Optimization-Based Wearable Tactile Rendering

Alvaro G. Perez*   Daniel Lobo*   Francesco Chinello   Gabriel Cirio
Monica Malvezzi   José San Martín   Domenico Prattichizzo   Miguel A. Otaduy

Abstract—Novel wearable tactile interfaces offer the possibility to simulate tactile interactions with virtual environments directly on our skin. But, unlike kinesthetic interfaces, for which haptic rendering is a well explored problem, they pose new questions about the formulation of the rendering problem. In this work, we propose a formulation of tactile rendering as an optimization problem, which is general for a large family of tactile interfaces. Based on an accurate simulation of contact between a finger model and the virtual environment, we pose tactile rendering as the optimization of the device configuration, such that the contact surface between the device and the actual finger matches as close as possible the contact surface in the virtual environment. We describe the optimization formulation in general terms, and we also demonstrate its implementation on a thimble-like wearable device. We validate the tactile rendering formulation by analyzing its force error, and we show that it outperforms other approaches.

Index Terms—Tactile rendering, wearable haptics, soft skin, virtual environments.

✦

1 INTRODUCTION

HAPTIC rendering stands for the process by which desired sensory stimuli are imposed on the user in order to convey haptic information about a virtual object [1]. Haptic rendering has been implemented mostly using kinesthetic devices, where the problem can be formulated as the simulation of a tool object in contact with other environment objects, and feedback is displayed by either commanding the configuration of this tool object to the device (in admittance display), or by computing coupling forces between the tool object and the device (in impedance display) [2].

In recent years we have witnessed the advent of multiple cutaneous haptic devices, using a variety of stimuli to convey haptic information (vibrotactile feedback, local contact surface modulation, skin stretch, or even ultrasound feedback). Currently, haptic rendering of virtual environments is mostly limited to tool-based interaction, but the progress on cutaneous devices opens the door to direct hand interaction too. Moreover, cutaneous feedback, which operates with smaller forces than kinesthetic feedback, does not need to be grounded on an external support, and can therefore be wearable. As the hardware technology becomes available, the question then arises: *How should haptic rendering be formulated for cutaneous devices?*

In this work, we propose a formulation of tactile rendering as an optimization problem. Given a simulation of virtual contact between a model of the user's skin and a virtual environment, we formulate the control of a tactile interface as the problem of maximizing the similarity of contact between the user's real skin and the tactile interface. This paper is an extended version of a previously published paper [3], which proposed an optimization-based tactile rendering algorithm for a large family of wearable cutaneous devices that stimulate the skin through local contact
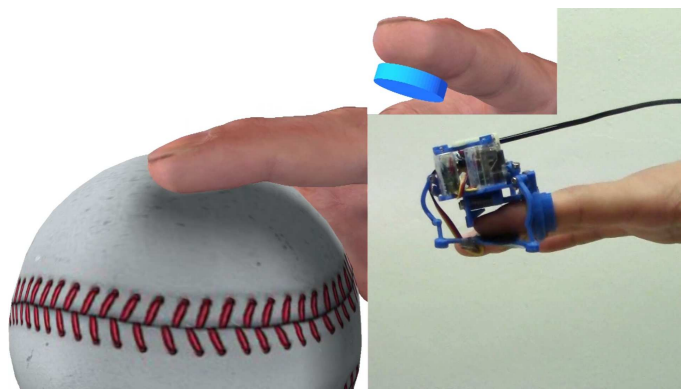

Fig. 1. Example of tactile rendering during the exploration of a ball. The image on the left shows virtual contact between the soft finger model and the ball. Based on the colliding finger points, our optimization-based algorithm computes the optimal device configuration, shown on the right, such that the contact surface displayed to the user is as similar as possible to the virtual contact surface. The inset shows a virtual representation of the optimal device configuration in the local reference of the finger, simulating the deformation produced by the device in contact with the finger.

surface modulation (LCSM). The rendering algorithm was based on the principle of *contact surface matching*, i.e., minimizing the deviation between the contact surface in the virtual environment and the contact surface rendered by the device. In this paper, we augment optimization-based tactile rendering to account for workspace limits of the devices, turning the formulation into a constrained optimization. We also support a larger set of devices, both parallel and open-chain mechanisms.

As we summarize in Section 3, as a first step we follow a strategy similar to tool-based kinesthetic rendering algorithms: we simulate the interaction between a model of the user's skin and the virtual environment. For optimal estimation of the contact surface with the virtual environment, we simulate the skin using a nonlinear model [4].

As a second step, we formulate the computation of the device configuration as an optimization problem, minimizing the contact

---

• *A. G. Perez and D. Lobo contributed equally to this work and should be considered joint first authors.*
• *A. G. Perez, D. Lobo, G. Cirio, J. San Martín, and M. A. Otaduy are with the Department of Computer Science, Universidad Rey Juan Carlos, Madrid, Spain.*
  Contact: see http://mslab.es
• *F. Chinello, M. Malvezzi, and D. Prattichizzo are with the University of Siena, Italy, and the Istituto Italiano di Tecnologia, Genoa, Italy.*

surface deviation between the virtual environment and the actual device. In Section 4, we formulate tactile rendering in general terms as a constrained optimization, both for open-chain and parallel mechanisms, and accounting for device workspace constraints.

We demonstrate the application of our tactile rendering algorithm on a wearable thimble-like device [5]. In Section 5 we discuss specifics of the implementation of the rendering algorithm for this device.

We have tested our rendering algorithm on a variety of contact configurations, such as the exploration of a ball shown in Fig. 1. Most importantly, we have analyzed the error between the contact forces in the virtual environment and the forces produced by our tactile rendering algorithm. We have compared this force error for several methods, and we demonstrate that the constrained optimization formulation outperforms our earlier unconstrained optimization, as well as device-specific heuristic approaches.

2 RELATED WORK

As of today, there is no standardized skin stimulation method for cutaneous haptic rendering. Vibratory feedback is one stimulation method that has been successfully used for conveying information through the tactile sensory channel. The most common example nowadays is the use of vibrotactile displays [6], but vibratory feedback has also been integrated in wearable devices, e.g., on the user's back [7], using an arm suit [8], on the foot [9], or as a bracelet [10].

The stimulation method we adopt in our work can be referred to as local contact surface modulation or LCSM. It consists of displaying a virtual object by imposing on the skin a contact surface that approximates the one of the virtual object. LCSM can be achieved using pin arrays [11], [12], [13], a mobile platform located under the finger pad [5], [14], [15], or using a flexible membrane to control the ratio between contact force and contact area [16]. Dostmohamed and Hayward [17] studied the perception of shape by controlling the trajectory of the contact region, while Frisoli et al. [18] studied the effect of cutaneous feedback on the perception of contact surface orientation.

LCSM can be considered an extension of contact location display. Provancher et al. [19] designed a device that controls the position of a tactile element under the user's finger pad, and they demonstrated the ability to discriminate surface curvature as well as moving objects. Later, they extended the device to control both tangential skin stretch and normal contact force [20], and they also designed a rendering algorithm to faithfully account for edge sharpness in the optimization of contact location [21].

Skin stretch is yet another possible stimulation method. A precursor for this type of stimulation method was to modulate slip between the finger pad and a rotating object [22]. Other example implementations include the application of distributed and modulated local stretch at high frequencies to simulate texture exploration [23], applying stretch with a strap on the finger pad [24], 2D tangential displacement of the finger pad [25], [26], stretch of the finger pad skin with 3 degrees of freedom [27], or fabric-based bracelets [28].

Finally, a recent alternative is the use of air vortices or ultrasound for mid-air cutaneous stimulation [29], [30].

For kinesthetic rendering, two decades of research have led to an accepted algorithm standard: a tool object is simulated subject to contact constraints with the virtual environment, and forces are rendered as a function of the deviation between the constrained tool and the configuration of the haptic device [2], [31], [32], [33], [34].

