Soft Finger Tactile Rendering for Wearable Haptics

Alvaro G. Perez[1], Daniel Lobo[1], Francesco Chinello[2,3], Gabriel Cirio[1], Monica Malvezzi[2], José San Martín[1], Domenico Prattichizzo[2,3] and Miguel A. Otaduy[1]

***Abstract*— This paper introduces a tactile rendering algorithm for wearable cutaneous devices that stimulate the skin through local contact surface modulation. The first step in the algorithm simulates contact between a skin model and virtual objects, and computes the contact surface to be rendered. The accuracy of this surface is maximized by simulating soft skin with its characteristic nonlinear behavior. The second step takes the desired contact surface as input, and computes the device configuration by solving an optimization problem, i.e., minimizing the deviation between the contact surface in the virtual environment and the contact surface rendered by the device. The method is implemented on a thimble-like wearable device.**

I. 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 [27]. 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) [17].

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 interaction with virtual environments is mostly limited to tool-based interaction, but the progress on cutaneous 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 paper, we propose a tactile rendering algorithm for wearable cutaneous devices that stimulate the skin through local contact surface modulation (LCSM). Our algorithm is 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.

[1] URJC Madrid, c/ Tulipán s/n, 28933 Móstoles (Madrid), Spain.

[2] Department of Information Engineering and Mathematics, University of Siena, Via Roma 56, 53100 Siena, Italy.

[3] Department of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genova, Italy.

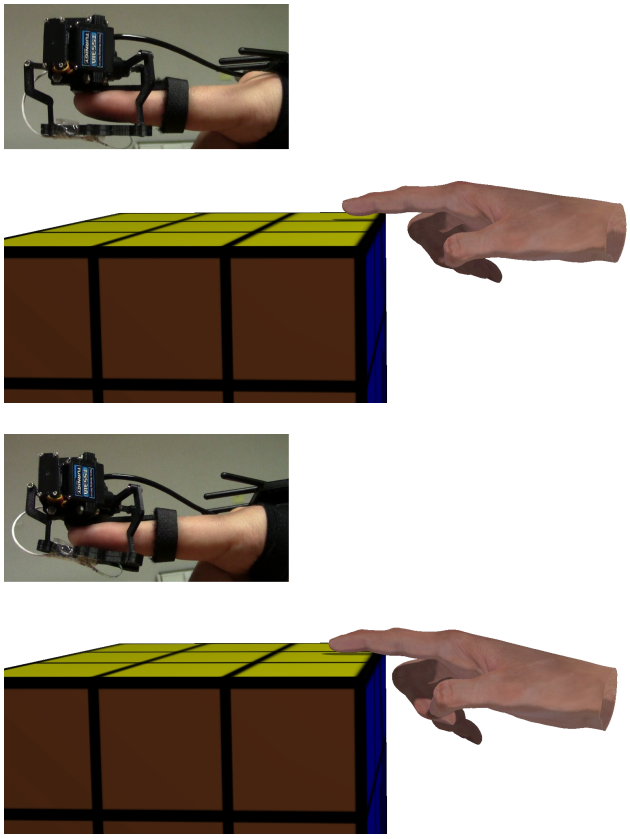

Fig. 1. Compression of a finger pad against a flat surface. With our tactile rendering algorithm, the device reproduces the contact and deformation computed in the virtual scenario.

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 model the skin using a nonlinear model [20]. As a second step, we optimize the device configuration to minimize contact surface deviation.

In our current developments, presented here, we apply our algorithm for tactile rendering of finger pad contact using a wearable device, similar to the one described in [22]. We have tested our rendering algorithm on a variety of contact configurations. We show that, with our algorithm, the device successfully replicates compressive motion (Fig. 1) and the exploration of smooth surfaces with varying orientation (Fig. 7). But it also approximates well interactions with sharp features (Fig. 2).

