Configuration-Based Optimization for Virtual Hand Haptic Simulation

Qianqian Tong, *Member, IEEE*, Qiong Wang, Yingkui Zhang, Xiangyun Liao, Wenxuan Wei, Yuru Zhang, *Senior Member, IEEE*, Jing Xiao, *Fellow, IEEE*, and Dangxiao Wang, *Senior Member, IEEE*

***Abstract*—Interacting with virtual objects via haptic feedback using the user's hand directly (virtual hand haptic interaction) provides a natural and immersive way to explore the virtual world. It remains a challenging topic to achieve 1 kHz stable virtual hand haptic simulation with no penetration amid hundreds of hand-object contacts. In this paper, we advocate decoupling the high-dimensional optimization problem of computing the *graphic-hand* configuration, and progressively optimizing the configuration of the graphic palm and fingers, yielding a decoupled-and-progressive optimization framework. We also introduce a method for accurate and efficient hand-object contact simulation, which constructs a virtual hand consisting of a sphere-tree model and five articulated cone frustums, and adopts a configuration-based optimization algorithm to compute the *graphic-hand* configuration under nonpenetration contact constraints. Experimental results show both high update rate and stability for a variety of manipulation behaviors. Nonpenetration between the graphic hand and complex-shaped objects can be maintained under diverse contact distributions, and even for frequent contact switches. The update rate of the haptic simulation loop exceeds 1 kHz for the whole-hand interaction with about 250 contacts.**

***Index Terms*—Decoupled-and-progressive optimization, haptic simulation, nonpenetration contact constraints, virtual hand interaction.**

Manuscript received 23 December 2021; revised 17 June 2022; accepted 31 July 2022. Date of publication 4 August 2022; date of current version 28 September 2022. This work was supported by the National Natural Science Foundation of China under Grants 62002185 and 61973016, and in part by the China Postdoctoral Science Foundation under Grant 2019M663018. This article was recommended for publication by Associate Editor Gionata Salvietti and Editor-in-Chief Marcia K O'Malley upon evaluation of the reviewers' comments. *(Corresponding author: Dangxiao Wang.)*

Qianqian Tong is with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518000, China, and also with the Department of Mathematics and Theories, Peng Cheng Laboratory, Shenzhen, Guangdong Province 518055, China (e-mail: tongqq@pcl.ac.cn).

Qiong Wang, Yingkui Zhang, and Xiangyun Liao are with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518000, China (e-mail: wangqiong@siat.ac.cn; yk.zhang1@siat.ac.cn; xyunliao@gmail.com).

Wenxuan Wei and Yuru Zhang are with the State Key Laboratory of Virtual Reality Technology and Systems and Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University, Beijing 100191, China (e-mail: sy2007609@buaa.edu.cn; yuru@buaa.edu.cn).

Jing Xiao is with the Department of Robotics Engineering and Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609 USA (e-mail: jxiao2@wpi.edu).

Dangxiao Wang is with the Department of Mathematics and Theories, Peng Cheng Laboratory, Shenzhen, Guangdong Province 518055, China, and also with the State Key Laboratory of Virtual Reality Technology and Systems and Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University, Beijing 100191, China (e-mail: hapticwang@buaa.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TOH.2022.3196625, provided by the authors.

Digital Object Identifier 10.1109/TOH.2022.3196625

I. Introduction

As an inherent demand of virtual reality technology, directly using the user's hand to grasp/manipulate virtual objects with haptic feedback (virtual hand haptic interaction, or hand-based haptic interaction) [1], [2] is desired to facilitate intuitive and immersive interaction experiences. Different from typical 3-DoF/6-DoF (degree of freedom) stylus-type devices used in tool-based haptic interaction [3]–[5], the human hand has a complex kinematic structure with more than 20 degrees of freedom (DoFs) [1], allowing the user to interact with virtual objects using diverse manipulation behaviors [6]. Due to the high-DoF characteristics of the hand motion, virtual hand interaction involves numerous contact points/regions and diverse contact distributions, making it complicated and time-consuming to achieve nonpenetration and stable virtual hand haptic simulation at a haptic rate, such as 1 kHz for high contact stiffness in a stable manner [7].

For the simulation of virtual hand haptic interaction, we usually capture the pose of the user's hand using hand-tracking devices (such as a data glove equipped with 6-DoF tracking function). We refer to the avatar of the user's hand in the virtual environment as the *haptic hand*, which is obtained by mapping the real hand to the virtual world according to the captured hand data. The *haptic hand* is invisible during the haptic interaction, and it can penetrate into virtual objects due to the absence of constraints. For realistic graphic display, the visual avatar is required not to penetrate into virtual objects. We define this visual avatar as the *graphic hand*. In terms of intuitive and immersive virtual hand haptic simulation, it is crucial to estimate the configuration of the *graphic hand* robustly for ensuring natural and stable haptic simulation with no hand-object penetration under various contact states [8]. In addition, it is a fundamental issue how to compute interaction forces efficiently to meet the requirement of high update rate for virtual hand haptic interaction.

A. Related Work

Many methods, mainly including heuristic-based and physics-based methods, have been explored to simulate the interaction between the virtual hand and virtual objects with/without haptic feedback.

Some heuristic-based methods were proposed to perform stable grasping for specific tasks. Zachmann et al. [9] used the

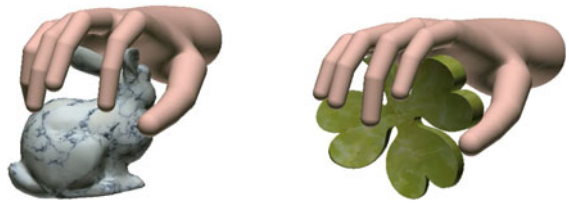


Fig. 1. Examples of complicated contact scenarios for hand-based haptic interaction.

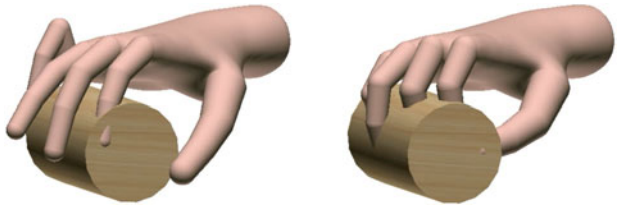


Fig. 2. Examples of abnormal situations when naively using the configuration-based optimization model in [25].

contact distribution among finger phalanges, thumb phalanges and the palm to discriminate three types of grasping manipulations (i.e., push, precision grasp, and power grasp), and grasps were maintained by considering the relative position and orientation of the object with respect to the palm. Moehring et al. [10] checked the contact condition between the virtual object and phalanxes through a ray test, and decoupled the finger-object interaction using a hierarchical grasp heuristic that prioritizes grasps with higher stability. In addition, grasping heuristics were further extended through the use of grasp pairs of fingers [11], which are determined by whether the line connecting collision points is included in the friction cones calculated with Normal Proxies.

Heuristic-based methods can handle the predefined grasping behaviors stably. Due to the sacrifice of physical reality during the contact simulation, heuristic-based methods will produce unrealistic hand-object contacts (such as interpenetration) in practical applications that usually involve various manipulation behaviors. To perform more physically accurate hand-object interaction, researches have explored many physics-based approaches in recent years. Next, we will introduce two main physics-based approaches, namely penalty-based methods and constraint-based methods for virtual hand interaction.

Penalty-based methods compute penalty forces/energies to push the *graphic hand* toward a nonpenetration configuration. To address the problem of visual hand-object interpenetration, Borst et al. [12], [13] coupled the simulated hand and the tracked configuration using linear and torsional spring-dampers, obtaining a few hundred updates for force rendering. Garre et al. [14] used a two-way viscoelastic virtual coupling [15] to connect the simulated proxy hand handle and the haptic device, and the coupling was linearized to avoid the expensive computation for hand-object contact simulation, achieving 1-kHz force computation. Verschoor et al. [16] designed an efficient energy minimization framework for simulating all the relevant mechanical and interaction elements of the virtual hand as well as hand-object contacts. This framework modeled constrains through penalty energies, obtaining a stable simulation for interactions between a soft hand and virtual objects at 45–65 Hz. Thanks to the local contact handling, penalty-based methods are computationally simple. As stated in [13], good grasp stability requires high friction coefficients, whereas the unelaborate design of such parameters could lead to excessive sticking. Moreover, numerical integration is required for stiff system to enhance stability, which will bring computational errors, making it challenging to simulate precise manipulations or interactions between a virtual hand and objects with complex geometric features (e.g., convex and concave features in Fig. 1) using penalty-based methods.