For cutaneous rendering, on the other hand, algorithmic research is scarce. In the case of data exploration and interaction on tactile displays, there are thorough rendering methods both for vibrotactile feedback [35] and for friction modulation using electrovibration [36]. In the case of LCSM, research on hardware aspects has typically been accompanied by proof-of-concept demonstrations not capable of rendering arbitrary contact. The thimble-like device presented by Prattichizzo et al. [15] modulates contact area by pressing and orienting a small mobile platform. But this device also supports force rendering, by controlling the force exerted by the platform on the finger pad, which allows the use of typical kinesthetic rendering algorithms. To date, the common approach to cutaneous rendering is to design a simplified contact model for each finger pad, compute a single force (and possibly torque) per finger pad, and display this to the user. The existing simplified finger contact models include: a non-penetrating frictional point [37], a point contact with frictional moments [38], or one-dimensional deformation models [39]. These models ignore the high-resolution mechanoreceptor density of finger skin and largely oversimplify the complex force fields perceivable by the finger pad into a single force.

Cutaneous rendering enjoys an important advantage over kinesthetic rendering. Without kinesthetic feedback, the haptic loop is intrinsically passive [40]. As a result, stability of cutaneous rendering does not impose impedance or update rate restrictions.

This paper constitutes an extended version of a previous conference work [3]. Here, we extend this previous work in multiple ways: we outline the optimization formulation for both open-chain and parallel mechanisms, we incorporate device workspace constraints thanks to a constrained optimization formulation, we discuss implementation details for a type of LCSM device, and we compare the accuracy of our method to other approaches.

3 TACTILE RENDERING OVERVIEW

In our context, tactile rendering consists of defining control commands for a tactile device, such that the user perceives forces and positions that simulate contact with a virtual environment. We do this following a model-based control approach. We track the position and orientation of the user's finger, and we use them to guide the simulation of a virtual model of the finger in the virtual environment. We compute contact information (i.e., forces and deformations) for the surface of the finger pad model, and we use this information to compute a configuration of the tactile device that produces the best-matching contact on the user's real finger pad.

In this work, we formulate the computation of the device configuration as a *contact surface matching* optimization problem. We optimize the geometry of contact with the user's finger pad, not contact forces. With our approach, optimization of contact geometry is computationally less expensive than optimization of contact forces, but it is best suited for interaction with rigid or stiff virtual objects, not with soft virtual objects.

Fig. 2 depicts the elements involved in the optimization problem. Without loss of generality, let us assume that contact takes place between a finger model $\mathbb{F}$ and a virtual object $\mathbb{O}$. At every simulation step, we identify the contact surface $\mathbb{S}_{\mathbb{O}}$ between $\mathbb{F}$ and $\mathbb{O}$. Using the tactile device, we will try to produce a contact surface $\mathbb{S}_{\mathbb{D}}$ between the device $\mathbb{D}$ and the real finger, such that both contact
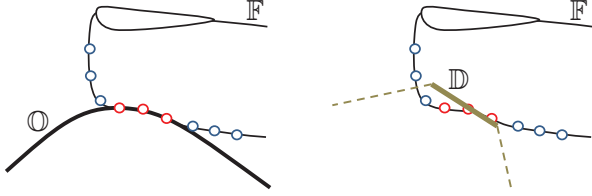
Fig. 2. Schematic depiction of Contact Surface Matching. Left: Contact between a finger model $\mathbb{F}$ and a virtual object $\mathbb{O}$ produces a set of points in contact $\mathbb{C}$, shown in red, and a set of points not in contact $\mathbb{N}$, in blue. Right: Contact Surface Matching aims to optimize the configuration of the device $\mathbb{D}$ such that the sets of points in contact and not in contact are preserved. The figure shows an unoptimzed device configuration. To compute signed distances for points not in contact, we extend the device as a 90-degree truncated cone (shown as dotted lines).

surfaces are as similar as possible, i.e., $\min \| \mathbb{S}_\mathbb{O} - \mathbb{S}_\mathbb{D} \|$ under an appropriate similarity metric. In Section 4 we describe our contact surface matching optimization algorithm in detail.

To estimate the contact surface $\mathbb{S}_\mathbb{D}$ between the device $\mathbb{D}$ and the real finger, we actually compute the contact surface between the device and the finger model $\mathbb{F}$. Therefore, the accuracy of our model-based control approach depends to a large extent on the accuracy of the finger model. As the device $\mathbb{D}$ moves against the user's actual finger, the surface of the skin will change. Therefore, to compute a correct surface matching, the simulation of contact between the finger model $\mathbb{F}$, the virtual object $\mathbb{O}$, and the device must be as realistic as possible, and must predict how the surface of the real finger will be affected by contact.

We simulate the skin using a strain-limiting deformation model [4], which is capable of reproducing the extreme non-linearities in human skin, solved efficiently with a nonlinear constrained dynamics solver [41]. At low forces, we compute deformations using a regular linear corotational finite element model (FEM) [42]. With a low Young modulus the finger pad of $\mathbb{F}$ deforms even with low forces, hence replicating the behavior of true skin. At high forces, we augment the linear corotational FEM formulation with strain-limiting constraints. Constraints are defined on the principal components of the deformation gradient, and they are activated locally on each element of the FEM model when its deformation exceeds a certain value. In this way, parts of the skin that reach the deformation limit start acting rigidly. The deformation of the finger pad of $\mathbb{F}$ saturates at high forces. This nonlinear model can be tuned for each particular user, with an error of less than 17% in its force-area response [43].

To couple the skin simulation to the user's motion, we follow the same overall architecture as in [44]. For the case of a finger, we track the motion of the user's finger in the real world, set a viscoelastic coupling between the tracked configuration and a simulated rigid body in the virtual world, and set stiff spring connections between this simulated rigid body and the nodes of the FEM model of the skin. As a result, when the user moves the finger, the motion is transmitted to the FEM model $\mathbb{F}$. When the simulated finger is constrained by contact, the user may continue moving the real finger in an unconstrained manner, due to the lack of kinesthetic feedback. However, no matter how large the coupling force is, the deformation limits of the finger model ensure that the deformation of the finger, and hence tactile rendering, remains valid.

4 CONTACT SURFACE MATCHING

The major novelty in our work is the formulation of tactile rendering as a constrained optimization problem on the configuration of the device. In this section, we describe in detail this optimization problem. We start with a generic description of the optimization formulation, discussing differences between open-chain and parallel mechanisms, and introducing device workspace limits as constraints. Then we formulate a contact surface deviation metric, which forms the core of contact surface matching as an optimization problem. And we conclude by discussing the solver for the optimization problem and additional required computations.

4.1 Open-Chain Vs. Parallel Mechanisms

The formulation of contact surface matching differs slightly depending on the type of kinematic structure of the tactile device. Here, we consider two broad types of devices, those built using an open-chain mechanism, and those built using a parallel mechanism. For these two types, the natural search space of the optimization algorithm is different, to account for the kinematics functions that can be expressed in closed-form and those that cannot.

Let us define the actuator coordinates of the device as $\mathbf{q}$, and the end-effector coordinates as $\mathbf{w}$. For an open-chain mechanism, we can express in closed-form the forward kinematics $\mathbf{w}(\mathbf{q})$. For a parallel mechanism, instead, we can express in closed-form the inverse kinematics $\mathbf{q}(\mathbf{w})$.

A LCSM device defines a surface geometry $\mathbb{D}$, which is a direct outcome of the end-effector coordinates, i.e., $\mathbb{D}(\mathbf{w})$. Contact surface matching can be expressed as the minimization of some objective function f that depends on the device geometry $\mathbb{D}$. But the search for the optimal device configuration should account for the workspace constraints of the device, which can be expressed in terms of the actuator coordinates as $\mathbf{C}(\mathbf{q}) \geq 0$. Then, putting it all together, contact surface matching is expressed as a constrained optimization problem.

For an open-chain mechanism, we exploit the closed-form expression of forward kinematics, and compute optimal actuator coordinates $\mathbf{q}^*$ as the solution to the following constrained optimization problem:

$$\mathbf{q}^* = \arg \min f(\mathbb{D}(\mathbf{w}(\mathbf{q}))), \quad \text{s.t.} \quad \mathbf{C}(\mathbf{q}) \geq 0. \tag{1}$$

For a parallel mechanism, we exploit the closed-form expression of inverse kinematics, and compute optimal end-effector coordinates $\mathbf{w}^*$ as the solution to the following constrained optimization problem:

$$\mathbf{w}^* = \arg \min f(\mathbb{D}(\mathbf{w})), \quad \text{s.t.} \quad \mathbf{C}(\mathbf{q}(\mathbf{w})) \geq 0. \tag{2}$$

And then we compute the optimal actuator coordinates $\mathbf{q}^*$ using the inverse kinematics.