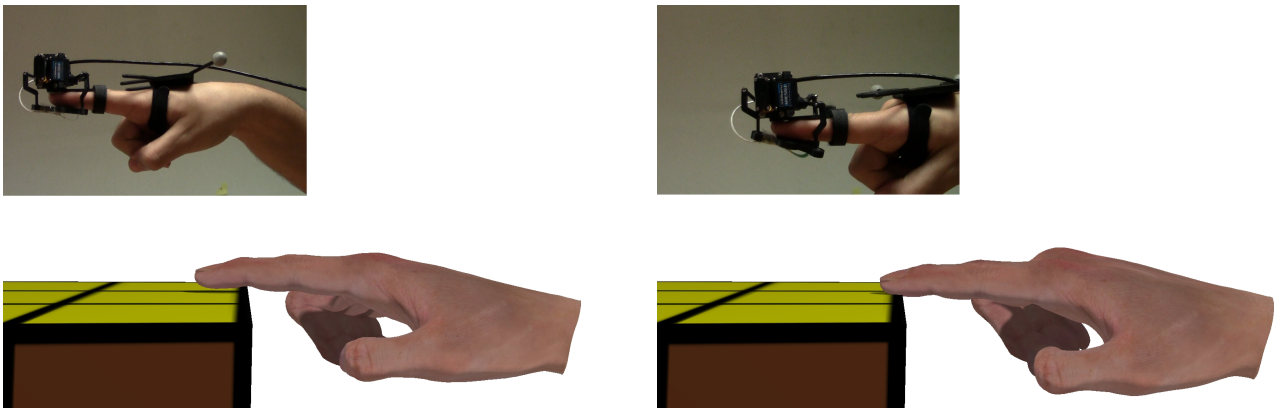

Fig. 2. An exploratory motion of the finger pad over an edge. Notice how the platform reorients to exert a force only on the portion of the finger pad that is under contact.

II. Related Work

As of today, there is no standardized skin stimulation method for cutaneous haptic rendering. Vibratory feedback has been successfully used for conveying information through the tactile sensory channel. The most common example nowadays is the use of vibrotactile displays [3], but vibratory feedback has also been integrated in wearable devices, e.g., on the user's back [33], using an arm suit [12], on the foot [10], or as a bracelet [29].

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 [36], [37], [28], a mobile platform located under the finger pad [6], [22], or using a flexible membrane to control the ratio between contact force and contact area [30]. Dostmohamed and Hayward [4] studied the perception of shape by controlling the trajectory of the contact region, while Frisoli et al. [5] 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. [24] 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 [25], and they also designed a rendering algorithm to faithfully account for edge sharpness in the optimization of contact location [18].

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 [26]. Other example implementations include the application of distributed and modulated local stretch at high frequencies to simulate texture exploration [19], applying stretch with a strap on the finger pad [14], 2D tangential displacement of the finger pad [8], [32], or fabric-based bracelets [1].

Finally, a recent alternative is the use of ultrasound for mid-air cutaneous stimulation [31], [13].

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 [38], [16], [21], [17], [35].

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 [2] and for friction modulation using electrovibration [11]. 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. [22] 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.

Cutaneous rendering enjoys an important advantage over kinesthetic rendering. Without kinesthetic feedback, the haptic loop is intrinsically passive [23]. As a result, stability of cutaneous rendering does not impose impedance or update rate restrictions.

III. Tactile Rendering Algorithm

Our tactile rendering algorithm consists of two major components: a simulation of a skin model in contact with the virtual environment, and an optimization that computes the device command based on contact surface matching. We first summarize the simulation of the skin model, and then we describe in detail contact surface matching.

A. Simulation of Skin Contact

For the purpose of rendering cutaneous feedback to the user, the first step in our algorithm aims at computing accurate contact information between a model of the user's skin and the virtual environment. Without loss of generality, let us assume that contact takes place between a finger $\mathbb{F}$ and a virtual object $\mathbb{O}$. At every simulation step, we wish to identify the contact surface $\mathbb{S}_{\mathbb{O}}$ between $\mathbb{F}$ and $\mathbb{O}$.