Constraint-based methods [17], [18] attempt to precisely solve all penetrations between objects in contact by solving a single computational problem, and to find contact forces that meet physical laws. Due to the computational complexity of such methods, it is challenging to meet the high-update-rate requirement of haptic simulation [7] for the virtual hand interaction that involves numerous contacts and various contact distributions. Multi-rate strategies help to deal with this problem. For example, Garre et al. [19] simulated the hand-object contact using a constraint-based method solved with an iterative algorithm [20] in a visual loop, and they set a six-dimensional viscoelastic coupling for each pair of tracked and proxy bones in the haptic loop, achieving a haptic update rate of several thousand Hertz with 1441-triangle mesh for collision detection. As the authors reported, excessive smoothing of haptic rendering occurred under limited visual thread updates [19]. Jacobs et al. [21] resorted to extending the God-object method [18] that works efficiently for 6-DoF haptic devices to support multi-finger interactions, which avoided deep virtual hand-object penetrations with about 600 Hz for a few dozens of contacts and dozens of Hertz for a hundred contacts. In order to improve the computation of contact mechanics involving many contact points, Talvas et al. [22] introduced volume contact constraints [23] and aggregated all contacts on each phalanx into a minimal set of constraints, and one-hand (15 phalanges) grasping with about 80 contacts could attain about 25 Hz. To achieve interactive play with a virtual music instrument, Wang et al. [24] achieved 6-DoF haptic rendering with 1 kHz update for the interactions between a hand with a fixed gesture and a stringed musical instrument by extending configuration-based optimization methods [24]–[26]. When applying such method to the natural hand-based interaction involving almost 20 DoFs and hundreds of contacts, the update rate is less than 20 Hz and abnormal situations may occur, such as malformed posture, joint angle jump and hand-object penetrations depicted in Fig. 2.

B. Contributions of This Work

As surveyed above, it is still challenging to reproduce penetration-free hand-object interactions involving multi-region and numerous contacts at a 1-kHz haptic update rate. This paper presents three contributions for hand-based haptic simulation, aiming to deal with various contacts between the whole hand (i.e., including the palm and fingers) and complex-shaped objects with a 1-kHz haptic update rate.

First, we introduce a decoupled-and-progressive optimization framework for virtual hand haptic simulation. The high-dimensional optimization problem of computing the *graphic-hand* configuration (21 DoFs in this work) is decoupled into a 6-DoF optimization problem (for the palm) plus five 3-DoF optimization problems (for each finger). The configuration of the *graphic-hand* is then optimized progressively, in which the palm configuration is solved before solving five-finger configurations. This decoupled-and-progressive framework avoids complex high-dimensional calculations, enabling efficient solution of a plausible *graphic-hand* configuration.

Second, we construct a hybrid virtual hand model, which represents the palm and fingers using a sphere-tree model and articulated cone frustums, respectively. The high-level representation of fingers facilitates efficient contact handling, ensuring stable simulation of simultaneous multi-region contacts between the virtual hand and complex-shaped objects. This hybrid hand model preserves the characteristic of multi-finger independent motion of the human hand, allowing decoupling of the complex optimization problem for the calculation of *graphic-hand* configuration.

Third, we present an efficient and stable hand-object contact handling approach based on our hybrid virtual hand model, mainly including a contact detection algorithm to accurately determine finger-object contact states, and a configuration optimization method by constructing nonpenetration finger-object contact constraints. By introducing this hand-object contact handling approach into our decoupled-and-progressive optimization framework, we can perform virtual hand haptic simulation efficiently. Experimental results show stable force feedback with an update rate of 1 kHz, and there is no perceptible penetration between the *graphic hand* and virtual objects under various contact states.

II. Decoupled-and-Progressive Optimization Framework

For hand-based haptic simulation, we use the following model to compute the feedback force and torque $\mathbf{F}$:

$$\mathbf{F}^t = \mathbf{G}\left(\mathbf{Q}_g^t - \mathbf{Q}_h^t\right), \tag{1}$$

where $\mathbf{G}$ is a diagonal stiffness matrix, and t denotes the current time. $\mathbf{Q}_g^t$ and $\mathbf{Q}_h^t$ are the configuration[1] of the *graphic hand* and *haptic hand*, respectively. $\mathbf{Q}_h^t$ is obtained by using a measurement equipment (such as a data glove equipped with 6-DoF tracking), and $\mathbf{Q}_g^t$ can be solved following the configuration-based optimization approach [25]:

$$\begin{cases} Min : \frac{1}{2}\left(\mathbf{Q}_g^t - \mathbf{Q}_h^t\right)^T \mathbf{G}\left(\mathbf{Q}_g^t - \mathbf{Q}_h^t\right), \\ Subject\ to : V_H \cap V_O = \emptyset \end{cases} \tag{2}$$

where V_H and V_O denote the spatial boundary of the geometrical elements (such as mesh, sphere, etc.) of the *graphic hand* and virtual object, respectively. $V_H \cap V_O = \emptyset$ represents that there are no overlaps between the geometrical elements of the *graphic hand* and the virtual object. This constraint is used to maintain no perceptible hand-object penetration when a hand-object contact occurs.

For the whole-hand haptic simulation in this work, $\mathbf{Q}_g^t$ consists of a 6-DoF palm configuration $\boldsymbol{q}_g^t$ (3-DoF translation and 3-DoF rotation) and five-finger configurations. For each finger, the configuration $\boldsymbol{\theta}_g^t$ includes three variables (joint angles) $\theta_{1\,g}^t, \theta_{2\,g}^t$ and $\theta_{3\,g}^t$, which represent the metacarpophalangeal joint angle, proximal phalangeal joint angle and distal phalangeal joint angle of a finger, respectively. A total of 21 variables need to be optimized in (2), which is not trivial to achieve under multiple contact constraints at a haptic update rate. Essentially, the number of hand-object contacts increase as the number of fingers in contact with the virtual object increases, which makes it difficult to compute a plausible configuration of the *graphic hand* efficiently.

The human hand can be regarded as an articulated object consisting of multiple subobjects (i.e., a palm and five fingers). We model the palm as the parent node of five fingers to represent the kinematic characteristic between the palm and fingers. Once the palm configuration is determined, the configuration of the *graphic hand* can be obtained by computing joint angles of five fingers. According to the above observation, we introduce a decoupled-and-progressive configuration-based framework for hand-based haptic simulation. As shown in Fig. 3, we construct a hybrid virtual hand model consisting of cone-frustum fingers and a sphere-tree palm (Section III). Inspired by the kinematic characteristic of the human hand, we decouple the configuration optimization of the *graphic hand* to the independent computation of the palm and five fingers based on the configuration-based optimization method, i.e. first computing the palm configuration and then progressively five fingers. As shown in Fig. 4 (a), the palm and one finger of the *haptic hand* penetrate into the object (i.e. the light green hexagonal object represented using a sphere-tree model). By performing collision detection and configuration-based optimization (Fig. 3), we first compute the graphic palm configuration $\boldsymbol{q}_g^t$ that is in exact contact with the object surface (Fig. 4(b)), whereas the haptic finger still penetrates into the object. In the second step, the configuration of the haptic finger $\boldsymbol{\theta}_h^t(\theta_{1\,h}^t, \theta_{2\,h}^t, \theta_{3\,h}^t)$ are optimized to obtain the configuration of the graphic finger $\boldsymbol{\theta}_g^t(\theta_{1\,g}^t, \theta_{2\,g}^t, \theta_{3\,g}^t)$, which maintains exact contact with the object (Fig. 4(c)).

In our decoupled-and-progressive optimization framework, the number of variables to be optimized and the number of contact constraints are reduced for each optimization task, helping to find a plausible solution of the *graphic hand* efficiently and accurately. Due to the mutual independence of five fingers during the interaction, the nonpenetration configuration of the *graphic hand* can be solved more quickly through a parallel solution, such as a multi-thread calculation method (mentioned in Section V).

III. Hybrid Virtual Hand Model

A. Structure of Human Hand

The human hand [27], [28] consists of ligaments, tendons, and muscles. The thumb has two bones (distal and proximal

[1] In this article, we model a hybrid virtual hand with 21 DoFs.

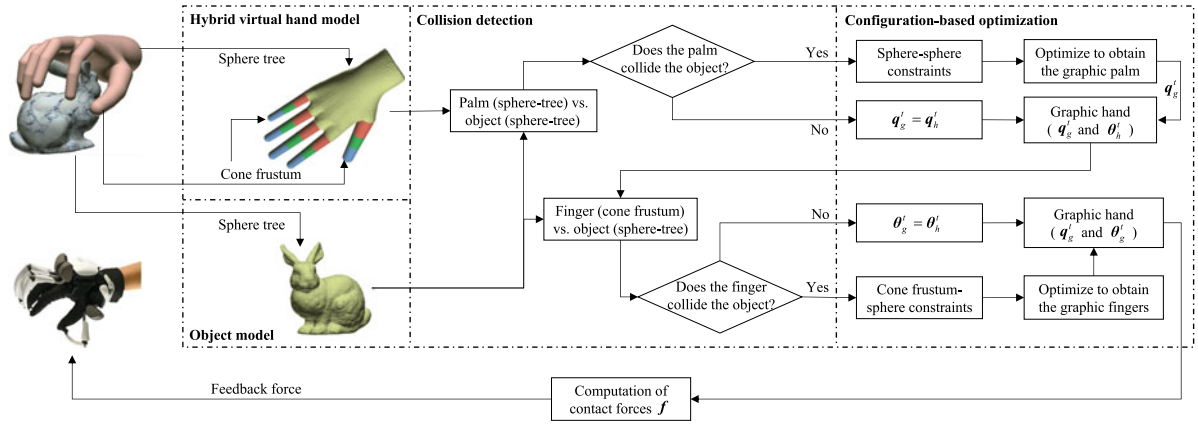


Fig. 3. Flowchart of our decoupled-and-progressive configuration-based optimization approach for hand-based haptic simulation.

