INVITED PAPER

Representations and Algorithms for Force-Feedback Display

By Miguel A. Otaduy, *Member IEEE*, Carlos Garre, and Ming C. Lin, *Fellow IEEE*

ABSTRACT | "Haptic rendering" or "haptic display" can be broadly defined as conveying information about virtual objects or data to a user through the sense of touch. Among all applications of haptic rendering, force-feedback display of contact interactions with rigid and deformable virtual models through the sense of touch has matured considerably over the last decade. In this paper, we present a general framework for force-feedback display of rigid and virtual environments, and we outline its major building blocks. We focus on computational aspects, and we classify algorithms and representations successfully used in the three major subproblems of force-feedback display: collision detection, dynamics simulation, and constrained optimization. In addition, force-feedback display is an integral part of a multimodal experience, often involving both visual and auditory display; therefore, we also discuss the choice of algorithms and representations for force feedback as a part of multimodal display.

KEYWORDS | Collision detection; contact modeling; force-feedback display; haptic rendering; physically-based simulation

I. INTRODUCTION

Humans explore the surrounding environment predominantly through the visual, auditory, and somatosensory systems. The human somatosensory system can be divided into *proprioception*, which includes the sensation of muscle movement and joint positions, and *touch*, which includes tactile cues to sense temperature, pain, etc., and force feedback, which senses pressure and contact. Force feedback, in turn, can be classified into *cutaneous* cues perceived at the skin and *kinesthesia* perceived at muscles and joints. In his classic paper, "The ultimate display" [1], Sutherland first introduced the ground-breaking vision of a truly multisensory virtual environment, which "should serve as many senses as possible." Sutherland argued that a virtual environment should integrate *touch* in addition to sight and sound.

For more than 40 years, many researchers have been pursuing this vision of the ultimate display, where force feedback is used to identify and manipulate *digital representations* of objects that have no tangible physical presence in a virtual world created with a computer system. By extending the frontier of visual computing, *haptic rendering* or *haptic display* broadly refers to a computational process where a user interacts with a computer (generated) system and experiences a virtual environment through the sense of touch [2]–[6]. More specifically, haptic rendering enables a user to feel, sense, and explore virtual objects or other digital content through a mechanical haptic device. The device displays computer-controlled forces reflecting the user's interaction with virtual objects to provide tangible manifestations of their digital representations.

Despite the early conception of haptic display, it was not until the early 1990s that touch-enabled interfaces and haptic rendering finally began to receive considerable attention as an effective human–computer communication interface. Haptic rendering has since then made broad impact across several domains, such as scientific visualization [7], virtual prototyping [8], nanomanipulation [9], medical training [10]–[12], painting [13]–[15], sculpting [16], [17], music synthesis [18], and digital media control [19].

Fig. 1 illustrates the adoption of force-feedback display on three diverse examples. On the left, the user interacts with a mixed-reality environment that combines real and

Manuscript received May 6, 2012; revised September 4, 2012 and October 30, 2012; accepted January 30, 2013.
M. A. Otaduy is with the Department of Computer Science, Universidad Rey Juan Carlos (URJC Madrid), 28933 Móstoles, Spain (e-mail: miguel.otaduy@urjc.es).
C. Garre is with the Dipartimento di Meccanica, Università della Calabria, 87036 Rende, Italy (e-mail: carlos.garre@gmail.com).
M. C. Lin is with the Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175 USA (e-mail: lin@cs.unc.edu).
Digital Object Identifier: 10.1109/JPROC.2013.2246131

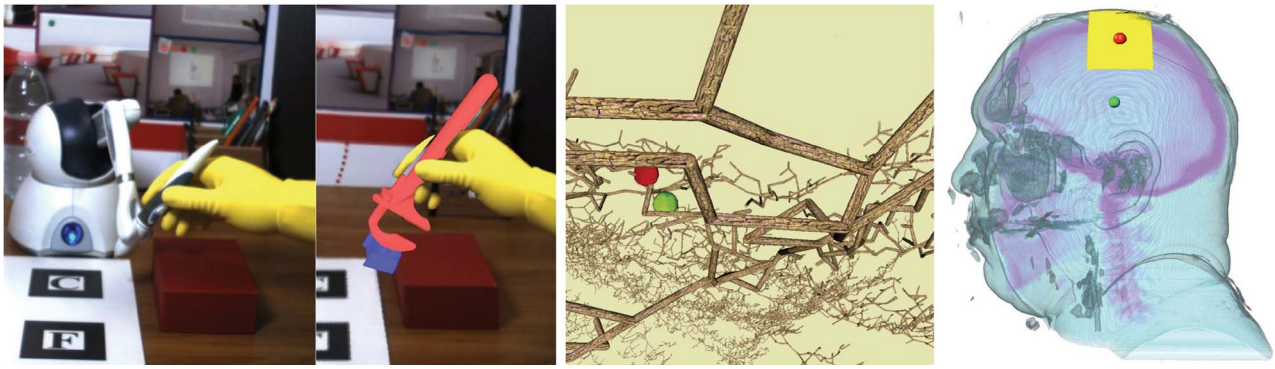

Fig. 1. *Examples of force-feedback display. (Left) Tool-based interaction with mixed-reality environments [20]. (Center) Navigation along filiform neural structures [21]. (Right) Exploration of volumetric CT data [22]. The three examples base the rendering algorithm on the same principle: a probe tracks the haptic interface subject to data constraints, and feedback forces are displayed as a function of the deviation between the probe and the haptic interface.*

virtual objects [20]. In addition, image-space techniques are used to visually remove the haptic device from the scene and seamlessly integrate the user's real hand and the virtual tool. Force feedback makes the interaction more natural, allowing the user to adapt the applied forces to the motion of the objects through fast sensorimotor control. In the center image, a user explores the filiform neural structure of a fly consisting of more than 8000 branches [21]. In this case, force feedback constrains the navigation to filiform structures, enabling the user to easily follow desired paths. On the right, force feedback enhances the exploration of CT data of the visible man's head [22]. Relevant features are easier to identify by constraining the user's motion to isosurfaces in the data.

In all these examples, force-feedback display identifies the regions of interest in the digital content or virtual environment based on the user's interaction; computes contact locations; formulates nonpenetration constraints; computes the resulting feedback forces; and displays these forces through a haptic device. Therefore, a typical force-feedback display algorithm would consist of a fast collision-detection algorithm, robust dynamic simulation of object motion, and efficient constraint-solving techniques.

In the rest of this paper, we first give an overview of the interaction paradigms of haptic rendering. Next, we focus on one area of haptic rendering, the force-feedback display of contact with rigid and deformable virtual models or data. Other areas of haptic rendering, such as cutaneous display of surface properties, are outside the scope of this paper. We present a general framework for force-feedback display, discussing both hardware and software issues, and then we focus on computational aspects of force-feedback display from a multimodal perspective. In Section IV, we describe representations used in each subproblem with the objective of optimizing the efficiency and performance of the algorithms. In Section V, we describe and classify successful algorithms for dynamic simulation, constraint satisfaction, and collision detection in force-feedback display. To conclude, we highlight some ongoing research directions and remaining problems.

II. INTERACTION PARADIGMS

Given the types of force-feedback devices available at the time, earlier force-feedback display algorithms focused mostly on interaction between the haptic probe and a virtual environment via a point contact in 3-D space. This interaction metaphor is commonly referred to as three-degree-of-freedom (3-DoF) haptic rendering. Some of the well-known techniques to model the point probe interaction with rigid objects include the god object [23], virtual proxy [24], and ideal haptic interface point [25]. Other techniques based on the point–object interaction paradigm have also been proposed for 3-DoF haptic rendering of deformable objects [26], 3-DoF haptic visualization of volumetric data [7], [27], and texture rendering [28]–[32]. The early point-based paradigms were extended to ray-based techniques to account for continuous trajectories, both in the interaction with rigid objects [33] and with deformable objects [34].

More recent haptic devices enable the rendering of both forces and torques arising from the interaction between two 3-D objects. Algorithms for rendering such interaction are commonly referred to as 6-DoF haptic rendering. Many different methods have been proposed under this interaction paradigm, varying based on collision-detection algorithms, surface representations, dynamic simulation algorithms, or constraint satisfaction methods. One of the earliest 6-DoF rendering algorithms used a point-shell sampling technique [8], and other methods featured smooth surface representations [35], polygonal collision detection [36], normal cones [37], fast local penetration depth computation [38], multiresolution representations [39], textured surfaces [40], quasi-static

dynamics [41], combination of impulsive and penalty-based response [42], implicit integration for penalty-based contact [43], or constraint-based dynamics [44].