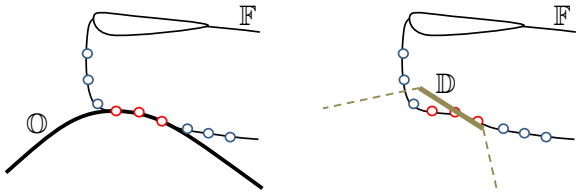

Fig. 3. Schematic depiction of Contact Surface Matching. Left: Contact between a finger model $\mathbb{F}$ and a virtual object $\mathbb{O}$ produces a set of points in contact $\mathbb{C}$, shown in red, and a set of points not in contact $\mathbb{N}$, in blue. Right: Contact Surface Matching aims to 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).

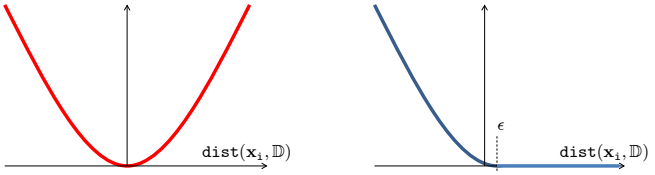

Fig. 4. The cost functions are different for points in contact or not in contact. For points in contact (left), we penalize equally the distance to the device. For points not in contact (right), we only penalize those that penetrate the device.

Using the cutaneous device, we will try to reproduce the contact surface $\mathbb{S}_\mathbb{O}$ on the real finger of the user. As the device moves against the user's finger, the surface of the skin will change. Therefore, to compute a correct surface matching, the simulation of contact between $\mathbb{F}$ and $\mathbb{O}$ 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 [20], which is capable of reproducing the extreme nonlinearities in human skin. At low forces, we compute deformations using a regular linear corotational finite element model (FEM) [15]. With a low Young modulus the finger pad 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 saturates at high forces. The original work in [20] demonstrates that this nonlinear model replicates the force-area response of the finger pad.

To couple the skin simulation to the user's motion, we follow the same overall architecture as in [7]. 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. Due to the lack of kinesthetic rendering, it is not possible to constrain the user's motion upon virtual contact, but the excess of user motion is absorbed by the viscoelastic coupling between tracked configuration and rigid body simulation, and does not affect the quality of contact and skin simulation.

B. Contact Surface Matching

Let us consider a generic LCSM device $\mathbb{D}$, whose surface is described by a set of degrees of freedom (DoFs) $\mathbf{q}$. Contact between the device and the finger $\mathbb{F}$ produces a contact surface $\mathbb{S}_\mathbb{D}$. Our goal is to determine the device DoFs such that the contact surfaces $\mathbb{S}_\mathbb{D}$ and $\mathbb{S}_\mathbb{O}$ match best.

One key element in our rendering algorithm is a descriptor of contact surface deviation. Conceptually, given two surfaces, we want the sets of points in contact in both surfaces to be the same, and we also want the sets of points not in contact to be the same. In our application, points in contact between the finger $\mathbb{F}$ and the virtual object $\mathbb{O}$ have zero distance, and we wish the same points to have zero distance between the finger and the device $\mathbb{D}$. But for points not in contact between the finger and the virtual object, we simply want them to have positive distance between the finger 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, as shown in Fig. 3. Our surface deviation 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 relevant 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 $\mathbb{F}$, we split them into a set $\mathbb{C}$ of points in contact with the virtual object $\mathbb{O}$, and a set $\mathbb{N}$ of points not in contact. This information is provided by the skin contact simulation described in the previous section. For points in contact, $i \in \mathbb{C}$, 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. 4-left. For points not in contact, $i \in \mathbb{N}$, 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. 4-right. In practice, we want the distance of points not in contact to be larger than a small tolerance ε. Altogether, we define the contact surface deviation metric as:

$$F = \sum_{i \in \mathbb{C}} \text{dist}(\mathbf{x}_i, \mathbb{D})^2 + \sum_{i \in \mathbb{N}} \min(0, \text{dist}(\mathbf{x}_i, \mathbb{D}) - \varepsilon)^2. \quad (1)$$

Now we are ready to formulate tactile rendering with an LCSM device as the minimization of the contact surface deviation metric between $\mathbb{S}_\mathbb{O}$ and $\mathbb{S}_\mathbb{D}$. The variables of this minimization are the DoFs of the device, $\mathbf{q}$. Then, the optimization problem is formally posed as:

$$\mathbf{q} = \arg\min F(\mathbf{q}). \quad (2)$$

The estimation of the contact surface $\mathbb{S}_\mathbb{D}$ between device and finger should use an accurate model of the finger skin,