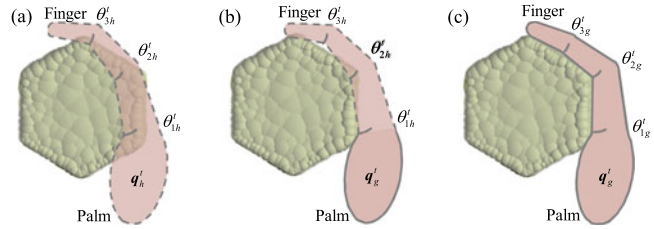


Fig. 4. The progressive optimization scheme for computing the plausible configuration of the *graphic hand*. The dashed regions denote the hand structures (i.e., finger or/and palm) to be optimized, and the solid ones are optimal configuration. Note that the palm in this figure is simplified to an ellipse, and we represent the palm using a sphere-tree model in this article.

phalanges) and two joints (Metacarpophalangeal-MCP and distal interphalangeal-DIP joints), whereas each of the other four fingers has three bones (distal, middle, and proximal phalanges), and three joints (MCP, DIP, and proximal interphalangeal-PIP joints). All five fingers can bend inward (i.e. flexion and extension) and swing slightly from side to side (i.e. adduction and abduction). The metacarpal of the thumb has rotational freedom, and the movement of the remaining four metacarpals is very small. The human palm is also a complex object consisting of five metacarpals and eight carpal bones.

The complex kinematic structure of the human hand allows it to flexibly manipulate various objects, which produces various contact distributions that need to be dealt with efficiently and stably for natural virtual hand haptic interaction. Building a high-resolution anatomical model is computationally expensive for haptic interaction simulations. Simplifications are generally adopted to keep models suitable for specific needs [28]. Likewise, we construct a hybrid virtual hand to represent the human hand in a simplified way for efficient computation.

B. Construction of Hybrid Virtual Hand Model

We build an articulated hand model consisting of a rigid palm and five rigid fingers. Considering the limited movement of four metacarpals other than that of the thumb, we model a virtual palm without the thumb metacarpal. Existing studies have shown the simplicity and efficiency of sphere-tree models [29] for haptic simulation [24], [25], and thus we adopt a sphere-tree model to represent the complex-shaped palm by Bradshaw's sphere-tree tool-kit[2]. Fig. 5 shows the palm modeled using multi-level sphere trees.

The phalanges of the human hand are not regular cylinders, and we use articulated cone-frustum models to approximate the characteristic of different thicknesses of real fingers, and one cone frustum for a phalanx[3] (Fig. 5). For each finger, a sphere is used to represent the joint between adjacent phalanges, and one hemisphere is connected to the distal-phalanx cone frustum to represent the fingertip of the finger. As mentioned above, the sphere-tree palm model does not contain the thumb metacarpal, and we regard the thumb metacarpal as a phalanx and represent it using a cone-frustum model like other phalanges, which is just similar to the design of some robotic hands.

To avoid the *graphic hand* penetrating virtual objects during the haptic interaction, we use sphere-tree models to construct virtual objects, and perform nonpenetration contact handling between the hybrid virtual hand and virtual objects. A contact constrained optimization method is used for palm-object contact handling based on a hierarchical sphere-sphere collision detection algorithm, which determines the contact states between the palm and virtual objects by comparing the distance between the centers of two spheres and the sum of their radii [30], [31]. In the following, we will describe how to determine the finger-object contacts, and how to optimize the finger configuration of *graphic hand*. Due to the absence of haptic feedback in the adduction/abduction direction for existing haptic gloves, the joint angles of fingers in this direction are not optimized in this paper, and these joint angles of the *graphic hand* are the same as that of the *haptic hand*. Our method is verified to be efficient and stable for haptic interactions with no lateral swing (please see Section V).

IV. Hand-Object Contact Handling Approach

A. Hand-Object Collision Detection

1) Palm-Object Collision Detection: In this paper, we use sphere-tree models to construct virtual objects [25]. A

[2] [Online]. Available: http://isg.cs.tcd.ie/spheretree/.
[3] In this article, a phalanx is modeled as a rigid body.

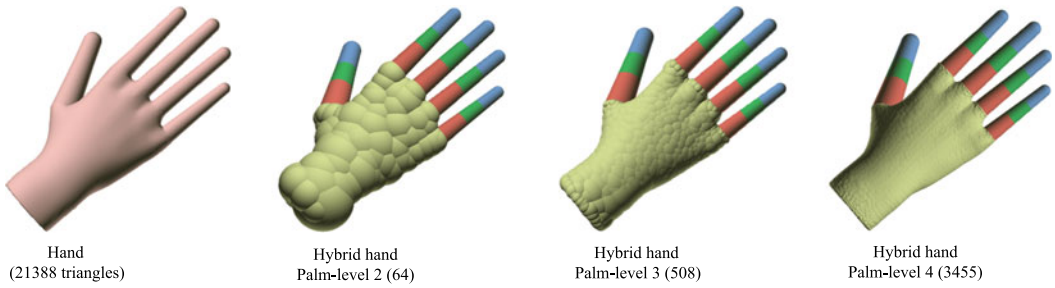


Fig. 5. Hand models. The left is a hand model with 21388 triangles, and the rest are hybrid hands consisting of articulated cone-frustum fingers and a palm modeled using different-level sphere trees.

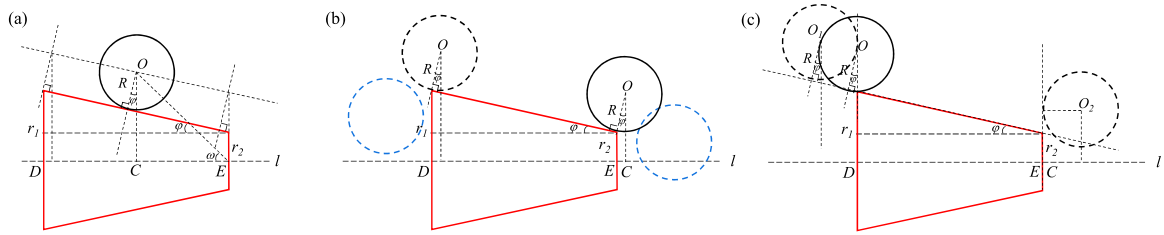


Fig. 6. Contact cases between a sphere and a cone frustum. (a) The back solid sphere is tangent to the side of the cone frustum. (b) The black solid and dotted spheres are just tangent to the side of the cone frustum, and the blue spheres are not tangent to the cone frustum. (c) The center of the black solid sphere is on the (extended) lower bottom, and the black dotted sphere is tangent to the (extended) upper/lower bottom of the cone frustum.

hierarchical sphere-sphere collision detection algorithm is adopted to determine the contact state between the palm and virtual objects, and the palm is in contact with the object if the sphere on the palm and the sphere on the virtual object meet the following condition:

$$(x_H - x_O)^2 + (y_H - y_O)^2 + (z_H - z_O)^2 \leq (r_H - r_O)^2 \quad (3)$$

where (x_H, y_H, z_H) and (x_O, y_O, z_O) represent the center coordinates of spheres on the palm and virtual object in global system, respectively. r_H and r_O are radii of these two spheres.

2) Finger-Object Collision Detection: As mentioned in Section II, our decoupled-and-progressive optimization framework conducts contact handling for each finger independently, and collision detection is performed for each phalanx of one finger. As shown in Fig. 6, the black solid sphere denotes a sphere on the virtual object and the red trapezoid (i.e., a longitudinal section of the cone frustum along its central axis l, called cone frustum in the following content) denotes a phalanx. The point C is the foot point from the sphere center O to the central axis l of the cone frustum. D and E denote the center points of the two bottoms on the cone frustum, and their coordinates are (x_D, y_D, z_D) and (x_E, y_E, z_E), respectively. We define the distance between the sphere center O and the central axis l as $\overrightarrow{OC}$. The contact state between the phalanx and the virtual object can be determined by comparing the current $|\overrightarrow{OC}|$ and the threshold $|\overrightarrow{OC}|_\Gamma$, and we calculate $|\overrightarrow{OC}|_\Gamma$ when the sphere is exactly tangent to the side or its extension of the cone frustum (Fig. 6(a)). If $|\overrightarrow{OC}| \leq |\overrightarrow{OC}|_\Gamma$ is satisfied, the collision between the two (i.e., the black solid sphere and the red cone frustum) can be determined. We use the following equation to compute $|\overrightarrow{OC}|$:

$$\left|\overrightarrow{OC}\right|^2 = \frac{\left|\overrightarrow{OD} \times \overrightarrow{DE}\right|^2}{\left|\overrightarrow{DE}\right|^2} = \frac{l_1^2 + l_2^2 + l_3^2}{\left|\overrightarrow{DE}\right|^2}, \quad (4)$$

where $|\overrightarrow{DE}| = \sqrt{(x_E - x_D)^2 + (y_E - y_D)^2 + (z_E - z_D)^2}$, denoting the distance between the point D and E. $l_1 = \begin{vmatrix} y_D - y_O & z_D - z_O \\ y_E - y_D & z_E - z_D \end{vmatrix}$, $l_2 = \begin{vmatrix} z_D - z_O & x_D - x_O \\ z_E - z_D & x_E - x_D \end{vmatrix}$, $l_3 = \begin{vmatrix} x_D - x_O & y_D - y_O \\ x_E - x_D & y_E - y_D \end{vmatrix}$, and $|\cdot|$ denotes the determinant of a vector.