The 6-DoF haptic rendering paradigm has also been extended to account for the interaction between deformable models. Cotin and Delinguette introduced the use of finite element models for haptic rendering of deformable objects [45], and later methods include data-driven, precomputed force models [46], [47], constraint-based dynamics [48], [49], finite element models for object–object interaction [50], time-critical rendering with sphere trees and reduced models based on modal analysis [51], and handle-space force linearization [52].

More recently, hand-like haptic devices have driven the attention to haptic rendering methods of full-hand interaction. Some enable the user to grasp virtual objects with springs for force feedback [53]; others add skinning of a hand model [54], a proxy-type skeleton and two-handed manipulation [55], localized soft finger models [56], or a full-hand model with articulated skeleton and deformable flesh [57].

III. A HAPTIC RENDERING FRAMEWORK

A haptic rendering system offers a bidirectional exchange with the user; therefore, the combined rendering-and-user system can be regarded as a closed-loop system. In practice, time delays in the actuators and/or the haptic rendering algorithm can cause instabilities in the closed-loop rendering-and-user system. The user and the device constitute a continuous-time subsystem, while the digital control and the rendering algorithm constitute a discrete-time subsystem. Therefore, the combined rendering-and-user system falls in the category of *sampled-data systems* [58], and stability analysis of haptic rendering requires the application of control theory for such sampled-data systems.

Computational limitations of the haptic rendering algorithm, together with electromechanical limitations of the device and actuators, lead to a tradeoff in haptic rendering: fidelity must be frequently sacrificed for stability. To guarantee stability, the mechanical impedance rendered by the system must be limited. The concept of Z-width [59] defines the fidelity of a haptic rendering system, i.e., the range of mechanical impedances that it can render in a stable manner. In this section, we present a haptic rendering framework that optimizes the stability–fidelity tradeoff in a simple manner by exploiting modularity. As discussed by Colgate *et al.* [60], a haptic rendering system is guaranteed to be stable if it is composed of discrete-time passive modules. Then, the representations and algorithms of the various subproblems of haptic rendering can be designed with the objective of maximizing fidelity while preserving discrete-time passivity. We do not cover in depth the topic of stability in haptic rendering, and we refer the reader to the work on passivity control, e.g., by Adams and Hannaford [61] and Kim and Hannaford [62]. It is worth mentioning that there are other approaches which increase fidelity, such as rendering of event-based feedback [63], although these are not detailed in this paper.

To start our discussion of the haptic rendering framework, let us specify the mechanical variables exchanged between the actuator control loop and the rendering algorithm. An impedance-based control strategy [5] senses the configuration (i.e., positions and orientations) of the user input and displays contact/interaction forces, while an admittance-based control strategy senses forces applied by the user and controls the configuration of the haptic device. There are also hybrid approaches [61]. Different control strategies are subject to different fidelity limitations. For example, impedance-based control cannot render high stiffness, while admittance-based control cannot render low inertia. We will assume an impedance-based control for the rest of the discussion, although the haptic rendering framework proposed here can be easily modified for other control strategies.

The force-feedback display algorithm interfaces with the haptic device and computes the appropriate output force based on the digital content and the current configuration of the user. The configuration of the user may be sensed by tracking the end effector of the haptic device, whose world-space coordinates are then transformed into the virtual workspace of the digital content. Forces computed in the virtual workspace need to be transformed back to world space. The dimensions of the two workspaces need not match, and intelligent navigation algorithms may be used to make the mismatch transparent to the user [64]–[66]. We denote the configuration of the haptic interface as $\mathbf{q}_h$, and the feedback force as $\mathbf{F}$, both expressed in the reference system of the virtual workspace.

The force-feedback display algorithm for the three examples in Fig. 1 is based on the same principle: a probe $\mathbf{q}_p$ in the virtual workspace tracks the configuration of the haptic interface $\mathbf{q}_h$, but its motion is constrained by the digital content. Feedback forces $\mathbf{F}$ are computed as a function of the deviation between the probe and the haptic interface. In the mixed reality example, the configurations of the end effector of the haptic interface and the virtual tool are clearly depicted, and the tool is constrained by the surfaces of other (virtual or real) objects in the scene. In the other two examples, the haptic interface is shown in green and the probe is shown in red. The probe is constrained to the filiform structure in one case, and to a dynamically extracted isosurface in the other. Constraint-based haptic rendering, pioneered by Zilles and Salisbury [23], provides very intuitive interaction with the digital content in a conceptually simple manner. The user perceives surfaces through the display of normal forces that prevent the user's motion to violate surface constraints, and the haptic exploration is naturally guided by the user's intent.

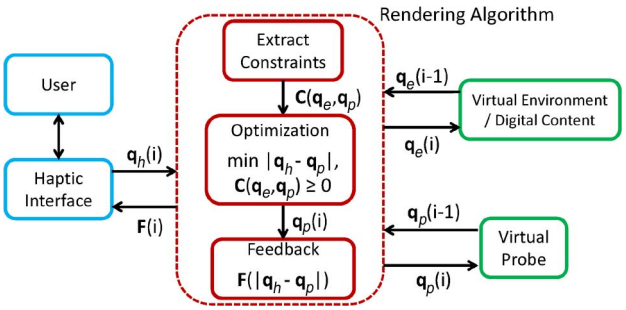

Fig. 2. ***General framework for constraint-based force-feedback display. The environment and the probe are simulated as part of the rendering algorithm, while the haptic interface is handled by the user.***

Fig. 2 illustrates a general algorithm for constraint-based force-feedback display. Given the previous configurations of the probe $\mathbf{q}_p(i-1)$ and the digital environment $\mathbf{q}_e(i-1)$ at time $i-1$, the first step extracts (local) constraints of the environment, formulated as $\mathbf{C}(\mathbf{q}_e, \mathbf{q}_p) \geq 0$. The second step computes the new configuration of the probe $\mathbf{q}_p(i)$ by minimizing the distance to the current configuration of the haptic interface $\mathbf{q}_h(i)$. If the environment is moving, the new configuration of the environment $\mathbf{q}_e(i)$ is computed in this step too. Finally, the third step computes the feedback force $\mathbf{F}(i)$ as a function of the distance between the haptic interface and the probe. The subproblem of constraint definition can be classified in general as *collision detection*, and is discussed in more detail in Section V-C. The subproblem of probe optimization has been formulated in multiple ways, although it can be classified in general as a problem of *contact modeling and resolution*, and is discussed in detail in Section V-B. The solution to this problem depends also on the choice of model for the environment, which is discussed in Section V-A.

As discussed earlier, the framework follows a modular approach to simplify the design of a stable haptic rendering system. Colgate *et al.* [60] described a general approach to ensure stability in haptic rendering which relies on the passivity of the haptic probe simulation. Stability guarantees are then easily enforced through the parameters of a viscoelastic linkage [i.e., a proportional–derivative (PD) controller], referred to as *virtual coupling*, which connects the probe and the haptic interface. The integration of constraint-based force-feedback display with virtual coupling has been widely adopted in diverse settings, including 6-DoF haptic rendering [8], [43], [44], interaction with deformable bodies [49], [51], [52], navigation through filiform data [21], and exploration of volume data [22], [27]. The various algorithms differ in the properties and characteristics of the digital content and the type of optimization methods to compute the configuration of the probe, as described in Section V.

IV. REPRESENTATIONS FOR A MULTIMODAL SIMULATION

In most applications, haptic rendering is part of a richer multimodal experience involving visual (and possibly auditory) rendering of digital content. Therefore, this section provides an overview of some of the key perceptual aspects for visuohaptic rendering and describes suitable representations for the various subproblems to be solved.

A. Multimodal Interaction Principles

Visual rendering requires update rates of approximately 30 Hz. The haptic kinesthetic system has a bandwidth of 20–30 Hz, while the closed-loop human motor control system has a bandwidth below 10 Hz [67]. Such low bandwidths seem to suggest that haptic update rates of approximately 30 Hz are sufficient, but the stability of the combined haptic rendering and controller places more stringent conditions. Tactile sensitivity has a much higher bandwidth than the motor control system, and rendering instabilities can be haptically perceived. The haptic update rate determines the time delay of the haptic rendering process, which effectively limits the contact stiffness that can be rendered in a stable manner [59]. A haptic update rate of 1 kHz is often considered as sufficient for rendering rigid, hard contact, although higher rates have shown to improve the perception of rigidity [68].

