and the camera frame can be obtained as

$$\mathbf{R}_c^f = \mathbf{R}_x \left( -\pi + \frac{\pi}{6} \right). \quad (11)$$

We have used the Haptik Library [3] to interface the Falcon with the system. With this library it is possible to acquire the end-effector position and send back the force to be rendered. The haptic loop and vision loop run in different threads. The former is updated with a fixed refresh rate of 1kHz, while the latter is carried out with an update of about 30Hz. This is mainly limited by the frame rate of the camera. Haptik Library includes all needed API operation and, furthermore, it allows to use several haptic devices.

Concerning finger-worn device interfacement, we have used a PCI-serial connection. No specific library has been necessary. The control of these interface is obtained directly by C code that handles serial port. To avoid Actuation matrix redefinition as specified in Sect. II-B, the fixed grasp finger configuration is obtained by drawing the contact point directly in the marker support. This decision does not interfere with the user comfort and enhances the interactive experience. Moreover, this configuration allows the user to avoid occlusion problems.

Table 2 shows the pseudo code of our haptic Augmented Reality application.

The presented setup has been shown at a public of 15 subjects (all male ages 23-33 and good health) trying out the application. From these trials we have gained insight into how people use the proposed experience and especially the

problems we have to solve. The experiment has been splitted in two different phases: one with virtual weight rendering and one without. The subjects tried the case without weight rendering first and the case with weight rendering second. For both cases is asked to the user to move the marker and interact with the virtual object. The cube mass was 0.1 kg and the friction coefficient was varied from 0.1 to 0.4 with steps of 0.05. After each experiment, the following questions were asked. Does the weight rendering influence the perception of the virtual object? Does the weight rendering improve the interaction with the Falcon avatar? Did you feel the friction coefficient variation in the first experiment? Did you feel the friction coefficient variation in the second experiment? Does the weight rendering help you to feel the friction coefficient variation? Each answer was measured by a 7-point Likert scale ranging from “not at all” (1) to “very much” (7). More than 80% of the interviewed people have asserted that to move an object weightless does not help to feel the virtual world merged into the real one. During the second experiment, with the finger-worn device enabled, people have confirmed that the experience was improved especially with high friction coefficient. This was mainly caused by the fact that when the user touches the cube, the finger-worn devices allow to render both the mass of the object and the forces exerted by the user via the Falcon interface (see Fig. 6). In this case, we extrapolate from the questionnaires that the interaction was easy and intuitive. Many users were fascinated that they could interact with virtual object and feeling the weight at the same time.

#### IV. CONCLUSION AND FUTURE WORK

We have integrated different haptic interfaces in an Augmented Reality context. In particular finger-worn devices have been used to render the gravity sensation while a Falcon has been used to render object shape. The system has been tested with an application where a virtual cube is placed over a marker. The user can interact with the system by moving the marker and touching the Augmented reality through the Falcon. Moreover, different weight force directions are rendered according to marker inclination thanks to finger-worn interfaces.

Some problems arise during this work preparation. First of all, the fixed position of the camera allows to better define relation with reference frame but, decreases the user perception of the real environment. Then, fixed finger configuration during marker grasp consents to use a defined Actuation matrix and avoid marker obstruction, but can appear unconformable to users with different hand dimension. Another problem is underactuation. Rendering of force with less motor than the necessary, allows to have more wearable devices but in some case lead to wrong force direction displacement. Finally, we want to point out the difficult on integration of very different devices both from real time and computational aspect.

As future work, we have planned to mount two stereo cameras in the video eyewear to better capture the real world. Moreover, we want to introduce a finger tracking system to allow the dynamic computation of Actuation matrix and

thus, a less constrained grasp position. In the end, we have also planned to introduce more complicate dynamics then the weight force in order to improve the interaction between virtual object and real world.

#### ACKNOWLEDGEMENT

This work has been partially supported by the European Commission with the CP no. 248587, “THE Hand Embodied”, within the FP7-ICT-2009-4-2-1 program “Cognitive Systems and Robotics” and the CP “HANDS.DVI” within the program ECHORD (European Clearing House for Open Robotics Development). The authors are grateful to Dr. Kouta Minamizawa from University of Tokyo.

#### REFERENCES

- [1] G. Bianchi, M. Harders, B. Knoerlein, and G. Szekely. Calibration, registration, and synchronization for high precision augmented reality haptics. *IEEE Trans. on Visualization and Computer Graphics*, 15(1):138–149, 2009.
- [2] V. Buchmann, S. Violich, M. Billinghamurst, and A. Cockburn. FingerARtips: Gesture Based Direct Manipulation in Augmented Reality. In *Proc. GRAPHITE Int. Conf.*, pages 212–221, 2004.
- [3] M. De Pascale and D. Prattichizzo. The Haptik Library, a Component Architecture for Uniform Access to Haptic Devices. *IEEE Robot. and Autom. Mag.*, 14(4):64–75, December 2007.
- [4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.
- [5] NOVINT Falcon Haptic Interface. [Online]: <http://home.novint.com/>.
- [6] S. Kamuro, K. Minamizawa, N. Kawakami, and S. Tachi. Ungrounded Kinesthetic Pen for Haptic Interaction with Virtual Environments. In *Proc. IEEE Int. Symp. on Robot and Human Interactive Communication*, pages 436–441, 2009.
- [7] H. Kato and M. Billinghamurst. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In *Proc. IEEE ACM Int. Workshop on Augmented Reality*, 1999.
- [8] ARToolkit: library for Augmented Reality. [Online]: <http://www.hitl.washington.edu/artoolkit/>.
- [9] Y. Ma, S. Soatto, J. Kosecká, and S.S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Interdisciplinary Applied Mathematics. Springer, 2003.
- [10] K. Minamizawa, S. Fukamachi, N. Kawakami, and S. Tachi. Interactive Representation of Virtual Object in Hand-Held Box by Finger-Worn Haptic Display. In *Proc. IEEE Int. Symp. on Haptic interfaces for virtual environment and teleoperator systems*, pages 367–368, 2008.
- [11] K. Minamizawa, H. Kajimoto, N. Kawakami, and S. Tachi. A Wearable Haptic Display to Present the Gravity Sensation - Preliminary Observations and Device Design. In *Proc. EuroHaptics Int. Conf.*, pages 133–138, 2007.
- [12] D. Prattichizzo and J. Trinkle. Grasping. In B. Siciliano and O. Kathib, editors, *Handbook on Robotics*, pages 671–700. Springer, 2008.
- [13] J. Vallino and C. Brown. Haptics in Augmented Reality. In *Proc. of the IEEE Int. Conf. on Multimedia Computing and Systems*, pages 195–200, 1999.
- [14] S. Walairacht, K. Yamada, S. Hasegawa, Y. Koike, and M. Sato. 4 + 4 Fingers Manipulating Virtual Objects in Mixed-Reality Environment. *Presence: Teleoper. Virtual Environ.*, 2:212–221, 2002.
- [15] C.B Zilles and J.K. Salisbury. A Constraint-Based God-Object Method For Haptic Display. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, volume 3, pages 146–151, 1995.