4.2 Definition of the Objective Function

Conceptually, given the surface of the virtual object $\mathbb{O}$ and the surface of the device $\mathbb{D}$, we want the contact surface between the finger model $\mathbb{F}$ and these two surfaces to be the same, i.e., $\mathbb{S}_\mathbb{O} = \mathbb{S}_\mathbb{D}$. In other words, the points in contact in both surfaces should be the same, and the points not in contact should also be the same. Points in contact between the finger $\mathbb{F}$ and the virtual object $\mathbb{O}$ have zero distance, and we wish the same points to
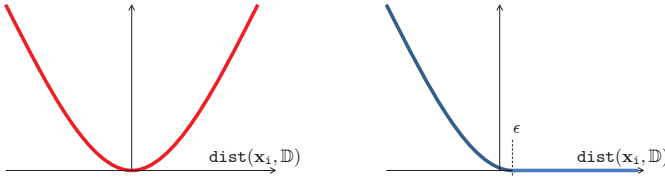
Fig. 3. The cost functions are different for points in contact or not in contact. For points in contact (left), we penalize equally the distance to the device. For points not in contact (right), we penalize only those that penetrate the device (i.e., with negative distance).

have zero distance between the finger $\mathbb{F}$ and the device $\mathbb{D}$. But for points not in contact between the finger model and the virtual object, we simply want them to have positive distance between the finger model and the device (where negative distance means that the points of the finger penetrate the device); in this case the values of distances do not need to match. Our surface matching descriptor is more relaxed than surface-to-surface distance metrics (e.g., Hausdorff distance). But, at the same time, it ensures that both points in contact and points not in contact are accounted for when determining the deviation of contact surfaces.

We formalize the contact surface deviation in the following way. Given a set of sample points $\{\mathbf{x}_i\}$ on the surface of the finger model $\mathbb{F}$, we split them into a set $\mathbb{C}_{\mathbb{O}}$ of points in contact with the virtual object $\mathbb{O}$, and a set $\mathbb{N}_{\mathbb{O}}$ of points not in contact. This information is provided by the skin contact simulation described in Section 3. For points in contact, $i \in \mathbb{C}_{\mathbb{O}}$, we wish their distance to the device $\mathbb{D}$ to be zero. To favor this fact, we design a quadratic cost function as shown in Fig. 3-left. For points not in contact, $i \in \mathbb{N}_{\mathbb{O}}$, we wish their distance to the device $\mathbb{D}$ to be positive. To favor this fact, we design an asymmetric cost function as shown in Fig. 3-right. In practice, we want the distance of points not in contact to be larger than a small tolerance ε. Then, let us define the set $\mathbb{C}_{\mathbb{D}}$ of points in contact with the device as those sample points on the finger model's surface that are closer than a distance ε from the device.

Altogether, we define the objective function of contact surface matching as the following contact surface deviation metric. It adds up two terms that use different distance functions: one for points in contact with the virtual object, and another one for points not in contact with the virtual object but in contact with the device:

$$f = \sum_{i \in \mathbb{C}_{\mathbb{O}}} \operatorname{dist}(\mathbf{x}_i, \mathbb{D})^2 + \sum_{i \in \mathbb{N}_{\mathbb{O}} \cap \mathbb{C}_{\mathbb{D}}} \left(\operatorname{dist}(\mathbf{x}_i, \mathbb{D}) - \varepsilon \right)^2. \tag{3}$$

This objective function is minimized for actuator coordinates following Eq. (1) in case of open-chain mechanisms, or it is minimized for end-effector coordinates following Eq. (2) in case of parallel mechanisms. In Section 5 we describe the objective function in more detail for the particular type of LCSM device used in our experiments.

The evaluation of distances between device $\mathbb{D}$ and finger model $\mathbb{F}$ in Eq. (3) should use an accurate model of the finger skin, which deforms accurately according to the configuration of the device. But computing this deformation as part of the optimization process would not be computationally feasible. Instead, we exploit the same skin simulation we use to compute the contact surface $\mathbb{S}_{\mathbb{O}}$ with the virtual object. If the device succeeds to produce a similar contact, we can safely assume that the real finger will be deformed similar to the simulated finger $\mathbb{F}$. Based on this observation, on every rendering frame we take the deformed finger model $\mathbb{F}$, and use this deformed finger to compute distances to the device model.

The objective function in Eq. (3) could include a temporal smoothing term to eliminate possible jitter and alleviate the presence of local minima. However, in our implementation we have not added such a term to focus the evaluation of results on raw contact surface matching.

4.3 Optimization Algorithm

We have explored several gradient-based methods to solve the constrained optimization problems in Eq. (1) and Eq. (2). In practice, we have obtained good performance using the SLSQP sequential quadratic programming routine in NLopt [45]. This routine requires the computation of gradients of the objective function and the constraints.

Let us consider the constrained optimization problem in Eq. (2) for parallel mechanisms; the formulation is similar for open-chain mechanisms. Then, the gradient of the objective function from Eq. (3) w.r.t. end-effector coordinates can be expressed in general terms as:

$$\frac{\partial f}{\partial \mathbf{w}} = 2 \sum_{i \in \mathbb{C}_{\mathbb{O}}} \operatorname{dist}(\mathbf{x}_i, \mathbb{D}) \frac{\partial \operatorname{dist}(\mathbf{x}_i, \mathbb{D})}{\partial \mathbb{D}} \frac{\partial \mathbb{D}}{\partial \mathbf{w}} \tag{4}$$
$$+ 2 \sum_{i \in \mathbb{N}_{\mathbb{O}} \cap \mathbb{C}_{\mathbb{D}}} (\operatorname{dist}(\mathbf{x}_i, \mathbb{D}) - \varepsilon) \frac{\partial \operatorname{dist}(\mathbf{x}_i, \mathbb{D})}{\partial \mathbb{D}} \frac{\partial \mathbb{D}}{\partial \mathbf{w}}.$$

Note that this gradient adds up two terms: one for points in contact with the virtual object, and another one for points not in contact with the virtual object but in contact with the device.

And the gradient of the workspace constraints w.r.t. end-effector coordinates can be expressed as:

$$\frac{\partial \mathbf{C}}{\partial \mathbf{w}} = \frac{\partial \mathbf{C}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \mathbf{w}}. \tag{5}$$

Given a parameterization of the surface of the device, $\mathbb{D}$, the computation of gradients makes use of four derivative terms: the derivative of the distance function w.r.t. to the parameterization of $\mathbb{D}$, $\frac{\partial \operatorname{dist}(\mathbf{x}_i, \mathbb{D})}{\partial \mathbb{D}}$; the derivative of this parameterization w.r.t. end-effector coordinates, $\frac{\partial \mathbb{D}}{\partial \mathbf{w}}$; the derivative of workspace constraints w.r.t. actuator coordinates, $\frac{\partial \mathbf{C}}{\partial \mathbf{q}}$; and the derivative of inverse kinematics $\frac{\partial \mathbf{q}}{\partial \mathbf{w}}$. Of course, all these derivatives are specific to each LCSM device. If the optimization method reaches a singular configuration of the device (i.e., a singular Jacobian of inverse kinematics $\frac{\partial \mathbf{q}}{\partial \mathbf{w}}$ for a parallel mechanism or a singular Jacobian of forward kinematics $\frac{\partial \mathbf{w}}{\partial \mathbf{q}}$ for an open-chain mechanism), a small regularization can be added to the solver. The test device used in our examples does not exhibit singular configurations within its workspace.