For the calculation of $|\overrightarrow{OC}|_\Gamma$, there are three tangent states to be considered, including point C on line segment ED, on the extension of line segment ED, and on the extension of line segment DE. Supposing that the radii of the lower and upper bottom of the cone frustum are r_1 and r_2 ($r_2 < r_1$), we can calculate $|\overrightarrow{OC}|_\Gamma$ using the following formula for the above three states:

$$\left|\overrightarrow{OC}\right|_\Gamma = \frac{R}{\cos\varphi} + k * (r_1 - r_2) + r_2, \quad (5)$$

where R is the radius of the sphere, φ denotes the angle between the side and the central axis of the cone frustum, and k is used to characterize where point C locates on the central axis l of the cone frustum. We calculate k according to the relationship between the magnitude and direction of collinear vectors $\overrightarrow{CE}$ and $\overrightarrow{DE}$:

$$k = \frac{\overrightarrow{CE}}{\overrightarrow{DE}} = \frac{\left|\overrightarrow{OE}\right| \cos\omega}{\left|\overrightarrow{DE}\right|} = \frac{\overrightarrow{OE} \cdot \overrightarrow{DE}}{\left|\overrightarrow{DE}\right|^2}, \quad (6)$$

where $\overrightarrow{OE} \cdot \overrightarrow{DE} = (x_O - x_E)(x_D - x_E) + (y_O - y_E)(y_D - y_E) + (z_O - z_E)(z_D - z_E)$. ω is the angle between vectors $\overrightarrow{OE}$ and $\overrightarrow{DE}$ (please see Fig. 6(a)).

As shown in Fig. 6(b), although the condition $|\overrightarrow{OC}| \leq |\overrightarrow{OC}|_\Gamma$ is satisfied, the blue spheres are not in contact with the cone frustum. More specifically, the sphere will not contact the cone frustum when $k > 1 + R/|\overrightarrow{DE}|$ or $k < -R/|\overrightarrow{DE}|$. The sphere is tangent to the (extended) lower bottom of the cone frustum when $k = 1 + R/|\overrightarrow{DE}|$ (i.e., the black dotted sphere O_1 in Fig. 6(c)), and the sphere is tangent to the (extended) upper bottom of the cone frustum when $k = -R/|\overrightarrow{DE}|$ (i.e., the black dotted sphere O_2 in Fig. 6(c)). In addition, the foot point C coincides the endpoint E when $k = 0$, and the foot point C coincides the endpoint D when $k = 1$. Overall, $-R/|\overrightarrow{DE}| < k < 1 + R/|\overrightarrow{DE}|$ is another condition that should be satisfied to determine the contact state between the sphere and the cone frustum. Therefore, the collision between the phalanx and the virtual object occurs under the following condition:

$$\begin{cases} \left|\overrightarrow{OC}\right|^2 \leq \left(\frac{R}{\cos\varphi} + k * (r_1 - r_2) + r_2\right)^2 \\ k \in [-\frac{R}{\left|\overrightarrow{DE}\right|}, 1 + \frac{R}{\left|\overrightarrow{DE}\right|}] \end{cases}. \tag{7}$$

B. Hand-Object Collision Response

To support hand-based haptic simulation involving multi-finger/whole-hand and multi-region contacts, we compute the plausible configuration of the *graphic hand* by constructing nonpenetration constraints.

1) Palm-Configuration Optimization: We construct the nonpenetration constraint in (2) using sphere-sphere contact constraints and compute the palm configuration of the *graphic hand* following the previous work [25]. According to (3), the sphere-sphere contact constraint denotes that the distance between the centers of two spheres is not smaller than the sum of their radii. In this paper, $\boldsymbol{q}_g^t$ and $\boldsymbol{q}_h^t$ denote the palm configuration of the *graphic hand* and *haptic hand*, where $\boldsymbol{q}_g^t = (x_g^t, y_g^t, z_g^t, \alpha_g^t, \beta_g^t, \gamma_g^t)$ and $\boldsymbol{q}_h^t = (x_h^t, y_h^t, z_h^t, \alpha_h^t, \beta_h^t, \gamma_h^t)$. The first three and last three coordinates are respectively translational and rotational variables in the world coordinate system, and α^t, β^t and γ^t refer to the rotation angles around axes X, Y and Z.

There is a unique mapping relationship between each sphere on the virtual palm and the configuration of the virtual palm $\boldsymbol{q}_g^t$. We adopt the coordinate transformation method mentioned in [25] to represent the coordinate $\mathbf{x}_W = (x_i^t, y_i^t, z_i^t)$ of the i^{th} sphere center on the virtual palm in the world coordinate system W, and $\mathbf{x}_W = \mathbf{T}_L^W \cdot \mathbf{x}_L$, where $\mathbf{x}_L = (x_i^0, y_i^0, z_i^0)$ is the local coordinate of the i^{th} sphere center in the local coordinate system L, and $\mathbf{T}_L^W = \begin{bmatrix} \mathbf{R}_L^W & \mathbf{p}_L^W \\ 0 & 1 \end{bmatrix}$, where $\mathbf{p}_L^W = [x_g^t \ y_g^t \ z_g^t]^T$, and $\mathbf{R}_L^W = \begin{bmatrix} c\gamma_g^t c\beta_g^t & c\gamma_g^t s\beta_g^t s\alpha_g^t - s\gamma_g^t c\alpha_g^t & c\gamma_g^t s\beta_g^t c\alpha_g^t + s\gamma_g^t s\alpha_g^t \\ s\gamma_g^t c\beta_g^t & s\gamma_g^t s\beta_g^t s\alpha_g^t + c\gamma_g^t c\alpha_g^t & s\gamma_g^t s\beta_g^t c\alpha_g^t - c\gamma_g^t s\alpha_g^t \\ -s\beta_g^t & c\beta_g^t s\alpha_g^t & c\beta_g^t c\alpha_g^t \end{bmatrix}$. Therefore, the contact constraints in (2) can be expressed as the following:

$$C_i\Big(x_i(\boldsymbol{q}_g^t), y_i(\boldsymbol{q}_g^t), z_i(\boldsymbol{q}_g^t)\Big) \geq 0, i = 1, 2, \ldots, N, \tag{8}$$

where N denotes the number of contact sphere pairs.

We linearize the contact constraints in (8) using a first-order Taylor expansion, that is $C_i(x_i(\mathbf{q}_g^t), y_i(\mathbf{q}_g^t), z_i(\mathbf{q}_g^t)) \approx C_i(x_i(\mathbf{q}_g^{t-1}),\ y_i(\mathbf{q}_g^{t-1}),\ z_i(\mathbf{q}_g^{t-1})) + \sum_{j=1}^{6} (\frac{\partial C_i}{\partial x_i}\frac{\partial x_i}{\partial (\mathbf{q}_g^t)_j} + \frac{\partial C_i}{\partial y_i}\frac{\partial y_i}{\partial (\mathbf{q}_g^t)_j} + \frac{\partial C_i}{\partial z_i}\frac{\partial z_i}{\partial (\mathbf{q}_g^t)_j}) \cdot (\mathbf{q}_g^t - \mathbf{q}_g^{t-1})$, aiming to speed up the optimization and solve the palm configuration of the *graphic hand* using the active-set method [25], [32]. In the optimization process, we iteratively solve the translational variables (x_g^t, y_g^t, z_g^t). Considering that the used force-feedback glove (Dexmo) in this paper cannot provide force feedback to the palm, we directly use the palm configuration $(\alpha_h^t, \beta_h^t, \gamma_h^t)$ of the *haptic hand* as the rotational variables $(\alpha_g^t, \beta_g^t, \gamma_g^t)$ of the *graphic hand*, aiming at avoiding the discordance between the graphic display and the configuration of the user's real hand.

2) Finger-Configuration Optimization: Cone-frustum articulated models are utilized to represent the fingers of the hybrid virtual hand, and cone frustum & sphere contact constraints should be constructed to optimize the finger configuration $\boldsymbol{\theta}_g^t(\theta_{1_g}^t, \theta_{2_g}^t, \theta_{3_g}^t)$ of the *graphic hand*. These three joint angels are optimized as a whole, and the optimization objective can be written as the following:

$$Min: \frac{1}{2}\Big(\boldsymbol{\theta}_g^t - \boldsymbol{\theta}_h^t\Big)^T \mathbf{G}_r \Big(\boldsymbol{\theta}_g^t - \boldsymbol{\theta}_h^t\Big), \tag{9}$$

where $\mathbf{G}_r$ is a 3×3 diagonal torsional stiffness matrix. According to the finger-object collision detection algorithm described in Section IV-A2, we construct nonpenetration finger-object contact constraints as the following:

$$\left|\overrightarrow{OC}\right|_j^2 - \left(\frac{R_j}{\cos\varphi} + k_j * (r_1 - r_2) + r_2\right)^2 \geq 0, \tag{10}$$

where $j = 1, 2, \ldots, M$, and M refers to the number of cone frustum & sphere contact pairs. $|\overrightarrow{OC}|_j$ represents the distance from the center $O(x_{Oj}, y_{Oj}, z_{Oj})$ of the j^{th} sphere to the central axis l of the cone frustum, R_j denotes the radius of the j^{th} sphere, and k_j represents the relationship between the magnitude and direction of collinear vectors $\overrightarrow{C_jE}$ and $\overrightarrow{DE}$.