The different update rate requirements of visual and haptic rendering suggest separate representations, placing resources where needed. When both visual and haptic cues are present, human perception is dominated by the most salient and/or reliable modality [69]. When contact with an object is perceived both visually and haptically, the visual modality dominates the haptic modality if the object is clearly visible. This observation can be leveraged for the design of efficient representations for visuohaptic rendering. Due to stability conditions and high update rate requirements, haptic representations will often be limited

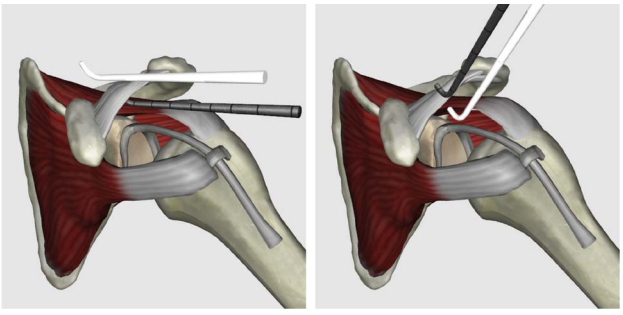

Fig. 3. ***Examples of haptic palpation of a human shoulder model. The probe is constrained by the coracoacromial ligament and deviates from the haptic interface (shown in white), indicating a moderately low haptic stiffness. However, the user experiences the illusion of stiff contact thanks to visual dominance.***

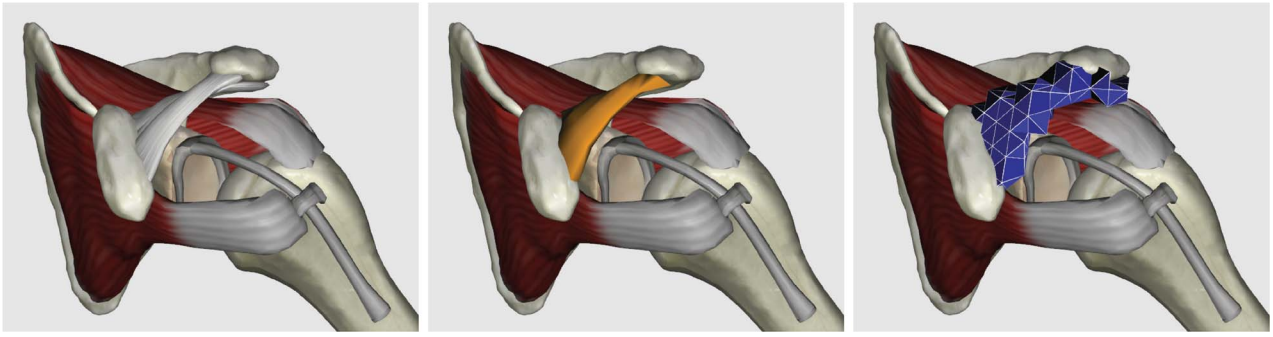

Fig. 4. *Model of a human shoulder for virtual arthroscopy training [70], with three different representations of the coracoacromial ligament for visuohaptic simulation: (left) textured triangle mesh (1632 triangles) for visual rendering; (center) in orange, low-resolution triangle mesh (146 triangles) for collision handling; and (right) in blue, low-resolution tetrahedral finite element mesh (279 tetrahedra) for motion dynamics.*

to moderately stiff objects with low-resolution geometry. Visual rendering, on the other hand, allows the use of stiff objects with high-resolution geometry. As long as contact events are synchronized, users will experience the illusion that the haptically perceived stiffness is as high as the visually perceived one.

In the haptic rendering algorithm illustrated in Fig. 2, the user perceives haptically the stiffness of the viscoelastic virtual coupling between the probe and the haptic interface. This stiffness is limited by stability conditions. On the other hand, the user perceives visually the stiffness of the environment objects touched by the probe. Fig. 3 shows two examples of haptic palpation of a human shoulder model. The probe tracks the haptic interface (whose configuration is shown in white), but is constrained by the coracoacromial ligament. Despite the large deviation between probe and haptic interface, which indicates a moderately low haptic stiffness, the user experiences the illusion of stiff contact, due to visual dominance.

B. Representations for Visual Simulation

A visual simulation of a virtual environment encompasses several important tasks: 1) simulation of the motion of virtual objects; 2) visual rendering; and 3) detection and resolution of collisions among virtual objects. Each of these tasks has different requirements and calls, therefore, for different representations. The motion representation plays a central role, as it serves as input for the other representations.

1) Motion Representation: Motion is described by a vector of generalized coordinates $\mathbf{q}$ and a vector of generalized velocities $\mathbf{v} = d\mathbf{q}/dt$ [71]. The choice of coordinates for an object must reflect its mechanical properties. For a rigid body, it is sufficient to represent the motion with the position of the center of mass and the orientation of a frame of reference attached to the center of mass. Thus, generalized velocities are given by the velocity of the center of mass and the angular velocity of the frame of reference. For deformable bodies, on the other hand, a common option is to discretize the continuum volume with discrete, finite elements, sample the deformation on nodal positions, and then describe the deformation inside the elements through interpolation using appropriate basis functions [72]. The motion representation of a rigid body is compact and allows an efficient computation of dynamics, making it well suited for haptic rendering. For deformable body representations, on the other hand, care must be taken to choose an appropriate resolution of the discretization, balancing computational cost and the accuracy of the deformations [73], [74].

2) Visual Rendering Representation: In spite of a low-resolution motion representation, the overall quality of a simulation can be noticeably enhanced by decoupling the visual rendering representation. The visual representation typically consists of a high-resolution triangle mesh. The 3-D positions of its n vertices can be concatenated in a vector $\mathbf{p}_v \in \mathbb{R}^{3n}$. Then, the positions and velocities of the visual rendering representation can be related to the motion representation in a compact manner as

$$\mathbf{p}_v = \mathbf{f}_v(\mathbf{q}) \qquad \frac{d\mathbf{p}_v}{dt} = \frac{\partial \mathbf{f}_v}{\partial \mathbf{q}}\mathbf{v}. \tag{1}$$

For rigid bodies, function $\mathbf{f}_v$ represents the rigid transformation of each vertex. For deformable bodies whose motion is described with a collection of (tetrahedral) finite elements, $\mathbf{f}_v$ represents barycentric interpolation for each vertex. An example of motion and visual rendering representations for deformable bodies is illustrated in Fig. 4. The figure shows a model of a human shoulder for virtual arthroscopy training [70], with different representations of the coracoacromial ligament. The right image

shows, in blue, the motion representation, consisting of a low-resolution tetrahedral finite element mesh (279 tetrahedra). The left image shows the visual rendering representation, consisting of a textured triangle mesh (1632 triangles). The triangle mesh is fully embedded in the tetrahedral mesh, and the position of each vertex is defined through barycentric interpolation of the nodes of its enclosing tetrahedron.

3) Collision Representation: The definition of contact constraints in the virtual environment requires a surface representation for collision detection. One option is to use the same triangle mesh as for visual rendering. However, since small errors remain imperceptible to the user, a lower resolution triangle mesh is valid, which reduces the computational cost of collision handling. In the same way that we calculated the vertices of the visual representation in (1), the positions of the vertices of the mesh for collision handling can be concatenated in a vector

$$\mathbf{p}_c = \mathbf{f}_c(\mathbf{q}) \qquad \frac{d\mathbf{p}_c}{dt} = \frac{\partial \mathbf{f}_c}{\partial \mathbf{q}}\mathbf{v}. \tag{2}$$

Fig. 4 shows also the collision representation of the coracoacromial ligament. The low-resolution triangle mesh (146 triangles) is embedded in the tetrahedral finite element mesh, and the position of each vertex is defined through barycentric interpolation of the nodes of its enclosing tetrahedron.

A nonpenetration contact constraint for two surface points $\mathbf{p}_a$ and $\mathbf{p}_b$ and surface normal $\mathbf{n}$ can be formulated as $\mathbf{n}^T(\mathbf{p}_a - \mathbf{p}_b) \geq 0$. In terms of the full collision representation, the constraint can be defined more generally as $C(\mathbf{p}_c) \geq 0$. Constraints can be enforced through normal forces, i.e., forces of the form $\mathbf{F} = F\mathbf{n}$. In terms of the full collision representation, they can be expressed as $\mathbf{F} = F(\partial C/\partial \mathbf{p}_c)^T$. However, recall that the collision representation is guided by the motion representation; therefore, constraint forces need to be transmitted to generalized coordinates of the motion representation to become truly effective

$$\mathbf{F} = F\frac{\partial \mathbf{f}_c}{\partial \mathbf{q}}^T \frac{\partial C}{\partial \mathbf{p}_c}^T. \tag{3}$$

With the motion, collision, and visual rendering representations, all adequately modeled and connected, the user experiences the interaction with detailed digital content in a computationally efficient manner. The motion of the objects is described using low-resolution representations, but the user perceives the smooth deformation of high-resolution textured surfaces, and contact is resolved on surfaces that geometrically approximate those ones being displayed.