5 RENDERING WITH A WEARABLE THIMBLE

We have implemented our general tactile rendering algorithm on the robotic wearable thimble shown in Fig. 4. In this section, we first provide a description of the main characteristics of the device. Then, we describe the specific details for the implementation of the optimization algorithm, namely the computation of contact distances as a function of end-effector coordinates and the computation of inverse kinematics.
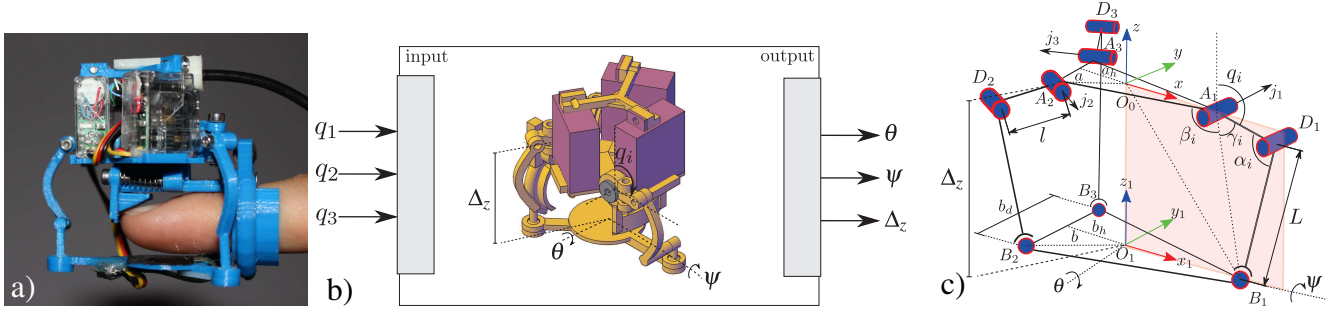
Fig. 4. Thimble-type device used in our experiments. From left to right: (a) actual device, worn by a user; (b) schematic drawing of the device; and (c) variables and dimensions used in the kinematics analysis. The device is wearable, with a fixed platform mounted on the nail and a mobile disk-like platform in contact with the finger pad. The parallel structure is controlled through three joint angles (q_1, q_2, q_3), which yield two rotational DoFs (pitch θ and roll ψ) and one translational DoF (normal translation Δz), which in turn determine the contact surface exposed to the finger pad.

5.1 The Device

We use the thimble-like cutaneous device designed by Chinello et al. [5], shown in Fig. 4-a. It is composed of a fixed and a mobile part. The fixed part is grounded on the middle phalanx of the index finger, on the nail side, and holds three servomotors. The joint angles of these servomotors constitute the actuator coordinates in our formulation, $\mathbf{q} = (q_1, q_2, q_3)$. The fixed and mobile parts are connected using three limbs with an RRS (Revolute-Revolute-Spherical) structure [46], which leads to a parallel mechanism with two angular DoFs (pitch θ and roll ψ) and one translational DoF (a displacement Δz), shown in Fig. 4-b. These constitute the end-effector coordinates in our formulation, i.e., $\mathbf{w} = (\theta, \psi, \Delta z)$. The mobile part is formed by a disk-shaped platform placed under the finger pad, and its motion exposes a locally controllable surface to the finger pad. We parameterize this disk-shaped platform using the center of its surface $\mathbf{p}$ and its unit normal $\mathbf{n}$, i.e., $\mathbb{D} = (\mathbf{p}, \mathbf{n})$.

The device is actuated using three servomotors with good stall torque and position control capabilities. When all three servomotors are actuated in the same direction, the disk platform may exert a force of up to 4.7 N. We communicate to the device firmware position commands (i.e., the optimal platform configuration) on an outer control loop running at 50 Hz. The device itself admits either position or force commands on the outer loop, as described in [15], but using the modified kinematics of the design in [5]. Then, an inner loop controls the position of each servomotor at a rate up to 1 kHz. The firmware transforms the desired platform configuration into desired joint angles, but note that our constrained optimization guarantees that these joint angles are always within the valid workspace of the device.

5.2 Contact Surface and Distance Function

The parameters of the mobile platform, $\mathbb{D} = (\mathbf{p}, \mathbf{n})$, can be expressed as a function of the end-effector coordinates $\mathbf{w}$ through the following kinematic relationships.

The three legs of the device are attached at fixed points on the mobile platform. These points have the following fixed positions in the local reference frame of the mobile platform:

$$\mathbf{B}_{1,0} = (b, \ 0, \ 0)^T, \tag{6}$$

$$\mathbf{B}_{2,0} = \left(-b\sin(\cos^{-1}(b_h/b)), \ -b_h, \ 0\right)^T,$$

$$\mathbf{B}_{3,0} = \left(-b\sin(\cos^{-1}(b_h/b)), \ b_h, \ 0\right)^T,$$

with platform dimensions $\{b = 20 \text{ mm}, b_h = 10.5 \text{ mm}\}$, as shown in Fig. 4-c.

The yaw angle ϕ of the platform can be obtained from roll and pitch angles as $\phi = \tan^{-1}\left(\frac{\sin\theta\sin\psi}{\cos\theta+\cos\psi}\right)$. Then, the rotation of the mobile platform w.r.t. a reference frame on the fixed platform is $\mathbf{R} = \mathbf{R}(z, \phi)\,\mathbf{R}(y, \theta)\,\mathbf{R}(x, \psi)$.

The center of the mobile platform is transformed to:

$$\mathbf{p} = \begin{pmatrix} b/2\,(\cos\phi\cos\theta - \sin\phi\sin\theta\sin\psi - \cos\phi\cos\psi) \\ -b\sin\phi\cos\theta \\ \Delta z \end{pmatrix}. \tag{7}$$

And the attachment points of the legs are transformed to:

$$\mathbf{B}_1 = \mathbf{p} + \mathbf{R}\,\mathbf{B}_{1,0}, \quad \mathbf{B}_2 = \mathbf{p} + \mathbf{R}\,\mathbf{B}_{2,0}, \quad \mathbf{B}_3 = \mathbf{p} + \mathbf{R}\,\mathbf{B}_{3,0}. \tag{8}$$

From these we obtain the transformed normal:

$$\mathbf{n} = \frac{(\mathbf{B}_2 - \mathbf{B}_1) \times (\mathbf{B}_3 - \mathbf{B}_1)}{\|(\mathbf{B}_2 - \mathbf{B}_1) \times (\mathbf{B}_3 - \mathbf{B}_1)\|}. \tag{9}$$

By differentiating these kinematic relationships, we also obtain the derivatives $\frac{\partial \mathbf{p}}{\partial \mathbf{w}}$ and $\frac{\partial \mathbf{n}}{\partial \mathbf{w}}$ needed in the computation of the gradient of the objective function in Eq. (4).

The evaluation of the objective function Eq. (3) requires the computation of distances from points on the surface of the finger model $\mathbb{F}$, $\{\mathbf{x}_i\}$, to the device platform. For points in contact with the virtual object, $i \in \mathbb{C}_\mathbb{O}$, we use an unsigned distance function to the mobile platform, because their cost function is symmetric. The distance computation distinguishes those points that are closer to the interior of the disk from those that are closer to the circumference of the disk. The same distinction is made for the computation of distance gradients $\frac{\partial \text{dist}(\mathbf{x}_i, \mathbb{D})}{\partial \mathbb{D}}$ in Eq. (4).

For points not in contact with the virtual object, $i \in \mathbb{N}_\mathbb{O}$, the cost function is not symmetric, hence they require the definition of a signed distance function. We follow a simple heuristic. We extend the device as a 90-degree truncated cone, as shown in Fig. 2, and we compute distances by distinguishing three cases: points that are closer to the interior of the disk, to the circumference of the disk, or to the surface of the cone. The cone approach worked well in practice, hence we did not investigate other options.

5.3 Inverse Kinematics and Workspace Constraints

With the proposed parallel mechanism, actuator joint angles $\mathbf{q}$ can be computed from the end-effector coordinates $\mathbf{w}$ using a closed-form solution of inverse kinematics.