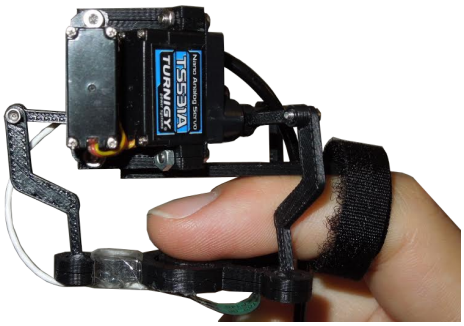

Fig. 5. We use a wearable cutaneous device with a fixed part mounted on the nail and a mobile part in contact with the finger pad. The parallel structure allows controlling the distance and the orientation of the mobile part.

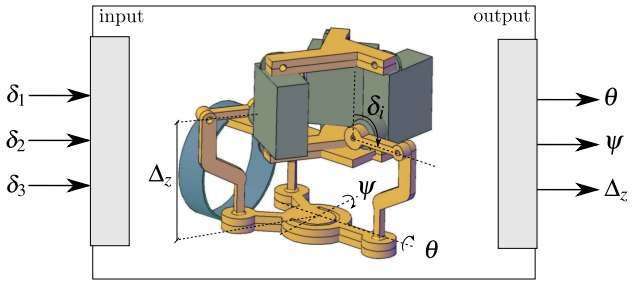

Fig. 6. Joint angles $(\delta_1, \delta_2, \delta_3)$ control the DoFs of a disk platform (two rotational: pitch θ, and roll ψ; and one translational: Δz), and determine the contact surface exposed to the finger pad.

and deform accordingly 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. Based on this observation, on every simulation step we take the deformed finger model $\mathbb{F}$, and use this deformed finger to compute distances to the device model.

IV. Implementation on a Wearable Device

We have implemented our general tactile rendering algorithm on the robotic wearable thimble shown in Fig. 5. In this section, we first provide a description of the main characteristics of the device. Then, we describe the specific implementation of the contact surface matching algorithm for this wearable thimble device.

A. The Device

The device is composed of a rigid and a mobile part. The rigid part is grounded on the middle phalanx of the index finger, on the nail side, and holds three servomotors, with joint angles $(\delta_1, \delta_2, \delta_3)$. The rigid and mobile parts are connected using three limbs with an RRS (Revolute-Revolute-Spherical) structure, which leads to a parallel mechanism with two angular and one translational DoFs [34]. The two angular DoFs are pitch, θ, and roll, ψ, angles, and the translational DoF is a displacement Δz, shown in Fig. 6. 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. Following the notation in Section III-B, its DoFs are denoted as $\mathbf{q} = (\theta, \psi, \Delta z)$. With the proposed parallel mechanism, joint angles can be computed from the DoFs using a closed-form solution of inverse kinematics.

The structure of our device is similar to the one presented by Prattichizzo et al. [22], but we substitute cable links between the rigid and mobile parts with the RRS limbs. The 3RRS parallel structure overcomes the underactuation problems of the cable-driven device, and the use of rigid links enables a controllable contact vs. no-contact condition of the platform. In addition, our device is actuated using three servomotors with greater 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.

The external control loop of the device is executed at a rate of 30 Hz, while an internal loop performs the position control of the mobile platform.

B. Algorithm Implementation

To simulate the deformation of the finger, we use a tetrahedral mesh with 347 elements. And we sample the surface triangle mesh with 120 vertices for the purpose of representing contact surfaces. In every simulation step, we split this set of vertices into those in contact, $\mathbb{C}$, and those not in contact, $\mathbb{N}$. Given this data, we are ready to compute the configuration of the device $\mathbf{q} = (\theta, \psi, \Delta z)$ using the contact surface matching algorithm described in Section III-B.