C. Haptic Representation

According to the constraint-based haptic rendering algorithm presented in Section III, the user perceives the digital content through the viscoelastic virtual coupling. In a straightforward approach with the probe configuration $\mathbf{q}_p$ updated at the visual rendering rate, the user would perceive a low haptic stiffness to ensure stability conditions.

Fortunately, the design of haptic rendering algorithms can exploit perceptual principles discussed in Section IV-A. In particular, thanks to visual dominance, a fine visual representation combined with an approximate haptic representation can result in an overall perceptual experience similar to touching the fine representation. This observation leads to the design of *multirate* haptic rendering algorithms. These algorithms execute two separate computational loops: a visual loop at visual rendering rates employs a detailed representation of the digital content and computes an *intermediate representation* [75], which is evaluated in a haptic loop at haptic rendering rates.

Fig. 5 illustrates a multirate constraint-based haptic rendering algorithm that extends the one in Fig. 2. The visual loop simulates both the environment $\mathbf{q}_e$ and the probe $\mathbf{q}_p$, subject to detailed environment constraints $\mathbf{C}$. In addition, it computes an approximate version of the constraints $\mathbf{C}^*$, which is passed on to the haptic loop. The haptic loop simulates, at a higher update rate, another version of the probe $\mathbf{q}_p^*$, subject to the approximate constraints. In this way, force feedback, computed as a function

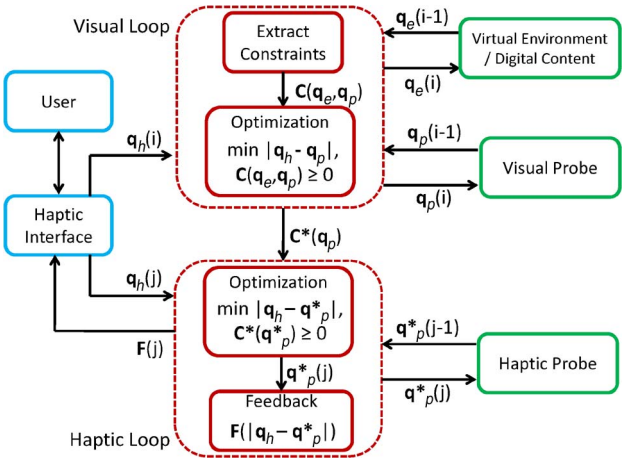

Fig. 5. ***Illustration of a multirate haptic rendering algorithm, with separate visual and haptic representations. The visual loop simulates both the environment and the probe, at a low frequency, and computes a local representation of environment constraints. The haptic loop simulates only the probe, at a high frequency, and computes feedback forces.***

of the separation between the haptic interface and the haptic probe, allows the use of a stiff virtual coupling.

Over the years, various multirate haptic rendering approaches have been proposed. They span approaches that use precomputed responses of deformable bodies [45], use simplified representations of deformable bodies [47], linearize environment forces for rigid bodies [43], use the set of active constraints for rigid bodies [44], identify a rigid handle in the interaction of deformable bodies and then linearize the coupling force [52], or use the set of active constraints for deformable bodies [76].

V. SIMULATION OF THE VIRTUAL ENVIRONMENT

In this section, we first discuss methods to compute the motion of objects in the virtual environment, including the probe. Then we consider various formulations of the probe's configuration as a constrained optimization problem, and we discuss several solutions. We conclude with an overview of collision detection methods for haptic rendering.

A. Motion of the Probe and the Virtual Environment

In the design of a specific haptic rendering algorithm, the first important choice is to model the probe either as a massless object or an object with mass. A massless object moves instantaneously to a target position, whereas an object with mass accelerates and decelerates, may overshoot, or may undergo oscillations. However, modeling a probe with mass may be interesting for several reasons: it could simplify computations, increase stability, or simply represent more realistically the simulated environment.

In the general case, let us consider that both the probe and the rest of the objects in the virtual environment are dynamic, and let their configurations be concatenated in a large vector of generalized coordinates $\mathbf{q}$ (and similarly for their generalized velocities $\mathbf{v}$). Then, the dynamics of the full environment are described by ordinary differential equations (ODEs)

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{F}. \tag{4}$$

$\mathbf{M}$ is the mass matrix of the full system, and $\mathbf{F}$ is a vector of forces that may include gravity, elasticity, viscosity, and inertial forces.

Given the positions and velocities in the previous frame, $\mathbf{q}(i-1)$ and $\mathbf{v}(i-1)$, new positions and velocities are computed through numerical integration of the ODEs. *Implicit integration* methods [77] offer a good tradeoff between stability and computational cost for interactive simulation, and they are typically the methods of choice in haptic rendering. However, these methods are nonlinear, and require the linearization of forces or the use of iterative solvers such as Newton's method. With the backward Euler integration method [77], linearized forces, and a time step Δt, (4) turns into a linear system

$$\mathbf{A}\mathbf{v}(i) = \mathbf{b}$$

with

$$\mathbf{A} = \mathbf{M} - \Delta t \frac{\partial \mathbf{F}}{\partial \mathbf{v}} - \Delta t^2 \frac{\partial \mathbf{F}}{\partial \mathbf{q}}$$

and

$$\mathbf{b} = \Delta t \mathbf{F}(i-1) + \left(\mathbf{M} - \Delta t \frac{\partial \mathbf{F}}{\partial \mathbf{v}}\right)\mathbf{v}(i-1). \tag{5}$$

The complexity of the linear system $\mathbf{A}$ depends on the mechanical properties of the objects in the virtual environment. For rigid bodies, the cost is small and the coordinates of the various bodies can be solved in parallel. For deformable bodies, the *stiffness matrix* $\partial \mathbf{F}/\partial \mathbf{q}$ dominates the cost.

Due to its associated computational cost, modeling of deformable bodies deserves special attention. There are multiple choices, including mass-spring systems or continuum mechanics methods [78], but for haptic rendering, linear corotational elasticity [79] offers a good tradeoff between quality and computational efficiency. The stiffness matrix is then sparse, symmetric, and positive definite, and (5) can be efficiently solved using iterative methods such as conjugate gradient or direct methods such as Cholesky factorization. As discussed in Section IV-B, computational efficiency can be maximized by embedding a high-resolution visual rendering mesh in a low-resolution tetrahedral finite element mesh. To maximize the accuracy of the deformations as well, it is possible to tune nonlinear basis functions based on high-resolution topological features and material inhomogeneities [74]. Modal analysis, which precomputes a small set of global modes that closely approximate typical deformations, is a powerful alternative to linear corotational elasticity thanks to its low constant computational cost [73], [80], [81].

B. Contact Modeling and Resolution

The motion of the probe and the virtual environment is determined by two major factors: 1) their mechanical properties, as seen above; and 2) contact constraints. This second factor is perhaps even more important from the haptic point of view, as it represents the digital data perceived haptically by the user. We will describe two methods to enforce contact constraints, which will lead to four categories of constrained motion problems.

Recall the definition of nonpenetration contact constraints in Section IV-B. In vector form, multiple constraints over the combined probe-and-environment configuration can be expressed as $\mathbf{C}(\mathbf{q}) \geq 0$. The method of *soft constraints* or *penalty-based method* formulates a potential energy as a function of the amount of constraint violation, and then a force as the negative gradient of this energy. With a diagonal matrix $\mathbf{K}$ storing penalty stiffness values, the constraint energy is

$$W = \frac{1}{2}\mathbf{C}^T\mathbf{K}\mathbf{C} \tag{6}$$

and the force is

$$\mathbf{F}_C = -\nabla W = -\nabla\mathbf{C}\mathbf{K}\mathbf{C}. \tag{7}$$

Here, we assume that $\mathbf{C}$ represents only active constraints, i.e., those for which $C_i < 0$. The penalty stiffness must be high to avoid perceptible interpenetrations, and this implies the use of implicit integration and robust collision detection.