The three legs of the device are attached at fixed points on the fixed platform. These points have the following positions in the reference frame of the fixed platform:

$$\mathbf{A}_1 = (a, \; 0, \; 0)^T, \tag{10}$$
$$\mathbf{A}_2 = \left(-a\sin(\cos^{-1}(a_h/a)), \; -a_h, \; 0\right)^T,$$
$$\mathbf{A}_3 = \left(-a\sin(\cos^{-1}(a_h/a)), \; a_h, \; 0\right)^T,$$

with platform dimensions $\{a = 15 \text{ mm}, a_h = 5 \text{ mm}\}$, as shown in Fig. 4-c.

For each joint $i \in \{1, 2, 3\}$, we compute the leg angle:

$$\beta_i = \pi - \cos^{-1}\left(\left(\frac{\mathbf{A}_i}{\|\mathbf{A}_i\|}\right)^T \frac{\mathbf{B}_i - \mathbf{A}_i}{\|\mathbf{B}_i - \mathbf{A}_i\|}\right); \tag{11}$$

the leg base angle:

$$\gamma_i = \cos^{-1}\left(\frac{L^2 - l^2 - \|\mathbf{B}_i - \mathbf{A}_i\|^2}{-2\,l\,\|\mathbf{B}_i - \mathbf{A}_i\|}\right); \tag{12}$$

and finally the joint angle:

$$q_i = \pi - \gamma_i - \beta_i, \tag{13}$$

with leg lengths $\{l = 10 \text{ mm}, L = 25 \text{ mm}\}$. The device would reach a singular configuration if the l and L legs in Fig. 4-c are aligned, but such situations are prevented through both hardware and software constraints.

On our device, workspace constraints are simple box constraints on the joint angles, i.e., $q_{\min} \leq q_i \leq q_{\max}$. The constraint gradients in Eq. (5) can be expressed by differentiating these box constraints, $\frac{\partial C_i}{\partial q_i} = \pm 1$, as well as the inverse kinematics formulation above to obtain $\frac{\partial q_i}{\partial \mathbf{w}}$.

6 EXPERIMENTS AND RESULTS

In this section, we provide implementation details about the full software and hardware platform in which we have tested our tactile rendering algorithm, and we discuss the results on different experiments. In particular, we discuss an error analysis of tactile rendering based on constrained optimization, compared to unconstrained optimization and a heuristic device-specific approach. Please also watch the accompanying video.

6.1 Implementation Platform and Performance

To simulate the deformation of the finger model $\mathbb{F}$, we use a tetrahedral mesh with 347 elements and 120 nodes, which is visible in Fig. 5-e and Fig. 5-f. Out of these nodes, we use 33 nodes located on the finger pad of the model to compute the contact surface deviation metric in Eq. (3). We chose the resolution of the finger model to achieve a good balance between accuracy and update rate. For LCSM tactile devices with few DoFs, the current model resolution is sufficient, but LCSM devices with more DoFs might benefit from models with higher resolution.

To track the user's finger, we use a LeapMotion device, which offers a sampling rate of 200 Hz. Its tracking resolution is highly dependent on external conditions.

However, in practice, the update rate is limited by our rendering algorithm, which runs at an average of 50 Hz. The dominant cost corresponds to the finger and contact simulation step (around 16 ms). The cost of device optimization grows from less than 1ms with unconstrained optimization to just under 4ms with constrained optimization. We have executed all our experiments on

a PC with an Intel Core-i7 2600 (3.4GHz) and 8GB of RAM. We have used Windows 10 in our examples, although our rendering algorithm and its implementation are multi-platform.

6.2 Exploration Examples

We have tested our tactile rendering algorithm on a variety of contact configurations. Fig. 5 shows three examples of users exploring virtual surfaces with various properties, while our tactile rendering algorithm commands the LCSM device used for testing.

Fig. 5-a and Fig. 5-b show a compressive motion of the finger pad against a flat surface. When the finger model $\mathbb{F}$ presses against the virtual surface, its contact area grows. As a result, our optimization computes a device platform configuration that increases the number of points in contact, and the platform moves towards the user's finger, generating an increasing normal force on the finger pad. The compressive deformation in this example is accurately rendered by the test device, as the relative motion between virtual finger and virtual surface matches exactly the translational DoF of the device.

Fig. 5-c and Fig. 5-d show an exploratory motion of the finger over an edge. The device used in our examples cannot render sharp features, but our optimization algorithm automatically finds a rounded edge as the most similar contact surface. Rendering of edge contact is a clear example of the influence of points not in contact in the objective function Eq. (3). In Fig. 5-d, the finger pad of the finger model is only partially in contact with the top flat surface. Using only points in contact for contact surface matching would bias the orientation of the device platform toward the orientation of the top flat surface. However, our rendering algorithm accounts for points in the finger pad not in contact, and finds a compromise device configuration by tilting the device platform and thus eliciting the perception of exploring a rounded edge.

Fig. 5-e and Fig. 5-f show an exploratory motion of the finger over the surface of a ball. In this case, the relative orientation and the contact location on the finger model vary during exploration. The optimization finds the device configuration that best approximates points in contact and points not in contact, subject to the DoFs and workspace limits of the device. A fully accurate planar approximation of the contact surface would require a LCSM device with 5 DoFs (i.e., full rigid motion except for the yaw angle), but the test device, not the tactile rendering algorithm, is limited to 3 DoFs.

6.3 Error Analysis and Comparisons

To validate the accuracy of our tactile rendering algorithm, we have designed a procedure to estimate the error between the contact force field computed in the simulated environment and the actual force field displayed by the device to the user. Note that our rendering algorithm does not use contact force information; therefore, our validation procedure avoids any bias in the comparison to other rendering methods. Due to the difficulty to measure a contact force field between the actual device and the user's finger, and thanks to the availability of an accurate finger simulation model [4], we perform a simulation-based estimation of the contact force field between the device and the user's finger. Moreover, simulation-based force estimation allows us to use controlled synthetic trajectories and to factor out other variables such as device bandwidth or device grounding, and we can focus on the validation of our tactile rendering approach alone.
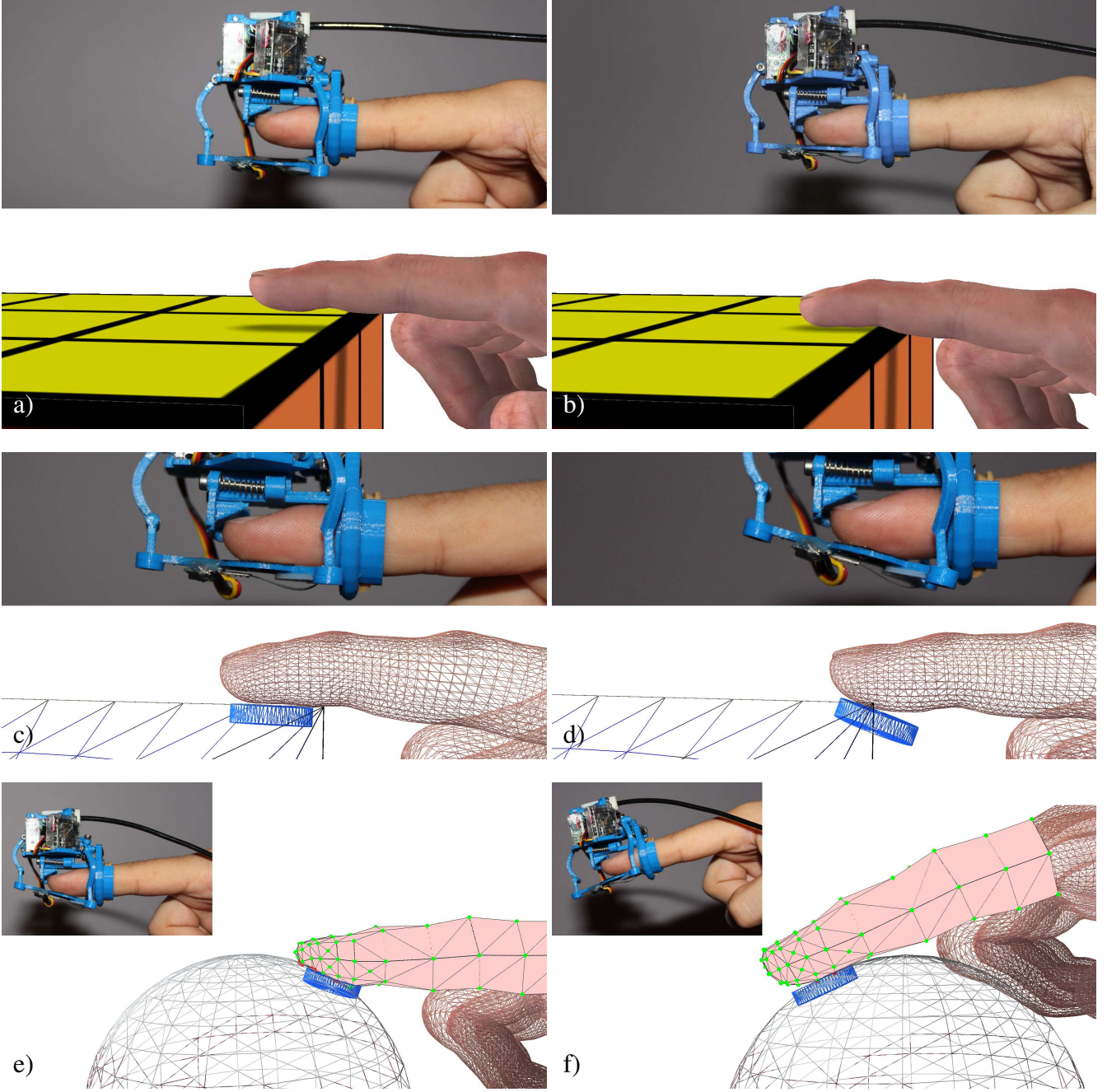
Fig. 5. Examples of tactile exploration on different surfaces. Thanks to our optimization-based tactile rendering algorithm, the device adapts its configuration to display a contact surface that maximizes the similarity with the contact surface in the virtual environment. From top to bottom, we show three different contact scenarios: (a,b) Pressing against a flat surface. The device moves normal to the finger pad to match the compression in the virtual environment. (c,d) Exploration of an edge. Even though the flat device cannot accurately render sharp features, our rendering algorithm estimates device orientations that display a best-fit rounded edge. (e,f) Exploration of a sphere. The device preserves the relative orientation between the finger pad and the surface being touched. In the sphere example, the images also show the low-resolution tetrahedral mesh used for the simulation of finger deformations.