The evaluation of the objective function Eq. (1) requires the computation of distances from the vertices $\{\mathbf{x}_i\}$ to the device platform. Since the device is not formed by a closed-surface, signed distances are not well defined. This is not an issue for points in contact, i.e., $i \in \mathbb{C}$, because their cost function is symmetric and works fine with unsigned distances. But for points not in contact, i.e., $i \in \mathbb{N}$, 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. 3, and for points not in contact we compute signed distances to the surface of this cone.

To optimize the objective function and compute the device DoFs as in Eq. (2), we have used a Levenberg-Marquardt nonlinear least-squares optimization, specifically the min-pack implementation in Eigen [9]. We approximate the gradient of the objective function through finite differencing in Eigen's built-in module.

Once the device DoFs are optimized, we compute the joint angles $(\delta_1, \delta_2, \delta_3)$ using the inverse kinematics model of the device, apply workspace limits to these angles, and send the rendering command to the actual device.

V. Experiments and Results

We have executed all the experiments on a PC with an Intel Core-i7 2600 (3.4GHz) and 8GB of RAM. To track

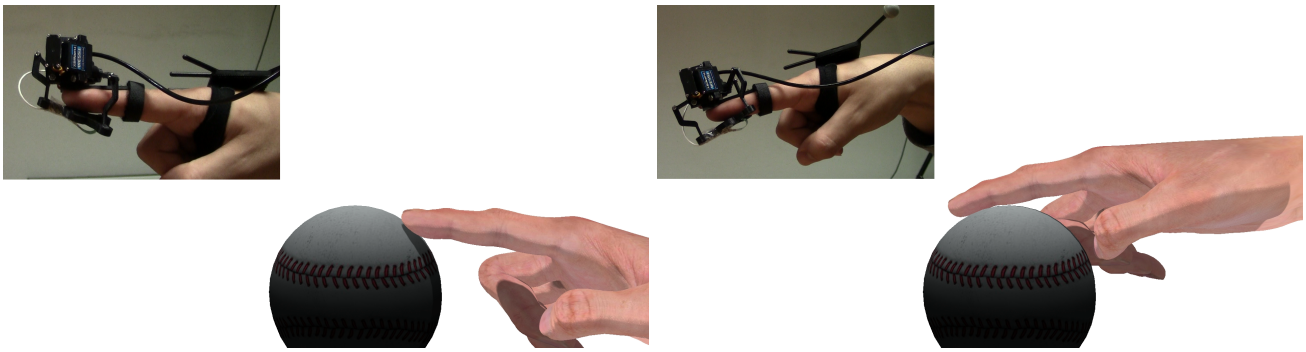

Fig. 7. Exploration of a sphere with the finger pad. Notice how the device preserves the relative orientation between the finger pad and the surface being touched.

the user's finger, we have used an OptiTrack system with 5 infrared cameras.

Our simulations ran at an average of 60 Hz, with the simulation step taking around 16ms and the optimization step taking less than 1ms.

We have tested a variety of contact interactions against different surfaces, as shown in the video accompanying this paper.

Fig. 1 shows a compressive motion of the finger pad against a flat surface. We can clearly observe how the platform is barely under contact with the finger pad right when pressing on the surface. During the compressive motion, the platform moves towards the finger, generating an increasing normal force on the finger pad.

Fig. 2 shows an exploratory motion of the finger over an edge. When only part of the finger pad is in contact with the surface, the optimization yields a platform configuration that exerts a force only on the contacting part. This behavior is coherent with the real-world scenario, and shows that our algorithm takes into account non-contacting nodes when computing the optimization. The device used in the experiments cannot exactly replicate contact against an edge, but our algorithm successfully approximates this interaction by smoothing the edge.

Fig. 7 shows an exploratory motion of the finger over the surface of a sphere. We can observe how the orientation of the platform successfully approximates the relative orientation between the finger pad and the surface under contact.

VI. Discussion and Future Work

In this work, we have explored contact surface matching as an approach for tactile rendering with local contact surface modulation devices. Our results demonstrate that the approach succeeds to display contact surfaces that approximate arbitrary interactions with virtual objects; not only smooth contact, but also edge contact. Our algorithm is general and could be applied to other LCSM devices, such as those based on pin arrays. We also believe it contributes to the establishment of a formal framework for cutaneous rendering.