The method of hard constraints requires a strict solution to a constrained optimization problem, as it leaves the constraints unchanged. In the context of a Lagrange multiplier approach to optimization [82], it can be formulated as adding a term $\mathbf{C}^T\lambda$ to the function to optimize, where λ represents (unknown) Lagrange multipliers. The constraint forces are, in this case

$$\mathbf{F}_C = \nabla\mathbf{C}\lambda. \tag{8}$$

Hard constraints are strictly enforced, and other contact effects, such as friction, can be added more elegantly. Soft constraints, on the other hand, can be solved with less computational effort.

Depending on the type of constraints and the mass properties of the probe, the optimization problem in haptic rendering can be formulated in one of four ways, as summarized in the following table:

	soft constraints	hard constraints
massless probe	(Quasi-)Static Equilibrium	Distance Minimization
probe with mass	Penalty-Based Dynamics	Constrained Dynamics

1) Static Equilibrium: The configuration of the probe (and the virtual environment, if it moves) is the one at which the penalty forces in (7) and all other forces $\mathbf{F}$ are at equilibrium, i.e.,

$$\mathbf{F} - k\mathbf{C}\nabla\mathbf{C} = 0. \tag{9}$$

The problem can be slightly modified into a quasi-static equilibrium by linearizing the forces. Note also that $\mathbf{F}$ includes the force of the virtual coupling, and possibly other forces such as elasticity of the probe and/or the environment. A key issue for the success of the approach is a good balance between the penalty stiffness and the stiffness of the virtual coupling. The method has been applied to haptic rendering of both rigid [8] and deformable bodies [51].

2) Distance Minimization: The configuration of the probe is defined strictly as the one that minimizes the distance to the haptic interface, subject to contact constraints, i.e.,

$$\min \|\mathbf{q}_p - \mathbf{q}_h\|, \qquad \text{subject to } \mathbf{C}(\mathbf{q}_p) \geq 0. \tag{10}$$

This is the problem described in Section III and solved by Zilles and Salisbury in their god-object approach [23], and is well suited for static environments. Constraint forces are not modeled explicitly, although they may be computed as part of a Lagrange multiplier solution, as discussed before. The distance minimization approach is popular for probes defined as 3-D points or spheres [21], [22], [24]. This approach is less popular for rigid and deformable bodies because of the increased nonlinearity and/or the complexity of the distance function, although some solutions exist [83].

3) Penalty-Based Dynamics: The addition of the penalty forces in (7) to the dynamics ODEs in (4) leaves the configuration of the probe and the environment as the solution to the problem

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{F} - k\mathbf{C}\nabla\mathbf{C}. \tag{11}$$

This is a regular dynamics simulation problem with penalty-based contact forces, and it has been used for haptic rendering of rigid bodies [43]. With implicit integration methods, it involves only minor modifications to the quasi-static equilibrium formulation discussed above.

4) Constrained Dynamics: Contact constraints can be combined with the dynamics formulation of (4) in multiple ways. One approach, described by Baraff and Witkin [84], is to combine them at the acceleration level. On the one hand, constraints $\mathbf{C}(\mathbf{q}) \geq 0$ are differentiated

twice to obtain constraints on accelerations, $d^2\mathbf{C}/dt^2 \approx \nabla\mathbf{C}^T\dot{\mathbf{v}} \geq 0$. On the other hand, the difference between constrained accelerations $\dot{\mathbf{v}}$ and unconstrained accelerations $\mathbf{M}^{-1}\mathbf{F}$ is minimized. The resulting constrained optimization problem can be written as

$$\min \|\dot{\mathbf{v}} - \mathbf{M}^{-1}\mathbf{F}\|, \qquad \text{subject to } \nabla\mathbf{C}^T\dot{\mathbf{v}} \geq 0. \quad (12)$$

Introducing the constraint forces using Lagrange multipliers, the constrained accelerations of the probe and the environment are obtained by solving the equivalent problem

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{F} + \nabla\mathbf{C}\lambda, \qquad \text{subject to } \nabla\mathbf{C}^T\dot{\mathbf{v}} \geq 0. \quad (13)$$

The accelerations are then integrated to obtain the probe and environment configurations. This approach has been followed for haptic rendering of rigid bodies [44], and it can be regarded as a dynamic extension of the original god-object approach [23]. However, it requires additional stabilization to avoid drift.

An alternative approach is to combine contact constraints and the dynamics formulation at the velocity level. On the one hand, the constraints are linearized and integrated (e.g., with backward Euler) to obtain constraints on velocities $\mathbf{C}(i-1) + \Delta t\nabla\mathbf{C}^T\mathbf{v}(i) \geq 0$. On the other hand, the unconstrained accelerations are integrated as in (5) to obtain unconstrained velocities $\mathbf{A}^{-1}\mathbf{b}$, and their difference to constrained velocities $\mathbf{v}$ is minimized. The resulting constrained optimization problem can be written as

$$\min \|\mathbf{v} - \mathbf{A}^{-1}\mathbf{b}\|, \qquad \text{subject to } \mathbf{C}(i-1) + \Delta t\nabla\mathbf{C}^T\mathbf{v}(i) \geq 0. \quad (14)$$

Using the Lagrange multiplier formulation, the constrained velocities of the probe and the environment are obtained by solving the equivalent problem

$$\mathbf{A}\mathbf{v}(i) = \mathbf{b} + \nabla\mathbf{C}\lambda, \qquad \text{subject to } \mathbf{C}(i-1) + \Delta t\nabla\mathbf{C}^T\mathbf{v}(i) \geq 0. \quad (15)$$

This approach has been applied to haptic rendering of both rigid and deformable bodies [49], [52].

A final important point to note is that additional constraints are typically added to both formulations in (13) and (15). Contact forces λ are enforced to be positive (i.e., repulsive), and inactive contact constraints cannot apply force. The resulting constrained problem is called a *complementarity problem* [85]. With high-resolution deformable bodies, solving the complementarity problem often becomes the bottleneck of haptic rendering; therefore, much effort has been devoted to the design of efficient solutions [86], [87].

C. Collision Detection

For haptic rendering, the goal of collision detection is to identify relevant nonpenetration constraints for modeling the local boundaries of objects in contact. Two major approaches have been used for collision detection in haptic rendering, due to their computational efficiency: 1) point shell versus distance field queries; and 2) bounding volume hierarchies for triangle meshes. In both approaches, each contact is described by a point on each object, a distance or penetration depth, and a direction or contact normal.

A distance field stores the distance to the closest point on the surface of an object in a grid, and possibly the distance gradient. For rigid bodies, distance fields may be precomputed, in which case the computation of penetration depth of a point inside a rigid body becomes trivial. When querying for collision between two objects, one of them is represented by a distance field, and the other object is represented by a collection of point samples on its surface, i.e., a *point shell*. Hierarchical storage of the point shell accelerates queries thanks to hierarchical culling [51]. Although mostly used for rigid bodies, distance fields can also be applied to deformable bodies by approximation of finite-element deformations [88], by approximation of modal deformations [51], or by fast recomputation [89].

Bounding volume hierarchies allow hierarchical culling in object–object collision detection queries, quickly narrowing down the search for contact points to the regions in close proximity. Queries between rigid bodies can take advantage of hierarchies based on oriented bounding boxes [90], while queries between deformable bodies are often solved efficiently with hierarchies of axis-aligned bounding boxes, which can be updated in linear time [91]. Bounding volume hierarchies can also be used to answer continuous collision detection queries, which identify the first time of contact [92], [93]. Finally, it is worth noting that hierarchical data structures, either based on bounding volumes [39] or on point shells [51], allow interruptible collision detection with a guaranteed upper bound on the computational cost.