Given a tactile rendering output, we execute a contact simulation between a virtual model of the device and the finger model, mimicking the interaction between the actual device platform and the user's finger. In this simulation, the finger model is fixed on the nail side, to reproduce the grounding of the fixed part of the device described in Section 5.1, and the device platform is positioned relative to its grounding, according to the configuration output by the tactile rendering algorithm. Then, we simulate the deformation of the finger model in contact with the device platform, using

the accurate nonlinear skin model. The resulting deformation and forces serve as an accurate estimate of the contact undergone by the user's real finger during tactile rendering. In the accompanying video and Fig. 1, we show an example of device contact simulation for the exploration of the ball. The left image shows the virtual contact between the finger model and the ball, the right image shows the real-world interaction between the device and the user's finger resulting from tactile rendering, and the inset shows the simulation of contact between the device model and the finger
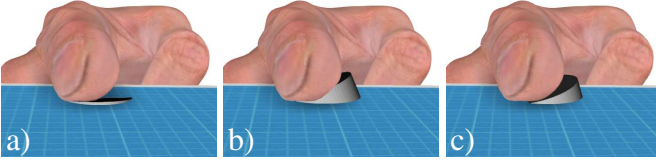
Fig. 6. These images highlight the rendering quality of our constrained optimization algorithm on a rolling motion of the finger. a) With unconstrained optimization, we obtain a device configuration that matches almost perfectly the underlying surface, but this configuration is not feasible due to device workspace constraints. b) We project the result of unconstrained optimization to the feasible workspace, but this produces a device configuration that penetrates deep into the finger model. This results in an excessive compression of the finger by the real-world device, hence in a high force error. c) With constrained optimization, we obtain a device configuration that satisfies the workspace constraints, yet it matches as close as possible the contact surface. Error grows quickly for the plane-fitting and unconstrained optimization methods when the device hits its maximum roll angle (35 degrees).
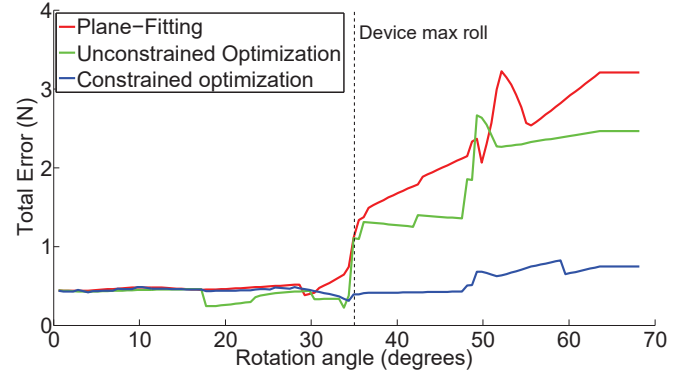

Fig. 7. Contact force field error for the finger rolling motion in Fig. 6. The error is compared for three different methods: a custom heuristic plane-fitting method (red), unconstrained optimization (green), and our constrained optimization method (blue).

model for error estimation.

For every tactile rendering step, and for the finger model interacting with the virtual environment, we evaluate the contact force $\overline{\mathbf{F}}_i$ on each of the finger surface nodes used for contact surface matching as described in Section 6.1. For the finger model interacting with the simulated device, we also measure the contact force $\mathbf{F}_i$ on each of the finger surface nodes. Then, we evaluate the contact force field error per rendering step as $\sum_i \|\mathbf{F}_i - \overline{\mathbf{F}}_i\|$.

We have evaluated the error of our rendering algorithm and we have compared it to other approaches on a finger rolling motion, shown in Fig. 6. We have designed a synthetic trajectory where the finger starts flat on a plane and then rolls slowly to one side. We compare the output of our tactile rendering using constrained optimization, unconstrained optimization as described in [3], and a device-dependent plane-fitting heuristic. A plane-fitting heuristic works reasonably well for contact with planar surfaces and for our particular device, but it does not generalize to arbitrary contact configurations or devices. Both unconstrained optimization and plane-fitting are followed by a constraint projection step to fit the actuator coordinates inside the workspace limits. Since the forward kinematics are not given in closed-form for our device, this projection is also formulated and solved as an optimization problem.

The snapshots in Fig. 6 depict the problems occurring with the unconstrained optimization, which are even more severe with simple plane-fitting. With unconstrained optimization, the device configuration matches almost perfectly the underlying plane (see Fig. 6-a), but this configuration is not feasible due to device workspace constraints. Once the device configuration is projected to the feasible workspace, the device penetrates deep into the finger model (see Fig. 6-b), which results in an excessive compression of the finger by the device, hence in a high rendering error. With our constrained optimization, instead, we obtain a device configuration that satisfies the workspace constraints, yet it matches as close as possible the contact plane (see Fig. 6-c).

Fig. 7 shows the contact force field error as a function of the roll angle, for all three methods. Once the finger reaches a roll angle of 35 degrees, the device hits its workspace limits, and the error grows quickly under unconstrained optimization or plane-fitting. With our tactile rendering approach based on constrained optimization, the contact force field is well approximated even when the device reaches its workspace limits. With all three methods, the force field exhibits an offset error of approximately 0.5 N,

which is due to the application of the input rolling trajectory.

Our error metric does not account for inaccuracies of the finger model, inaccuracies of the contact model, device bandwidth, or device mounting imperfections. Nevertheless, our error analysis provides conclusive evidence of the benefits of our rendering algorithm in contrast to simpler approaches. During actual tactile rendering of interaction with virtual environments, lack of collocation of the virtual and real fingers may constitute an additional source of perceptual error. In combination with visual rendering, and due to visual dominance over proprioception, the user expects to feel contact as visually perceived in the simulation; therefore, the perceived error due to lack of collocation is expected to be minimal. If visual feedback is not provided, lack of collocation resulting from wearability may have a larger influence and needs further analysis.