The proposed algorithm suffers several limitations, which could be addressed as part of the many open avenues for future work. The virtual environments that we have tested are computationally simple, the demonstrators show only finger tracking instead of full-hand tracking, and the implementation is currently limited to the finger pad and one particular LCSM device. 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.

As a final remark, the central idea of our paper, 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

The authors wish to thank the anonymous reviewers for their helpful comments. This project was supported in part by grants from the EU FP7 (project no. 601165 WEARHAP), the European Research Council (ERC Starting Grant no. 280135 Animetrics), and the Spanish Ministry of Economy (TIN2012-35840). The work of Gabriel Cirio was funded in part by the Spanish Ministry of Science and Education through a Juan de la Cierva Fellowship.

References

[1] 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*, pages 591–596, 2014.

[2] 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*, pages 15–23, 2004.

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

[4] H. Dostmohamed and V. Hayward. Trajectory of contact region on the fingerpad gives the illusion of haptic shape. *Exp Brain Res*, 164(3):387–394, Jul 2005.

[5] 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.*, 85(5):260–266, Jun 2011.

[6] A. Frisoli, M. Solazzi, F. Salsedo, and M. Bergamasco. A fingertip haptic display for improving curvature discrimination. *Presence*, 17(6):550–561, Dec 2008.

[7] 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.

[8] B. Gleeson, S. Horschel, and W. Provancher. Design of a fingertip-mounted tactile display with tangential skin displacement feedback. *Haptics, IEEE Transactions on*, 3(4):297–301, 2010.

[9] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[10] 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*, pages 573–577, 2006.

[11] 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*, pages 531–538, 2013.

[12] J. Lieberman and C. Breazeal. Tikl: Development of a wearable vibrotactile feedback suit for improved human motor learning. *Robotics, IEEE Transactions on*, 23(5):919–926, 2007.

[13] B. Long, S. A. Seah, T. Carter, and S. Subramanian. Rendering volumetric haptic shapes in mid-air using ultrasound. *ACM Trans. Graph.*, 33(6):181:1–181:10, 2014.

[14] 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*, pages 133–138, March 2007.

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

[16] 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*, 13(3):458–469, 2007.

[17] M. Otaduy, C. Garre, and M. Lin. Representations and algorithms for force-feedback display. *Proceedings of the IEEE*, 101(9):2068–2080, Sept 2013.

[18] J. Park, A. Doxon, W. Provancher, D. Johnson, and H. Tan. Haptic edge sharpness perception with a contact location display. *Haptics, IEEE Transactions on*, 5(4):323–331, 2012.

[19] 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*, pages 94–110, 2003.

[20] 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*, pages 79–84, 2013.

[21] I. Peterlik, M. Nouicer, C. Duriez, S. Cotin, and A. Kheddar. Constraint-based haptic rendering of multirate compliant mechanisms. *IEEE Transactions on Haptics*, 4(3):175 –187, 2011.

[22] 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*, 6(4):506–516, 2013.

[23] D. Prattichizzo, C. Pacchierotti, and G. Rosati. Cutaneous force feedback as a sensory subtraction technique in haptics. *EEE Trans. Haptics*, 5(4):289–300, 2012.

[24] 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*, 24(9):691–702, 2005.

[25] 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*, pages 27–33, Feb 2014.

[26] 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*, pages 217–224, 2002.

[27] 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*, pages 123–130, 1995.

[28] 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*, 5(3):252–263, 2012.

[29] 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*, pages 51:1–51:4, 2012.

[30] 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*, pages 4484–4490, 2013.

[31] R. Sodhi, I. Poupyrev, M. Glisson, and A. Israr. Aireal: Interactive tactile experiences in free air. *ACM Trans. Graph.*, 32(4):134:1–134:10, 2013.

[32] 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*, pages 31–36, 2011.

[33] 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*, pages 159–164, 2002.

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

[35] 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*, 6(2):167–180, 2013.

[36] 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*, pages 578–579, 2007.

[37] 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*, pages 13–17, 2009.

[38] 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*, volume 3, pages 146–151 vol.3, 1995.