VI. OUTLOOK

As we have surveyed in this paper, force-feedback display of rigid and deformable models has matured considerably in the last decade. For rigid and deformable bodies that retain their topology during haptic interaction, precomputed data structures can greatly accelerate haptic rendering. However, other models with topological changes due to fracturing, cutting, tearing, drilling, or

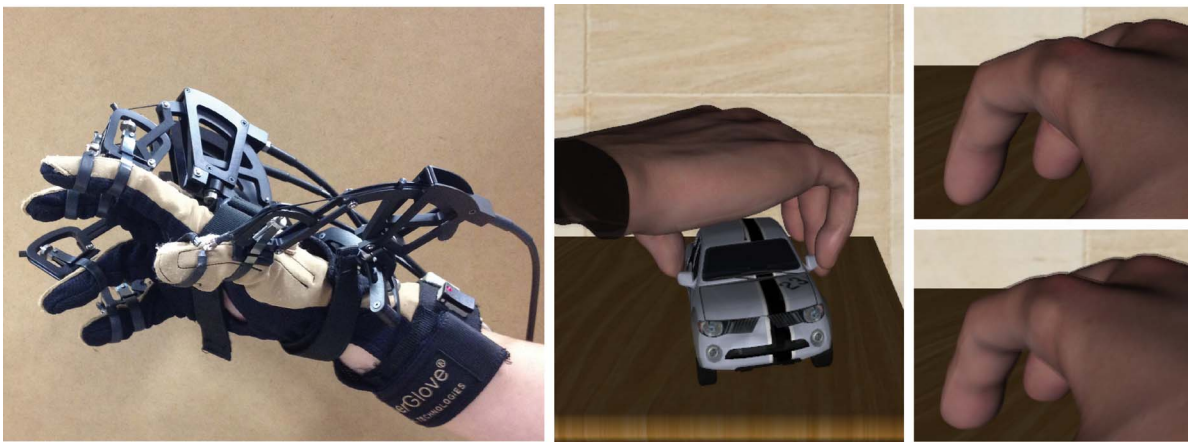

Fig. 6. ***(Left) Cybergrasp exoskeleton device with hand tracking and tactile feedback on the fingertips. (Middle) Interactive hand simulation for haptic rendering of grasping and manipulation tasks [57]. (Right) Detailed fingertip deformations that require combined kinesthetic and cutaneous haptic rendering.***

milling [94], or even fluids [95], can present new computational complexity. These models are of significant interest for many applications, such as planning of surgical procedures, training for routine maintenance, etc.

Overall, achieving high fidelity in haptic rendering is still an unsolved issue, due to the tradeoff with stability. An interesting trend toward increased fidelity is to measure haptic data from the real world and use these data directly in the design of haptic rendering models [96].

Most of the progress in haptic rendering in the last years has targeted kinesthetic feedback from tool-based interaction. Cutaneous feedback from direct-hand interaction, on the other hand, needs still further progress, both on the mechanical and computational ends. Fig. 6 shows an exoskeleton device with tactile feedback on the fingertips, and recent results of interactive hand simulation [57]. Computation of feedback forces for grasping and manipulation tasks like the one shown in the middle image is possible thanks to the simulation of the hand's soft tissue, its articulated skeleton, and their two-way coupling. Haptic rendering of detailed finger deformations like the one in the right image requires combined kinesthetic and cutaneous feedback.

Another topic that deserves more attention is the seamless integration of audio, haptic, and visual rendering. Although the state of the art for sound production still relies on the use of prerecorded sound clips triggered by events, physically-based sound synthesis has been receiving increasing attention, as it can naturally provide less repetitive sound effects that closely correlate with various events and automatically capture the shift of tone and timbre due to object interaction, material variation, and object geometry [97]. Furthermore, physics-based sound synthesis can integrate well with haptic rendering, as they both rely on similar dynamics computations. However, the integration of auditory, visual, and haptic representations to produce consistent and coherent multimodal interaction that minimizes cross-modal disparity has not received sufficient investigation. Recently, Ren *et al.* [98] proposed a three-level representation for both visual and auditory display that minimizes visual–aural disparity caused by texturing, lighting, and shading. Extending such an approach to develop a unified representation for audio, haptic, and visual display could help to enhance cross-modal perception and improve the overall performance of multimodal display. ■

REFERENCES

[1] I. Sutherland, "The ultimate display," in *Proc. Int. Fed. Inf. Process. Congr.*, 1965, pp. 506–508.

[2] J. K. S. , Jr., D. L. Brock, T. Massie, N. Swarup, and C. B. Zilles, "Haptic rendering: Programming touch interaction with virtual objects," in *Proc. Symp. Interactive 3D Graphics*, 1995, pp. 123–130.

[3] G. Burdea, *Force and Touch Feedback for Virtual Reality*. New York, NY, USA: Wiley, 1996.

[4] M. A. Srinivasan and C. Basdogan, "Haptics in virtual environments: Taxonomy, research status, and challenges," *Comput. Graph.*, vol. 21, no. 4, pp. 393–404, 1997.

[5] K. Salisbury, F. Conti, and F. Barbagli, "Haptic rendering: Introductory concepts," *IEEE Comput. Graph. Appl.*, vol. 24, no. 2, pp. 24–32, Mar.-Apr. 2004.

[6] M. C. Lin and M. A. Otaduy, *Haptic Rendering: Foundations, Algorithms, and Applications*. New York, NY, USA: AK Peters, 2008.

[7] R. S. Avila and L. M. Sobierajski, "A haptic interaction method for volume visualization," in *Proc. Visualization*, 1996, pp. 197–204.

[8] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Six degrees-of-freedom haptic rendering using voxel sampling," in *Proc. SIGGRAPH*, Aug. 1999, pp. 401–408.

[9] R. M. Taylor, II, W. Robinett, V. L. Chi, F. P. Brooks, Jr., W. V. Wright, R. S. Williams, and E. J. Snyder, "The nanomanipulator: A virtual reality interface for a scanning tunnelling microscope," in *Proc. SIGGRAPH*, Aug. 1993, pp. 127–134.

[10] M. Agus, A. Giachetti, E. Gobbetti, G. Zanetti, and A. Zorcolo, "Real-time haptic and visual simulation of bone dissection," *Presence*, vol. 12, no. 1, pp. 110–122, 2003.

[11] M. Eriksson, M. Dixon, and J. Wikander, "A haptic VR milling surgery simulator—Using high-resolution CT-data," in *Proc. 14th Med. Meets Virtual Reality (MMVR) Conf.*, 2006, pp. 138–143.

[12] D. Morris, C. Sewell, F. Barbagli, K. Salisbury, N. Blevins, and S. Girod, "Visuohaptic

simulation of bone surgery for training and evaluation," *IEEE Comput. Graph. Appl.*, vol. 26, no. 6, pp. 48–57, Nov.-Dec. 2006.

[13] W. Baxter, V. Scheib, M. Lin, and D. Manocha, "Dab: Interactive haptic painting with 3D virtual brushes," in *Proc. 28th Annu. Conf. Comput. Graph. Interactive Tech.*, 2001, pp. 461–468.

[14] A. Gregory, S. Ehmann, and M. Lin, "Intouch: Interactive multiresolution modeling and 3D painting with a haptic interface," in *Proc. IEEE Virtual Reality Conf.*, 1999, pp. 45–52.

[15] D. Johnson, T. Thompson, T. Thompson, M. Kaplan, D. Nelson, and E. Cohen, "Painting textures with a haptic interface," in *Proc. IEEE Virtual Reality Conf.*, 1999, pp. 282–285.

[16] H. Chen and H. Sun, "Real-time haptic sculpting in virtual volume space," in *Proc. ACM Symp. Virtual Reality Softw. Technol.*, 2002, pp. 81–88.

[17] F. D. H. Qin, A. Kaufman, and J. El-Sana, "Haptic sculpting of dynamic surfaces," in *Proc. Symp. Interactive 3D Graph.*, 1999, pp. 103–110.

[18] C. Cadoz, A. Luciani, J. Florens, and N. Castagné, "ACROE-ICA. Artistic creation and computer interactive multisensory simulation force feedback gesture transducers," in *Proc. Int. Conf. New Interfaces Musical Expression*, 2003, pp. 235–246.

[19] S. Snibbe and K. MacLean, "Haptic techniques for media control," in *Proc. 14th Annu. ACM Symp. User Interface Softw. Technol.*, 2001, pp. 199–208.

[20] F. Cosco, C. Garre, F. Bruno, M. Muzzupappa, and M. A. Otaduy, "Visuo-haptic mixed reality with unobstructed tool-hand integration," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 1, pp. 159–172, Jan. 2013.

[21] L. Raya, M. A. Otaduy, and M. García, "Haptic navigation along filiform neural structures," in *Proc. IEEE World Haptics Conf.*, 2011, pp. 71–76.

[22] L. Corenthy, J. San Martín, M. A. Otaduy, and M. García, "Volume haptic rendering with dynamically extracted isosurface," in *Proc. IEEE Haptics Symp.*, 2012, pp. 133–139.

[23] C. Zilles and K. Salisbury, "A constraint-based god object method for haptics display," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 1995, vol. 3, pp. 146–151.

[24] D. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in *Proc. ACM SIGGRAPH*, 1997, pp. 345–352.