7 DISCUSSION AND FUTURE WORK

In this work, we have presented an optimization-based approach for tactile rendering. The core of our approach is to search for the device configuration that produces a contact surface that matches as close as possible the contact surface in the virtual environment. Our optimization-based tactile rendering is general, as it is valid for all types of local contact surface modulation devices, either based on open-chain mechanisms or parallel mechanisms, and it also handles device workspace constraints. Thanks to this generality, this optimization-based approach establishes a formal framework for cutaneous rendering.

The demonstrations show only finger tracking instead of full-hand tracking, and virtual environments that are static and computationally simple. Using a novel fast solver for nonlinear constrained dynamics, we have demonstrated the tactile rendering algorithm in the context of multi-finger grasping interactions [41]. Although not tested in our examples either, it would be possible to apply the algorithm to other LCSM devices, including other parallel-kinematics devices and open-chain devices; extend the implementation beyond the finger pad; and adapt the geometric and mechanical parameter values of the finger model for each user [43]. The influence of each parameter on the final accuracy of tactile rendering requires further analysis though.

The performance of the optimization is roughly linear in the number of vertices, although this could be accelerated by reducing computations for far vertices. But the main performance

bottleneck is the number of DoFs of the device. Currently, with just three DoFs, this is not a problem, but more complex devices might need faster optimizations. With more complex devices, constrained optimization might suffer from local minima problems too.

As a final remark, the central idea of our approach, i.e., posing cutaneous rendering as a contact surface matching problem, admits extensions too. Ideally, one would want to match contact forces, or even internal stress in the finger, not just the geometry of contact surfaces, but the computation of contact forces and deformations in the context of an optimization framework would be far more complex. Indeed, the contact surface matching approach is valid only for virtual objects that are rigid or stiffer than the finger pad. With a soft object the contact area would grow fast even for very low forces, and an LCSM device with a rigid mobile platform would fail to render such effects correctly.

ACKNOWLEDGMENTS

REFERENCES

[1] K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles, "Haptic rendering: Programming touch interaction with virtual objects," in *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, 1995, pp. 123–130.

[2] M. Otaduy, C. Garre, and M. Lin, "Representations and algorithms for force-feedback display," *Proceedings of the IEEE*, vol. 101, no. 9, pp. 2068–2080, Sept 2013.

[3] A. G. Perez, D. Lobo, F. Chinello, G. Cirio, M. Malvezzi, J. San Martin, D. Prattichizzo, and M. A. Otaduy, "Soft finger tactile rendering for wearable haptics," in *World Haptics Conference (WHC), 2015 IEEE*, 2015, pp. 327–332.

[4] A. G. Perez, G. Cirio, F. Hernandez, C. Garre, and M. A. Otaduy, "Strain limiting for soft finger contact simulation," in *World Haptics Conference (WHC), 2013*, 2013, pp. 79–84.

[5] F. Chinello, M. Malvezzi, C. Pacchierotti, and D. Prattichizzo, "Design and development of a 3rrs wearable fingertip cutaneous device," in *Advanced Intelligent Mechatronics (AIM), 2015 IEEE International Conference on*, 2015, pp. 293–298.

[6] S. Brewster, F. Chohan, and L. Brown, "Tactile feedback for mobile interactions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2007, pp. 159–162.

[7] R. Traylor and H. Tan, "Development of a wearable haptic display for situation awareness in altered-gravity environment: some initial findings," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*, 2002, pp. 159–164.

[8] J. Lieberman and C. Breazeal, "Tikl: Development of a wearable vibrotactile feedback suit for improved human motor learning," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 919–926, 2007.

[9] H. Kim, C. Seo, J. Lee, J. Ryu, S. Yu, and S. Lee, "Vibrotactile display for driving safety information," in *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, 2006, pp. 573–577.

[10] S. Scheggi, F. Chinello, and D. Prattichizzo, "Vibrotactile haptic feedback for human-robot interaction in leader-follower tasks," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, 2012, pp. 51:1–51:4.

[11] G.-H. Yang, K.-U. Kyung, M. Srinivasan, and D.-S. Kwon, "Development of quantitative tactile display device to provide both pin- array-type tactile feedback and thermal feedback," in *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint*, 2007, pp. 578–579.

[12] T.-H. Yang, S.-Y. Kim, C. H. Kim, D.-S. Kwon, and W. Book, "Development of a miniature pin-array tactile module using elastic and electromagnetic force for mobile devices," in *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint*, 2009, pp. 13–17.

[13] I. Sarakoglou, N. Garcia-Hernandez, N. Tsagarakis, and D. Caldwell, "A high performance tactile feedback display and its integration in teleoperation," *Haptics, IEEE Transactions on*, vol. 5, no. 3, pp. 252–263, 2012.

[14] A. Frisoli, M. Solazzi, F. Salsedo, and M. Bergamasco, "A fingertip haptic display for improving curvature discrimination," *Presence*, vol. 17, no. 6, pp. 550–561, Dec 2008.

[15] D. Prattichizzo, F. Chinello, C. Pacchierotti, and M. Malvezzi, "Towards wearability in fingertip haptics: a 3-dof wearable device for cutaneous force feedback," *IEEE Transactions on Haptics*, vol. 6, no. 4, pp. 506–516, 2013.

[16] A. Serio, M. Bianchi, and A. Bicchi, "A device for mimicking the contact force/contact area relationship of different materials with applications to softness rendering," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013, pp. 4484–4490.

[17] H. Dostmohamed and V. Hayward, "Trajectory of contact region on the fingerpad gives the illusion of haptic shape," *Exp Brain Res*, vol. 164, no. 3, pp. 387–394, Jul 2005.

[18] A. Frisoli, M. Solazzi, M. Reiner, and M. Bergamasco, "The contribution of cutaneous and kinesthetic sensory modalities in haptic perception of orientation," *Brain Res. Bull.*, vol. 85, no. 5, pp. 260–266, Jun 2011.

[19] W. R. Provancher, M. R. Cutkosky, K. J. Kuchenbecker, and G. Niemeyer, "Contact location display for haptic perception of curvature and object motion," *International Journal of Robotics Research*, vol. 24, no. 9, p. 691–702, 2005.

[20] Z. Quek, S. Schorr, I. Nisky, W. Provancher, and A. Okamura, "Sensory substitution using 3-degree-of-freedom tangential and normal skin deformation feedback," in *Haptics Symposium (HAPTICS), 2014 IEEE*, Feb 2014, pp. 27–33.

[21] J. Park, A. Doxon, W. Provancher, D. Johnson, and H. Tan, "Haptic edge sharpness perception with a contact location display," *Haptics, IEEE Transactions on*, vol. 5, no. 4, pp. 323–331, 2012.

[22] M. Salada, J. Colgate, M. Lee, and P. Vishton, "Validating a novel approach to rendering fingertip contact sensations," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*, 2002, pp. 217–224.

[23] J. Pasquero and V. Hayward, "Stress: A practical tactile display system with one millimeter spatial resolution and 700 hz refresh rate," in *in Proc. Eurohaptics 2003*, 2003, pp. 94–110.

[24] K. Minamizawa, H. Kajimoto, N. Kawakami, and S. Tachi, "A wearable haptic display to present the gravity sensation - preliminary observations and device design," in *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint*, March 2007, pp. 133–138.

[25] B. Gleeson, S. Horschel, and W. Provancher, "Design of a fingertip-mounted tactile display with tangential skin displacement feedback," *Haptics, IEEE Transactions on*, vol. 3, no. 4, pp. 297–301, 2010.

[26] M. Solazzi, W. Provancher, A. Frisoli, and M. Bergamasco, "Design of a sma actuated 2-dof tactile device for displaying tangential skin displacement," in *World Haptics Conference (WHC), 2011 IEEE*, 2011, pp. 31–36.

[27] D. Leonardis, M. Solazzi, I. Bortone, and A. Frisoli, "A wearable fingertip haptic device with 3 dof asymmetric 3-rsr kinematics," in *World Haptics Conference (WHC), 2015 IEEE*, 2015, pp. 388–393.