After obtaining the optimized palm configuration $\boldsymbol{q}_g^t$ of the *graphic hand*, we can compute the world coordinate $\mathbf{H}_M$ of the point on the finger through matrix transformations, such as the point on the distal phalanx $\mathbf{H}_M = \mathbf{T}_L^W \mathbf{M}_1(\theta_{1_g}^t, T_1)\mathbf{M}_2(\theta_{2_g}^t, T_2)\mathbf{M}_3(\theta_{3_g}^t, T_3)\mathbf{H}_L$, where $\mathbf{H}_L$ is the local coordinate. $\mathbf{T}_L^W$ is the homogeneous transformation matrix mentioned above, which is derived from the graphic palm configuration $\boldsymbol{q}_g^t$. T_1, T_2 and T_3 are translation distance relative to their respective parent nodes, and they are respectively determined by the length of proximal phalanx distal L_1, the length of middle phalanx L_2, and the length of distal phalanx L_3. $\mathbf{M}_1(\theta_{1_g}^t, T_1)$ denotes the homogeneous transformation matrix of the proximal joint coordinate system with respect to the metacarpal joint coordinate system, $\mathbf{M}_2(\theta_{2_g}^t, T_2)$ is the homogeneous transformation matrix of

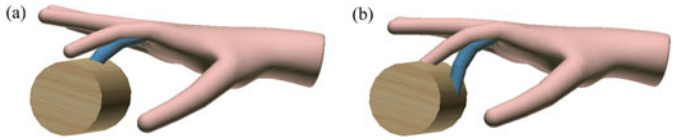


Fig. 7. Abnormal fingertip manipulations with poor fingertip-object contact constraints. (a) The virtual fingertip suspends above the virtual object. (b) The virtual fingertip penetrates into the virtual object. Skin color represents the optimized *graphic hand*, and blue represents the *haptic hand*.

the middle joint coordinate system with respect to the proximal joint coordinate system, and $\mathbf{M}_3(\theta^t_{3\,g}, T_3)$ is the homogeneous transformation matrix of the distal joint coordinate system with respect to the middle joint coordinate system. We compute the homogeneous transformation matrix $\mathbf{M}_{i_0}(i_0 = 1g, 2g, 3g)$ as follows:

$$\mathbf{M}_{i_0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta^t_{i_0} & -\sin\theta^t_{i_0} & 0 \\ 0 & \sin\theta^t_{i_0} & \cos\theta^t_{i_0} & L_{i_0} \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{11}$$

For a certain phalanx, the cone-frustum parameters r_1, r_2 and φ are constant. With the above coordinate transformation method, we can obtain the coordinates of two endpoints, i.e. $D(x_D(\boldsymbol{\theta}^t_g), y_D(\boldsymbol{\theta}^t_g),\ z_D(\boldsymbol{\theta}^t_g))$ and $E(x_E(\boldsymbol{\theta}^t_g), y_E(\boldsymbol{\theta}^t_g), z_E(\boldsymbol{\theta}^t_g))$. From (4), the calculation of $|\overrightarrow{OC}|_j$ in (10) is related to the coordinates of points D and E. Similar to (8), the cone frustum & sphere contact constraints in (10) can be therefore represented implicitly as the following:

$$\begin{aligned} &C_j(x_D(\boldsymbol{\theta}^t_g), y_D(\boldsymbol{\theta}^t_g), z_D(\boldsymbol{\theta}^t_g), \\ &\quad x_E(\boldsymbol{\theta}^t_g), y_E(\boldsymbol{\theta}^t_g), z_E(\boldsymbol{\theta}^t_g)) \geq 0. \end{aligned} \tag{12}$$

Human fingertips are adept at performing diverse manipulations, such as sliding, touching, and squeezing. In order to simulate fingertip manipulations stably and naturally, it is necessary to construct appropriate fingertip-object contact constraints. Otherwise, the virtual fingertip may suspend above (Fig. 7(a)) or penetrate (Fig. 7(b)) into the virtual object.

In our hybrid virtual hand model, each distal phalanx is composed of a cone frustum and a hemisphere, and the hemisphere is connected to the cone frustum to represent the fingertip (Fig. 8(a)). In order to maintain optimal fingertip manipulations, we jointly consider the contact between the cone frustum and object spheres (called cone frustum-sphere contact) as well as the contact between the fingertip hemisphere and object spheres (called hemisphere-sphere contact).

The black solid sphere depicted in Fig. 8(a) is exactly tangent to the side of the cone frustum at the connection point between the cone frustum and the hemisphere, in which case $k = -R * \sin\varphi / |\overrightarrow{DE}|$. We add this k value as one condition to determine the contact state between the distal phalanx and the virtual object. According to this modified contact condition, the contact constraints between the distal phalanx and the virtual object can be reformulated as the following:

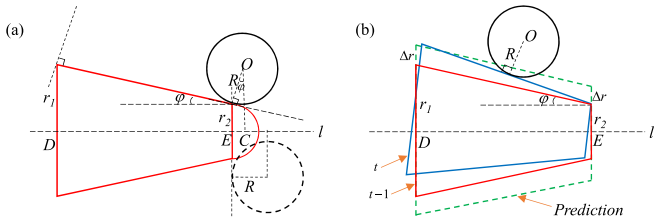


Fig. 8. Schematic diagram of the phalanx-object contact constraint. (a) The black solid sphere is exactly tangent to the side of the cone frustum at the connection point between the distal-phalanx cone frustum and the hemisphere, and the black dotted sphere is tangent to the extended upper bottom of the cone frustum. (b) Schematic diagram of the phalanx-object contact handling based on contact constraint prediction method.

$$\begin{cases} \begin{cases} C^1_{i_1}(x_D, y_D, z_D, x_E, y_E z_E) \geq 0 \\ k \in [-\frac{R*\sin\varphi}{|\overrightarrow{DE}|}, 1 + \frac{R}{|\overrightarrow{DE}|}], i_1 = 1, 2, \ldots, N_1 \end{cases} \\ \begin{cases} C^2_{i_2}(x_E, y_E, z_E) \geq 0, \\ k \in [-\frac{R+r_2}{|\overrightarrow{DE}|}, -\frac{R*\sin\varphi}{|\overrightarrow{DE}|}), i_2 = 1, 2, \ldots, N_2 \end{cases} \end{cases}, \tag{13}$$

where $C^1_{i_1}$ and $C^2_{i_2}$ can be obtained by using cone-frustum & sphere contact constraints (i.e., (11)) and sphere & sphere contact constraints, respectively. N_1 denotes the number of cone frustum & sphere contact pairs between the distal phalanx and the virtual object, and N_2 denotes the number of sphere & sphere contact pairs.

There will be two issues for hand-based haptic simulation when the fingers of the *haptic hand* deeply penetrate into the virtual object. 1) This will lead to a large number of constraints, as mentioned in previous work [25], which is not conducive to efficient contact handling, and 2) the constraints related to surface spheres of the virtual object can be missed, which will lead to potential hand-object penetrations.

Inspired by the contact constraint-prediction algorithm in [25], we resort to using the configuration of the *graphic hand* rather than the *haptic hand* during the contact handling. As shown in Fig. 8(b), the red cone frustum represents a phalanx of the *graphic hand* at the previous moment $t-1$. We grow the red cone frustum at time $t-1$ by Δr ($\Delta r = 0.5$ in this paper), and the green line shows the enlarged area. Next, we find the cone frustum & sphere contact pairs using the collision detection algorithm described in Section IV-A2. These intersecting cone frustum & sphere pairs are then used to form the contact constraints according to (10). With these contact constraints, we can compute the penetration-free configuration of the phalanx at time t, such as the blue cone frustum depicted in Fig. 8(b).

We linearize the contact constraints using Taylor expansion and compute the finger configuration $\boldsymbol{\theta}^t_g(\theta^t_{1\,g}, \theta^t_{2\,g}, \theta^t_{3\,g})$ of the *graphic hand* using the active-set method [25], [32].

C. Force Modeling

After obtaining the nonpenetration configuration of the *graphic hand*, the contact forces between the virtual hand and the virtual object are calculated according to their contact state. Taking a phalanx as an example, there are N_3 spheres in contact with the phalanx. Suppose the contact point between the graphic

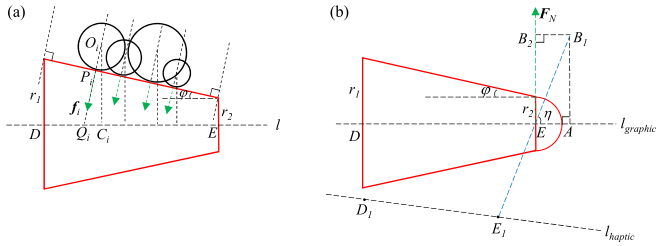


Fig. 9. Schematic diagram of the force computation for the virtual hand interaction: (a) the contact force between the finger and the object; (b) the normal force applied to the fingertip.

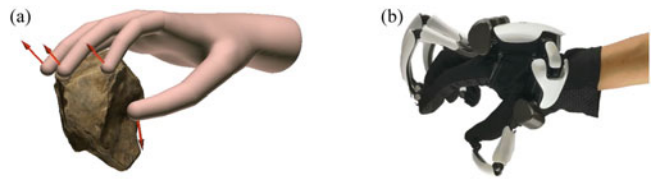


Fig. 10. Hand-based haptic interaction with a Dexmo force-feedback glove. (a) A hand-object haptic interaction scenario. (b) A user wears a Dexmo glove to experience the feedback forces exerted at fingertips.

phalanx and the i^{th} sphere (sphere center O_i) is P_{ig}), and its corresponding point on the haptic phalanx is P_{ih}.