[25] C. Ho, C. Basdogan, and M. Srinivasan, "Efficient point-based rendering techniques for haptic display of virtual objects," *Presence*, vol. 8, no. 5, pp. 477–491, 1999.

[26] C. Mendoza, K. Sundaraj, and C. Laugier, "Faithful force feedback in medical simulators," in *Proc. Int. Symp. Exp. Robot.*, 2002, vol. 8, pp. 414–423.

[27] K. L. Palmerius, A. Ynnerman, and B. Gudmundsson, "Proxy-based haptic feedback from volumetric density data," in *Proc. Eurohaptics Conf.*, 2002, pp. 104–109.

[28] M. Minsky, "Computational haptics: The sandpaper system for synthesizing texture for a force-feedback display," Ph.D. dissertation, Dept. Comput. Sci., Univ. North Carolina Chapel Hill, Chapel Hill, NC, USA, 1995.

[29] S. Choi and H. Z. Tan, "An analysis of perceptual instability during haptic texture rendering," in *Proc. 10th Int. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst.*, 2002, pp. 1261–1268.

[30] J. Siira and D. K. Pai, "Haptic textures: A stochastic approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1996, pp. 557–562.

[31] A. Okamura, J. Dennerlein, and R. Howe, "Vibration feedback models for virtual environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1998, pp. 674–679.

[32] S. Wall and W. Harwin, "Modeling of surface identifying characteristics using Fourier series," in *Proc. ASME Dyn. Syst. Control Div.*, 1999, vol. 67, pp. 65–71.

[33] C. Basdogan, C. Ho, and M. Srinivasan, "A ray-based haptic rendering technique for displaying shape and texture of 3D objects in virtual environments," in *Proc. ASME Winter Annu. Meeting*, 1997, vol. 61, pp. 77–84.

[34] C. Forest, H. Delingette, and N. Ayache, "Surface contact and reaction force models for laparoscopic simulation," in *Proc. Int. Symp. Med. Simul.*, 2004, pp. 168–176.

[35] D. D. Nelson and E. Cohen, "Optimization-based virtual surface contact manipulation at force control rates," in *Proc. IEEE Virtual Reality Conf.*, 2000, pp. 37–44.

[36] A. Gregory, A. Mascarenhas, S. Ehmann, M. Lin, and D. Manocha, "Six degree-of-freedom haptic display of polygonal models," in *Proc. IEEE Visualization*, 2000, pp. 139–146.

[37] D. Johnson and P. Willemsen, "Six degree of freedom haptic rendering of complex polygonal models," in *Proc. Haptics Symp.*, 2003, pp. 229–235.

[38] Y. Kim, M. Otaduy, M. Lin, and D. Manocha, "Six-degree-of-freedom haptic rendering using incremental and localized computations," *Presence*, vol. 12, no. 3, pp. 277–295, 2003.

[39] M. A. Otaduy and M. C. Lin, "Sensation preserving simplification for haptic rendering," in *Proc. ACM SIGGRAPH*, 2003, pp. 543–553.

[40] M. A. Otaduy, N. Jain, A. Sud, and M. C. Lin, "Haptic display of interaction between textured models," *Proc. IEEE Visualization*, pp. 297–304, 2004.

[41] M. Wan and W. A. McNeely, "Quasi-static approximation for 6 degrees-of-freedom haptic rendering," *Proc. IEEE Visualization*, pp. 257–262, 2003.

[42] D. Constantinescu, S. E. Salcudean, and E. A. Croft, "Haptic rendering of rigid contacts using impulsive and penalty forces," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 309–323, Jun. 2005.

[43] M. A. Otaduy and M. C. Lin, "A modular haptic rendering algorithm for stable and transparent 6-DoF manipulation," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 751–762, Aug. 2006.

[44] M. Ortega, S. Redon, and S. Coquillart, "A six-degree-of-freedom god-object method for haptic display of rigid bodies with surface properties," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 3, pp. 458–469, May-Jun. 2007.

[45] S. Cotin and H. Delinguette, "Real-time surgery simulation with haptic feedback using finite elements," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1998, vol. 4, pp. 3739–3744.

[46] M. Mahvash and V. Hayward, "Haptic simulation of a tool in contact with a nonlinear deformable body," in *Proc. Int. Symp. Surgery Simul. Soft Tissue Model.*, 2003, pp. 311–320.

[47] M. Mahvash and V. Hayward, "High-fidelity haptic synthesis of contact with deformable bodies," *IEEE Comput. Graph. Appl.*, vol. 24, no. 2, pp. 48–55, Mar.-Apr. 2004.

[48] C. Duriez, C. Andriot, and A. Kheddar, "Signorini's contact model for deformable objects in haptic simulations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2004, vol. 4, pp. 3232–3237.

[49] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot, "Realistic haptic rendering of interacting deformable objects in virtual environments," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 1, pp. 36–47, Jan.-Feb. 2006.

[50] S. Laycock and A. Day, "Incorporating haptic feedback for the simulation of a deformable tool in a rigid scene," *Comput. Graph.*, vol. 29, pp. 341–351, 2005.

[51] J. Barbič and D. L. James, "Six-DoF haptic rendering of contact between geometrically complex reduced deformable models," *IEEE Trans. Haptics*, vol. 1, no. 1, pp. 39–52, Jan.–Jun. 2008.

[52] C. Garre and M. A. Otaduy, "Haptic rendering of complex deformations through handle-space force linearization," in *Proc. World Haptics Conf.*, 2009, pp. 422–427.

[53] C. W. Borst and A. P. Indugula, "Realistic virtual grasping," in *Proc. IEEE Virtual Reality Conf.*, 2005, pp. 91–98.

[54] X. Han and H. Wan, "A framework for virtual hand haptic interaction," in *Transactions on Edutainment IV*, vol. 4. Berlin, Germany: Springer-Verlag, 2010, pp. 229–240.

[55] R. Ott, F. Vexo, and D. Thalmann, "Two-handed haptic manipulation for CAD and VR applications," *Comput. Aided Design Appl.*, vol. 7, no. 1, pp. 125–138, 2010.

[56] M. Ciocarlie, C. Lackner, and P. Allen, "Soft finger model with adaptive contact geometry for grasping and manipulation tasks," in *Proc. World Haptics Conf.*, 2007, pp. 219–224.

[57] C. Garre, F. Hernández, A. Gracia, and M. A. Otaduy, "Interactive simulation of a deformable hand for haptic rendering," in *Proc. IEEE World Haptics Conf.*, 2011, pp. 239–244.

[58] J. E. Colgate and G. G. Schenkel, "Passivity of a class of sampled-data systems: Application to haptic interfaces," in *Proc. Amer. Control Conf.*, 1994, pp. 3236–3240.

[59] J. E. Colgate and J. M. Brown, "Factors affecting the z-width of a haptic display," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, pp. 3205–3210.

[60] J. E. Colgate, M. C. Stanley, and J. M. Brown, "Issues in the haptic display of tool use," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1995, pp. 140–145.

[61] R. Adams and B. Hannaford, "Stable haptic interaction with virtual environments," *IEEE Trans. Robot. Autom.*, vol. 15, no. 3, pp. 465–474, Jun. 1999.

[62] Y. S. Kim and B. Hannaford, "Some practical issues in time domain passivity control of haptic interfaces," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2001, vol. 3, pp. 1744–1750.

[63] K. J. Kuchenbecker, J. Fiene, and G. Niemeyer, "Improving contact realism through event-based haptic feedback," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 2, pp. 219–230, Mar.-Apr. 2006.

[64] M. A. Otaduy and M. C. Lin, "User-centric viewpoint computation for haptic exploration and manipulation," *Proc. IEEE Visualization*, pp. 311–318, 2001.

[65] L. Dominjon, A. Lecuyer, J. Burkhardt, G. Andrade-Barroso, and S. Richir, "The bubble technique: Interacting with large virtual environments using haptic devices with limitedworkspace," in *Proc. World Haptics Conf.*, 2005, vol. 1, pp. 639–640.

[66] F. Conti and O. Khatib, "Spanning large workspaces using small haptic devices," in *Proc. World Haptics Conf.*, 2005, vol. 1, pp. 183–188.

[67] T. L. Brooks, "Telerobotic response requirements," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 1990, pp. 113–120.

[68] J. Florens, A. Luciani, C. Cadoz, and N. Castagné, "ERGOS: Multi-degrees of freedom and versatile force-feedback panoply," in *Proc. Eurohaptics Conf.*, 2004, pp. 356–360.