[28] M. Bianchi, G. Valenza, A. Serio, A. Lanata, A. Greco, M. Nardelli, E. Scilingo, and A. Bicchi, "Design and preliminary affective characterization of a novel fabric-based tactile display," in *Haptics Symposium (HAPTICS), 2014 IEEE*, 2014, pp. 591–596.

[29] R. Sodhi, I. Poupyrev, M. Glisson, and A. Israr, "Aireal: Interactive tactile experiences in free air," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 134:1–134:10, 2013.

[30] B. Long, S. A. Seah, T. Carter, and S. Subramanian, "Rendering volumetric haptic shapes in mid-air using ultrasound," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 181:1–181:10, 2014.

[31] C. Zilles and J. Salisbury, "A constraint-based god-object method for haptic display," in *Intelligent Robots and Systems 95. 'Human Robot*

Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on, vol. 3, 1995, pp. 146–151 vol.3.

[32] 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 Transactions on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 458–469, 2007.

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

[34] D. Wang, X. Zhang, Y. Zhang, and J. Xiao, "Configuration-based optimization for six degree-of-freedom haptic rendering for fine manipulation," *Haptics, IEEE Transactions on*, vol. 6, no. 2, pp. 167–180, 2013.

[35] S. Brewster and L. M. Brown, "Tactons: Structured tactile messages for non-visual information display," in *Proceedings of the Fifth Conference on Australasian User Interface - Volume 28*, 2004, pp. 15–23.

[36] S.-C. Kim, A. Israr, and I. Poupyrev, "Tactile rendering of 3d features on touch surfaces," in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, 2013, pp. 531–538.

[37] W. S. Harwin and N. Melder, "Improved haptic rendering for multi-finger manipulation using friction cone based god-objects," in *Eurohaptics Conference*, 2002.

[38] H. Kawasaki, Y. Ohtuka, S. Koide, and T. Mouri, "Perception and haptic rendering of friction moments," *IEEE Transactions on Haptics*, vol. 4, no. 1, pp. 28–38, 2011.

[39] F. Barbagli, A. Frisoli, K. Salisbury, and M. Bergamasco, "Simulating human fingers: a soft finger proxy model and algorithm," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings. 12th International Symposium on*, 2004, pp. 9–17.

[40] D. Prattichizzo, C. Pacchierotti, and G. Rosati, "Cutaneous force feedback as a sensory subtraction technique in haptics," *EEE Trans. Haptics*, vol. 5, no. 4, pp. 289–300, 2012.

[41] A. G. Perez, G. Cirio, D. Lobo, F. Chinello, D. Prattichizzo, and M. A. Otaduy, "Efficient nonlinear skin simulation for multi-finger tactile rendering," in *2016 IEEE Haptics Symposium (HAPTICS)*, 2016, pp. 155–160.

[42] M. Müller and M. Gross, "Interactive virtual materials," *Proc. of Graphics Interface*, 2004.

[43] E. Miguel, M. D'Angelo, F. Cannella, M. Bianchi, M. Memeo, A. Bicchi, D. Caldwell, and M. Otaduy, "Characterization of nonlinear finger pad mechanics for tactile rendering," in *World Haptics Conference (WHC), 2015 IEEE*, 2015.

[44] C. Garre, F. Hernandez, A. Gracia, and M. A. Otaduy, "Interactive simulation of a deformable hand for haptic rendering," in *Proc. of World Haptics Conference*, 2011.

[45] S. G. Johnson, "The NLopt nonlinear-optimization package," http://ab-initio.mit.edu/nlopt.

[46] L.-W. Tsai, *Robot analysis: the mechanics of serial and parallel manipulators.* John Wiley & Sons, 1999.

Daniel Lobo is currently working toward a PhD at Universidad Rey Juan Carlos, Madrid, Spain, working with Miguel A. Otaduy. He obtained the BS degree in computer science and the MS degree in computer graphics, virtual reality and videogames from URJC Madrid in 2013 and 2014, respectively. His main research interests include virtual reality, mixed reality and haptic rendering.

Francesco Chinello received his MS Degree and the Ph.D. Degree at the Dept. of Information Engineering and Mathematics of the University of Siena. He is currently postdoctoral researcher at the the Dept. of Advanced Robotics of the Italian Institute of Technology, in Genova. His research interests include developing and testing haptic and robotic systems, focusing on cutaneous force feedback for virtual interaction and teleoperation.

Gabriel Cirio Gabriel Cirio is a currently a Post-doctoral fellow at Universidad Rey Juan Carlos (URJC) Madrid, working with Miguel A. Otaduy. He obtained a M.S. from INSA Lyon in 2007 and from the University of Lyon in 2008, and did a PhD in Computer Science at Inria Rennes from 2009 to 2011. His main research interests span the broad field of multimodal rendering and interaction, including physics-based computer animation, sound simulation, haptics rendering and virtual reality. He has served in the program committee of IEEE Virtual Reality 2015 and ACM Virtual Reality Software and Technology 2015.

Monica Malvezzi (M12) is an Assistant Professor of Mechanics and Mechanism Theory at the Dept. of Information Engineering and Mathematics of the University of Siena. She received the Laurea degree in Mechanical Engineering from the University of Florence in 1999 and the Ph.D. degree in Ap- plied Mechanics from the University of Bologna in 2003. From 2015 she has been also visiting scientist at the Department of Advanced Robotics, Istituto Italiano di Tecnologia, in Genova, Italy. Her main research interests are in control of mechanical systems, robotics, vehicle localization, multibody dynamics, haptics, grasping and dexterous manipulation.

Alvaro G. Perez received the engineering degree in computer science from the Polytechnic University of Madrid in 2007, the MS degree in computer graphics, videogames and virtual reality from the Universidad Rey Juan Carlos in 2010, and the PhD degree in computer science from the same university in 2015. He is currently CTO in the Spanish start-up Eurob Creative. Previously, he worked in Deimos Space and the European Space Agency. His research interests include physically based simulation, haptic rendering, virtual reality and 3D modeling.

José San Martín obtained a Mechanical Engineer Degree at UPCO-ICAI (Madrid, Spain) in 1997. He worked at ALSTOM and other firms until 2003 when he joined the Universidad Rey Juan Carlos-URJC. He obtained a PhD from URJC in Madrid in 2007 and is Associate Professor since 2007.He collaborated with Mechatronics Lab at Kyoto University in 2007-2008. His main research interests are haptics design and optimization, mechatronics and virtual reality based trainers.

Domenico Prattichizzo received the Ph.D. degree in Robotics and Automation from the University of Pisa in 1995. Since 2002 he is a Professor of Robotics at the University of Siena and since 2009 he is a Scientific Consultant at Istituto Italiano di Tecnoloogia. In 1994, he was a Visiting Scientist at the MIT AI Lab. Since 2014, he is Associate Editor of Frontiers of Biomedical Robotics. From 2007 to 2013 he has been Associate Editor in Chief of the IEEE Transactions on Haptics. From 2003 to 2007, he has been Associate Editor of the IEEE Transactions on Robotics and IEEE Transactions on Control Systems Technologies. He has been Chair of the Italian Chapter of the IEEE RAS (2006-2010), awarded with the IEEE 2009 Chapter of the Year Award. Research interests are in haptics, grasping, visual servoing, mobile robotics and geometric control. He is currently the Coordinator of the IP collaborative project WEARable HAPtics for Humans and Robots (WEARHAP).

Miguel A. Otaduy received the BS degree in electrical engineering from Mondragón University in 2000, and the MS and PhD degrees in computer science from the University of North Carolina at Chapel Hill, in 2003 and 2004, respectively. He is an associate professor in the Department of Computer Science, Universidad Rey Juan Carlos (URJC Madrid), where he leads the Multimodal Simulation Lab. From 2005 to 2008, he was a research associate at ETH Zurich. His research interests extend across physics-based simulation, covering algorithmic design or applied problems for virtual touch, animation, fashion, computational medicine, or fabrication. He has served on the editorial board for several journals and conferences, most notably IEEE Transactions on Haptics, IEEE Transactions on Visualization and Computer Graphics, 2013 - 2019 IEEE World Haptics Conference, 2013 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, and 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.