As shown in Fig. 9(a), the direction of the normal force $\boldsymbol{f}_i$ is the same as that of vector $\overrightarrow{O_iP_{ig}}$. We calculate $\boldsymbol{f}_i$ using the following model:

$$\boldsymbol{f}_i = G_i \cdot \mathbf{n} \cdot \left|\mathbf{p}_{ig} - \mathbf{p}_{ih}\right|, \tag{14}$$

where G_i is the stiffness coefficient, and $\mathbf{n} = \overrightarrow{O_iP_{ig}}/|\overrightarrow{O_iP_{ig}}|$. $\mathbf{p}_{ig}$ and $\mathbf{p}_{ih}$ denote the global coordinates of point P_{ig} and P_{ih}, and $|\mathbf{p}_{ig} - \mathbf{p}_{ih}|$ is the length of $\overrightarrow{P_{ig}P_{ih}}$.

In Fig. 9(a), C_{ig} is the foot point from the sphere center O_i to the central axis l of the cone frustum, and Q_{ig} is the intersection point between the extension line of $\overrightarrow{O_iP_{ig}}$ and the central axis l. We can obtain the global coordinate of point Q_{ig} and P_{ig} according to the radius R_i of the i^{th} contact sphere and the ratio k between vector $\overrightarrow{CE}$ and vector $\overrightarrow{DE}$.

With the computed ratio between vector $\overrightarrow{Q_{ig}E}$ and vector $\overrightarrow{DE}$, we can successively obtain the global coordinate of point Q_{ih} and the global coordinate of point P_{ih}. Note that the parameters (i.e., the coordinates of D and E, and the radii of the upper and lower bottom r_2 and r_1) used for the computation of the global coordinates of point Q_{ih} and P_{ih} are from the corresponding haptic phalanx.

According to the contact force model detailed above, we calculate the normal force $\boldsymbol{F}_N$ that should be exerted at the fingertips during the hand-object interaction. Fig. 9(b) shows a calculation method of $\boldsymbol{F}_N$. Note that $\overrightarrow{E_1E}=\overrightarrow{EB_1}$, $\mathbf{V}_{B_1B_2} \parallel l_{graphic}$ and $\mathbf{V}_{AB_1} \perp l_{graphic}$. The magnitude of $\boldsymbol{F}_N$ is proportional to the distance between the graphic fingertip and the corresponding haptic fingertip (i.e., $\overrightarrow{EE_1}$), and the direction of $\boldsymbol{F}_N$ is perpendicular to the central axis $l_{graphic}$ of the first phalanx of the finger (i.e., $\mathbf{n} = \overrightarrow{EB_2}/\overrightarrow{EB_2}$, and $\overrightarrow{EB_2}=\overrightarrow{E_1E} + k\overrightarrow{DE}$).

V. Experiments

A Dexmo force-feedback glove [33] is utilized to provide feedback forces at five fingertips in our hand-object interaction experiments (as shown in Fig. 10). The device only tracks the metacarpophalangeal joint on each finger directly using rotational sensors, which provides 11-DoF hand tracking with three DoFs for the thumb and two DoFs for each of the other four fingers. With the assumption that there exists a certain relationship between rotation angles at each finger joint when clenching a fist, the joint angles of each finger can be obtained using the linear regression method in [33]. We use several keys on the keyboard to control the position and orientation of the palm and objects to obtain different contact states between the virtual hand and the virtual object. The specifications of the computer are as follows: Intel(R) Core(TM) i7-8700 @ 3.70 GHz, 16 GB memory, NVIDIA GeForce RTX 2080 graphical card. Note that no GPU is used in our haptic simulation method. We validate our method in terms of the following three performance metrics:

(1) *Nonpenetration*. There should be no interpenetrations between the hand avatar and the virtual object, not only for a single hand-object contact point/region, but also for multiple contact points/regions, as well as for diverse manipulation behaviors.

(2) *Stability*. The manipulation behaviors of the optimized *graphic hand* should be smooth and natural, and the calculated contact forces are continuous, that is there is no big jump or vibration for either the *graphic hand* or contact forces between two adjacent simulation time steps.

(3) *Computation time*. High-update-rate haptic rendering is required to simulate high contact stiffness in a stable manner [7]. The computation time should be lower than 1 ms for various hand-object interaction scenarios, such as single-finger, multi-finger, and whole-hand interactions.

A. Results of Different Manipulation Behaviors

The human hand is skilled in interacting with various objects in the surrounding world through diverse manipulation behaviors. In order to obtain realistic hand-object haptic interaction simulation, hand-based haptic simulation methods should support objects with different shapes for different manipulation behaviors. Fig. 11 shows several typical hand-based interaction scenarios using our method, including five objects with different shapes (i.e., sphere, cylinder, triangular prism, cube, and hexagonal prism) and three manipulation behaviors (i.e., single-finger touching, two-finger pinching, and five-finger grasping involving palm-object contacts). Five objects are constructed by level-3 sphere-tree models consisting of more than 500 spheres.

As shown in Fig. 11, the sphere and the cylinder have curved surfaces with different curvature, and the surface angles of the triangular prism, the cube and the hexagonal prism are different, which are acute, right and obtuse angles, respectively. We can see that our method can produce nonpenetration simulation results for interaction scenarios in Fig. 11, which includes contacts with different surfaces and different edges, as well as single-point and multiregion/finger contacts.

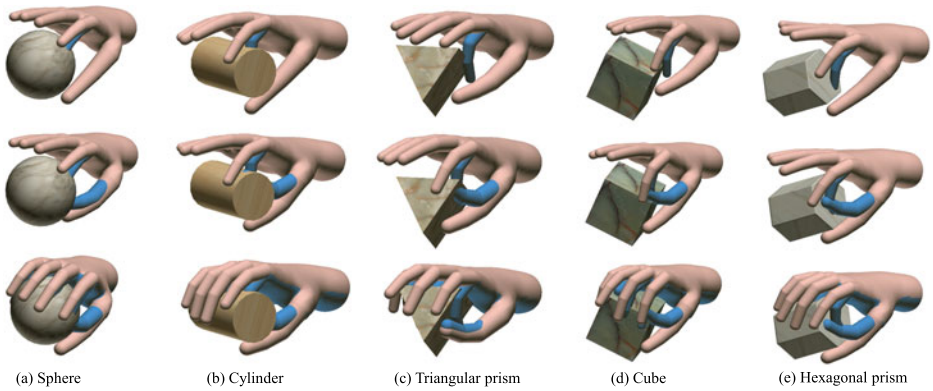


Fig. 11. Hand-based haptic interaction results of typical scenarios: including five objects, that is (a) sphere, (b) cylinder, (c) triangular prism, (d) cube, and (e) hexagonal prism, and three manipulation behaviors, that is single-finger touching (first row), two-finger pinching (second row), and five-finger grasping involving palm-object contacts (third row). Skin color represents the optimized *graphic hand*, and blue represents the *haptic hand*.

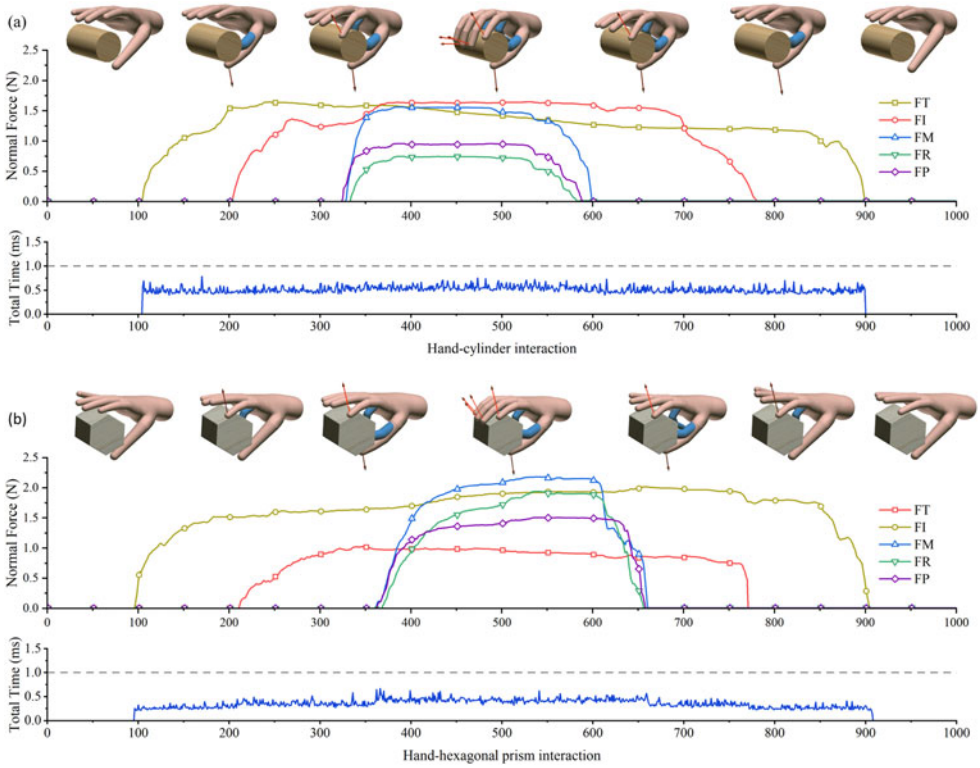


Fig. 12. The normal forces of five fingers and the total time cost of our method for hand-based haptic interaction. (a) Results of hand-cylinder haptic interaction. (b) Results of hand-hexagonal prism haptic interaction. The red lines with arrows represent the normal forces during the hand-object interaction. The arrows and the length of these lines show the direction and the magnitude of the calculated normal forces, respectively. The interaction scenarios at the top of the figure correspond to the data in the figure. Skin color represents the optimized *graphic hand*, and blue represents the *haptic hand*. FT-Force on thumb; FI-Force on index; FM-Force on middle; FR-Force on ring; FT-Force on pinky.