[69] M. O. Ernst and M. S. Banks, "Humans integrate visual and haptic information in a statistically optimal fashion," *Nature*, vol. 415, no. 1, pp. 429–433, 2002.

[70] M. A. Otaduy, C. Garre, J. Gascón, E. Miguel, Á. G. Pérez, and J. S. Zurdo, "Modeling and simulation of a human shoulder for interactive medical applications," in *Proc. Spanish Conf. Comput. Graph.*, 2010, pp. 229–237.

[71] A. A. Shabana, *Dynamics of Multibody Systems*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[72] T. J. R. Hugues, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. New York, NY, USA: Dover, 2000.

[73] J. Barbič and D. James, "Real-time subspace integration for St. Venant-Kirchhoff deformable models," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 982–990, Aug. 2005.

[74] M. Nesme, P. G. Kry, L. Jeřábková, and F. Faure, "Preserving topology and elasticity for embedded deformable models," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 52:1–52:9, Jul. 2009.

[75] Y. Adachi, T. Kumano, and K. Ogino, "Intermediate representation for stiff virtual objects," in *Proc. Virtual Reality Annu. Int. Symp.*, 1995, pp. 203–210.

[76] I. Peterlik, M. Nouicer, C. Duriez, S. Cotin, and A. Kheddar, "Constraint-based haptic rendering of multirate compliant mechanisms," *IEEE Trans. Haptics*, vol. 4, no. 3, pp. 175–187, May-Jun. 2011.

[77] D. Baraff and A. P. Witkin, "Large steps in cloth simulation," in *Proc. 25th Annu. Conf. Comput. Graph. Interactive Tech.*, 1998, pp. 43–54.

[78] A. Nealen, M. Müller, R. Keiser, E. Boxermann, and M. Carlson, "Physically based deformable models in computer graphics," *Comput. Graph. Forum*, vol. 25, no. 4, pp. 809–836, Dec. 2006.

[79] M. Müller and M. Gross, "Interactive virtual materials," in *Proc. Graphics Interface*, 2004, pp. 239–246.

[80] C. Basdogan, "Real-time simulation of dynamically deformable finite element models using modal analysis and spectral Lanczos decomposition methods," in *Proc. Med. Meets Virtual Reality Conf.*, 2001, pp. 46–52.

[81] C. Basdogan, C. Ho, and M. Srinivasan, "Virtual environments for medical training: Graphical and haptic simulation of laparoscopic common bile duct exploration," *IEEE/ASME Trans. Mechatron.*, vol. 6, no. 3, pp. 267–285, Sep. 2001.

[82] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer-Verlag, 2006.

[83] X. Zhang, D. Wang, Y. Zhang, and J. Xiao, "Configuration-based optimization for six degree-of-freedom haptic rendering using sphere-trees," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, 2011, pp. 2602–2607.

[84] D. Baraff and A. Witkin, "Physically based modeling," in *SIGGRAPH Course Notes*, 2001.

[85] R. Cottle, J. Pang, and R. Stone, *The Linear Complementarity Problem*. New York, NY, USA: Academic, 1992.

[86] G. Saupin, C. Duriez, S. Cotin, and L. Grisoni, "Efficient contact modeling using compliance warping," in *Proc. Comput. Graph. Int.*, 2008.

[87] M. A. Otaduy, R. Tamstorf, D. Steinemann, and M. Gross, "Implicit contact handling for deformable objects," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 559–568, Apr. 2009.

[88] S. Fisher and M. C. Lin, "Fast penetration depth estimation for elastic bodies using deformed distance fields," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2001, vol. 1, pp. 330–336.

[89] A. Sud, N. K. Govindaraju, R. Gayle, and D. Manocha, "Interactive 3D distance field computation using linear factorization," in *Proc. ACM Symp. Interactive 3D Graph. Games*, 2006, pp. 117–124.

[90] S. Gottschalk, M. Lin, and D. Manocha, "OBB-Tree: A hierarchical structure for rapid interference detection," in *Proc. ACM SIGGRAPH*, 1996, pp. 171–180.

[91] G. V. den Bergen, "Efficient collision detection of complex deformable models using AABB trees," *J. Graph. Tools*, vol. 2, no. 4, pp. 1–14, 1997.

[92] S. Redon, A. Kheddar, and S. Coquillart, "Fast continuous collision detection between rigid bodies," in *Proc. Eurographics*, 2002, pp. 279–287.

[93] M. Tang, S. Curtis, S.-E. Yoon, and D. Manocha, "ICCD: Interactive continuous collision detection between deformable models using connectivity-based culling," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 4, pp. 544–557, Jul.-Aug. 2009.

[94] C. Syllebranque and C. Duriez, "Six degree-of freedom haptic rendering for dental implantology simulation," in *Proc. Int. Symp. Comput. Models Biomed. Simul.*, 2010, pp. 139–149.

[95] G. Cirio, M. Marchal, S. Hillaire, and A. Lécuyer, "Six degrees-of-freedom haptic interaction with fluids," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 11, pp. 1714–1727, Nov. 2011.

[96] R. Höver, M. Di Luca, G. Székely, and M. Harders, "Computationally efficient echniques for data-driven haptic rendering," in *Proc. World Haptics Conf.*, 2009, pp. 39–44.

[97] N. Raghuvanshi and M. C. Lin, "Interactive sound synthesis for large scale environments," in *Proc. Interactive 3D Graph. Symp.*, 2006, pp. 101–108.

[98] Z. Ren, H. Yeh, and M. C. Lin, "Synthesizing contact sounds between textured models," in *Proc. IEEE Virtual Reality Conf.*, 2010, pp. 139–146.

ABOUT THE AUTHORS

Miguel A. Otaduy (Member, IEEE) received the B.S. degree in electrical engineering from Mondragón University, Mondragón, Spain, in 2000 and the M.S. and Ph.D. degrees in computer science from the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, in 2003 and 2004, respectively.

He is an Associate Professor in the Department of Computer Science's Modeling and Virtual Reality Group (GMRV), Universidad Rey Juan Carlos (URJC), Móstoles, Spain. His main research areas are physically based computer animation and haptic rendering. From 2005 to 2008, he was a Research Associate at ETH Zurich, and then he joined URJC Madrid. He has published over 60 papers in computer graphics and haptics.

Prof. Otaduy is the Conference Co-Chair of the 2013 ACM Symposium on Interactive 3D Graphics and Games and Co-Chair of the editorial board of the 2013 IEEE World Haptics Conference. He also co-chaired the program committee of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation in 2010.

Carlos Garre received the B.S. degree in computer science from the Universidad Politécnica de Madrid, Madrid, Spain, in 2006 and the M.S. degree in computer graphics and virtual reality from the Universidad Rey Juan Carlos (URJC), Madrid, Spain, in 2008.

From 2006 to 2012, he was a Teaching Assistant and Ph.D. candidate in the Department of Computer Science's Modeling and Virtual Reality Group (GMRV), URJC. He has completed his PhD and is waiting to defend. He is currently a FP7 Marie Curie Experienced Researcher in the Department of Mechanic Engineering at the Università della Calabria, Arcavacata di Rende, Italy. His main research areas are haptic rendering, physically based computer animation, and real-time systems.

Ming C. Lin (Fellow, IEEE) received the Ph.D. degree in electrical engineering and computer science from the University of California Berkeley, Berkeley, CA, USA.

She is currently John R. & Louise S. Parker Distinguished Professor of Computer Science at the University of North Carolina (UNC) at Chapel Hill, Chapel Hill, NC, USA. She has (co)authored more than 240 refereed publications, and coedited/authored four books. Her research interests include physically-based modeling, haptics, real-time 3-D graphics for virtual environments, robotics, and geometric computing.

Dr. Lin received several honors and awards, including 2010 IEEE VGTC Technical Achievement Award in Virtual Reality, and eight best paper awards at international conferences on computer graphics and virtual reality. She is a Fellow of the Association for Computing Machinery (ACM). She has served on over 120 program committees of leading conferences on virtual reality, computer graphics, robotics, haptics, and computational geometry; and cochaired more than 25 international conferences and workshops. She is the Editor-in-Chief of the IEEE Transactions on Visualization and Computer Graphics, a member of several editorial boards, and a guest editor for many special issues of scientific journals and technical magazines. She has also served on several steering committees and advisory boards of international conferences, as well as technical advisory committees constituted by government agencies and industry.