Fig. 12 shows the computed normal forces of five fingers and the total time cost of our method for two hand-based haptic interaction scenarios (i.e., hand-cylinder haptic interaction and hand-hexagonal prism haptic interaction). The interaction scenarios at the top of the figure include a series of manipulation behaviors, that is none-contact, single-finger, two-finger, five-finger, two-finger, single-finger, and none-contact hand-object haptic interaction. The red lines with arrows represent the normal forces during the interaction. The arrows and the length of these lines show the direction and the magnitude of the calculated normal forces, respectively.

In Fig. 12, the curve lines with different colors show the smoothness and stability of the calculated normal forces using our method, and the total time cost of the whole interaction process is less than 1 ms. We calculate the average of the total time cost for the five typical hand-object haptic interaction scenarios shown in Fig. 11, and the results are 0.41 ms, 0.31 ms, 0.31 ms, 0.32 ms, and 0.48 ms for hand-sphere, hand-cylinder, hand-triangular prism, hand-cube, and hand-hexagonal prism haptic interaction, respectively.

Sliding on the surface of an object is also a typical operation. We further test the performance of our method by

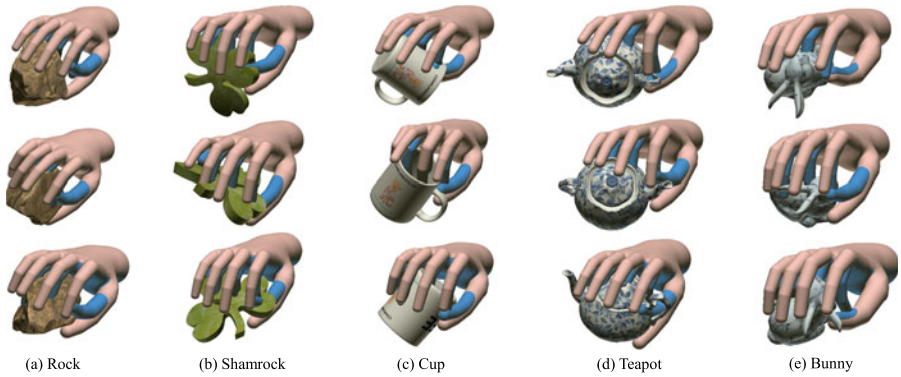


Fig. 13. Hand-based haptic interaction results of complex scenarios: five complex-shaped objects are manipulated with five fingers, including (a) rock, (b) shamrock, (b) cup, (c) teapot, and (d) bunny. Skin color represents the optimized *graphic hand*, and blue represents the *haptic hand*.

sliding on the surface of five objects shown in Fig. 11, and the results (please refer to the attached video) show that our method can maintain nonpenetration between the *graphic hand* and the virtual objects during the haptic interaction. Due to the limitation of the force-feedback device used in our experiments, the user cannot perceive the friction forces during the sliding process.

B. Results From Complex Interaction Scenarios

Different form objects with simple shapes like cylinders and cubes, objects with complex shapes have rich surface details and different topologies. The rock surface in Fig. 13(a) is uneven. The shamrock in Fig. 13(b) has concave surfaces and many corners. The inside of the cup in Fig. 13(c) is empty. The teapot in Fig. 3(d) has many special structures, such as thin handle, curved spout, and sunken cover. The bunny in Fig. 13(e) has different curve surfaces and its body shape is irregular, especially in its head with two ears. From the above observation, it involves different kinds of contact states when interacting with objects with complex shapes, which brings challenges to optimize a nonpenetration graphic configuration efficiently during the hand-object haptic interaction.

As mentioned in [25], compared with simple-shaped objects, finer sphere-tree models are required to simulate complex-shaped objects. We use level-4 sphere trees to model the objects shown in Fig. 13, that is 3089, 3511, 4047, 1792 and 2896 spheres are adopted for a rock, a shamrock, a cup, a teapot, and a bunny, respectively. Fig. 13 shows interactive simulation results of manipulating these objects using our method, and plausible configurations of the *graphic hand* are calculated for different manipulation behaviors during the hand-object interaction. The *graphic hand* can hold the rock from different directions with no penetration (Fig. 13(a)). The fingers maintain nonpenetration between two structures (i.e. a leave and the stem) of the shamrock (the second figure in Fig. 13(b)). Five fingers grab the cup with the thumb restricted by the handle, the middle finger inside the cup, and the index finger and ring finger on the rim of the cup (the second figure in Fig. 13(c)). In the second figure in Fig. 13(d), the thumb and the index finger pinch the curved spout and the middle finger and ring finger are restricted by the upper edge of the teapot. The index finger, middle finger and ring finger are perfectly blocked by the head and ear of the bunny (the fourth figure in Fig. 13(e)). We can see that our method can well handle multi-region contacts, and yield penetration-free interaction results.

Fig. 14 depicts the computed contact forces and the total time cost by using our method for two complex hand-based haptic interaction scenarios (i.e., the hand-cup haptic interaction and hand-bunny haptic interaction). The red lines with arrows represent the contact forces computed for the first phalanx of the index during the interaction. The arrows and the length of these lines show the direction and the magnitude of the calculated contact forces, respectively. In the force figure, the yellow curve denotes the magnitude of the total contact force, and the other three curves are the magnitude of its three components.

In Fig. 14, several contact states are examined during the haptic simulation. For the hand-cup haptic interaction (Fig. 14(a)), the contact forces increase smoothly as five fingers squeeze the cup (State 1). Five fingers hold the cup in State 2 to validate the contact stability in contact with the thin region. In State 3, fingers slide along the surface of the static cup to validate whether our method can simulate continuous hand-object contacts. In Fig. 14 (b), five fingers pinch the bunny and two fingers touch the thin ear in State 1. In State 2, five fingers grab the static bunny with the index finger restricted by two ears and maintain it for a while. In State 3, fingers slide along the body of the bunny, and frequent contact switches occur when the index finger slide from the ear to another. Overall, the smoothness and stability of the force curves demonstrate the robustness of our haptic simulation method. The total time cost of most interactive instances is less than 1 ms, and all is less than 1.5 ms. Therefore, our method can handle complex hand-object contacts stably and efficiently.

C. Computation Time Analysis

We further measure the computation time for the haptic simulations in regard to complex interaction scenarios shown in Fig. 13. The contact handling of five fingers is processed in parallel using a multi-threaded method, and all steps in one haptic simulation loop are performed sequentially. We counted the time cost for collision detection, collision response and the whole haptic simulation loop, as well as the number of contacts during the hand-object haptic simulation, which includes single-

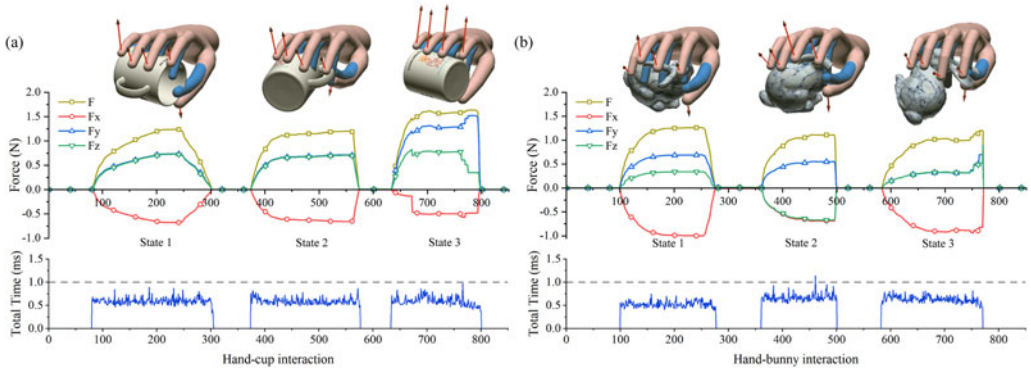


Fig. 14. The contact forces and the total time cost of our method for complex hand-based haptic interaction scenarios. (a) Results of hand-cup haptic interaction. (b) Results of hand-bunny haptic interaction. The red lines with arrows represent the contact forces calculated for the first phalanx of the index. The arrows and the length of these lines show the direction and the magnitude of the contact forces, respectively. The interaction scenarios at the top of the figure correspond to the data in the figure. Skin color represents the optimized *graphic hand*, and blue represents the *haptic hand*.

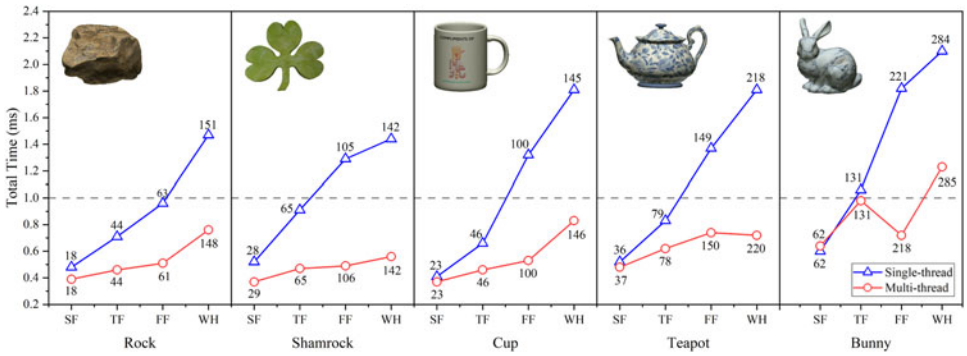


Fig. 15. Performance of our method in terms of computation time on different manipulation scenarios. SF: single finger. TF: two fingers. FF: five fingers. WH: whole hand. The number next to the marker on the line represents the number of contact points.

finger (SF), two-finger (TF), five-finger (FF) and whole-hand (WH) interactions. In addition, we also count the abovementioned metrics of our method implemented in a single thread. The results shown in Fig. 15 are averaged over 100 interactive instances.

As can be seen from Fig. 15, the number of contacts is relevant to that of fingers used in contact with the virtual object. For both multi-threaded and single-threaded implementations, the total time cost increases as the number of contact increases for all interaction scenarios. When the interaction involves a small number of contacts, such as the single-finger interaction, it seems that the multi-threaded method has no obvious advantages in terms of the time cost. As the number of contacts increases, the multi-threaded method increasingly surpasses the single-threaded method, especially for the whole-hand interaction, for which the time cost is reduced by almost 100% for the approximate number of contacts.

D. Discussion

Our method has been validated for the interaction between the virtual hand and rigid virtual objects, and the update rate of the haptic simulation loop can reach about 1 kHz. Whereas the time cost may be higher than 1 ms for some complex interaction scenarios, such as the hand-bunny haptic simulation with the whole hand even though a multi-threaded method is adapted (Fig. 15). We can also find that it took almost 1 ms for the two-finger interaction with the bunny (131 contacts) through multithreading. The phenomenon results from the fact that the index finger touches the bunny ear that needs numerous spheres to represent, which consumes significant computational time. For the interaction between the virtual hand and deformable or elastic/viscoelastic objects, it will be more time-consuming due to the possible increase in the number of contacts. One possible solution is to use GPUs to accelerate the computation process. Besides, the reasonable reduction of contact constraints can be another effective method to improve the efficiency, such as the aggregate constraint method [22].

It is still an open research question how to quantitatively define none-penetration and stability. In existing studies of virtual hand interaction and haptic simulation, the evaluation of nonpenetration and stability was assessed and reported qualitatively, such as the visual display of the interaction with virtual objects for the evaluation of nonpenetration [21], [24], and the display of force curve for stability [34], [35]. Similarly, in this paper Figs. 11 and 13 depict visual results of hand-object interaction using our method, and Figs. 12 and 14 demonstrate force curves. To quantitatively evaluate our method, we define the penetration depth between the graphic

finger and sphere models of virtual objects, that is the biggest penetration distance between the graphic finger and object spheres during one interaction cycle. We calculated the penetration depth during the interaction with different objects, and the result is about 0.1 mm. In addition, the stability is defined as the variation rate of contact force **F** between two adjacent simulation time steps. We analyzed the variation rate of **F** for the index in the case of the haptic hand holding the object with no movement, and the average variation rate of the magnitude of **F** and its three components are 0.18%(±0.32%), 0.18%(±0.32%), 0.16%(±0.29%), 0.18%(±0.32%), respectively. The participants didn't feel perceptible penetration and instability during the above experiments.

We build a model to compute the contact forces for the hand-object interaction. Theoretically, the contact forces at all contact points can be obtained. Due to the limitations of the force-feedback glove used in the experiment, we calculate the normal forces that should be exerted at fingertips based on our contact force model, and users can only experience the force feedback at their fingertips. In addition to the normal forces applied to fingertips, other forces can be also calculated based on the calculated contact forces, such as frictional forces during the sliding process. Considering the fidelity of the force feedback, a contact force model that corresponds to real manipulation behaviors is required, such as using data-driven methods [36], [37].

VI. Conclusion

We have proposed a decoupled-and-progressive optimization framework for virtual hand haptic simulation. This framework dealt with the challenge of high computational complexity of computing the *graphic-hand* configuration by decoupling and progressively computing the graphic palm and fingers. We have also introduced a hand-object contact simulation method using configuration-based optimization, and integrated this method into our decoupled-and-progressive framework. We show that our method can stably simulate complex hand-object contact situations without penetration at an update rate of 1 kHz, such as interaction with complex-shaped (concave surface and narrow space) objects, whole-hand interaction with multiple contact distributions, and sliding contacts.

The decoupling of the high-dimensional optimization reduces the computational complexity for the hand-object contact simulation, helps to improve the computational efficiency and accuracy. Benefiting from the uniformly nonpenetration hand-object contact constraints, our constraint-based method is appropriate for multiple contact scenarios, which is verified for the interactions with static virtual objects. In the future, we plan to further extend our method to simulate dynamic interactions (e.g., considering external forces and inertia of virtual objects), for instance complicated hand-based object manipulations in virtual assembly/maintenance tasks, even to simulate the manipulation of moving objects with complex physical properties, such as deformable and viscoelastic objects. User studies could be conducted to evaluate the usability of the method for complicated manipulation behaviors, such as bimanual haptic interaction tasks within virtual environments.

Acknowledgments

We would like to thank the anonymous reviewers, the participants of the study. In addition, thanks to Yuan Guo and Xuesong Bian at Beihang University for video recording.

References

[1] S. Mulatto, A. Formaglio, M. Malvezzi, and D. Prattichizzo, "Using postural synergies to animate a low-dimensional hand avatar in haptic simulation," *IEEE Trans. Haptics*, vol. 6, no. 1, pp. 106–116, Jan.–Mar. 2013.

[2] C. S. Tzafestas, "Whole-hand kinesthetic feedback and haptic perception in dextrous virtual manipulation," *IEEE Trans. Syst., Man, Cybern.-A: Syst. Humans*, vol. 33, no. 1, pp. 100–113, Jan. 2003.

[3] M. C. Lin and M. Otaduy, *Haptic Rendering: Foundations, Algorithms, and Applications*. Boca Raton, FL, USA: CRC Press, 2008.

[4] M. A. Otaduy, C. Garre, and M. C. Lin, "Representations and algorithms for force-feedback display," *Proc. IEEE*, vol. 101, no. 9, pp. 2068–2080, Sep. 2013.

[5] P. Xia, "New advances for haptic rendering: State of the art," *Vis. Comput.*, vol. 34, no. 2, pp. 271–287, 2018.

[6] I. M. Bullock and A. M. Dollar, "Classifying human manipulation behavior," in *Proc. IEEE Int. Conf. Rehabil. Robot.*, 2011, pp. 1–6.

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

[8] E. Burns, S. Razzaque, A. T. Panter, M. C. Whitton, M. R. McCallus, and F. P. Brooks Jr, "The hand is more easily fooled than the eye: Users are more sensitive to visual interpenetration than to visual-proprioceptive discrepancy," *Presence: Teleoperators Virtual Environments*, vol. 15, no. 1, pp. 1–15, 2006.

[9] G. Zachmann and A. Rettig, "Natural and robust interaction in virtual assembly simulation," in *Proc. 8th ISPE Int. Conf. Concurrent Engineering: Res. Appl.*, 2001, pp. 425–434.

[10] M. Moehring and B. Froehlich, "Pseudo-physical interaction with a virtual car interior in immersive environments," in *Proc. 11th Eurographics Conf. Virtual Environ.*, 2005, pp. 181–189.

[11] M. Moehring and B. Froehlich, "Enabling functional validation of virtual cars through natural interaction metaphors," in *Proc. IEEE Virtual Reality Conf.*, 2010, pp. 27–34.

[12] C. W. Borst and A. P. Indugula, "A spring model for whole-hand virtual grasping," *Presence*, vol. 15, no. 1, pp. 47–61, Feb. 2006.

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

[14] C. Garre and M. A. Otaduy, "Toward haptic rendering of full-hand touch," in *Proc. Congreso Español de Informatica*, 2009, pp. 19–26.

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

[16] M. Verschoor, D. Lobo, and M. A. Otaduy, "Soft hand simulation for smooth and robust natural interaction," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2018, pp. 183–190.

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

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

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

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

[21] J. Jacobs, M. Stengel, and B. Froehlich, "A generalized god-object method for plausible finger-based interactions in virtual environments," in *Proc. IEEE Symp. 3D User Interfaces*, 2012, pp. 43–51.

[22] A. Talvas, M. Marchal, C. Duriez, and M. A. Otaduy, "Aggregate constraints for virtual manipulation with soft fingers," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 4, pp. 452–461, Apr. 2015.

[23] J. Allard, F. Faure, H. Courtecuisse, F. Falipou, C. Duriez, and P. G. Kry, "Volume contact constraints at arbitrary resolution," in *Proc. ACM SIGGRAPH Papers*, 2010, Art. no. 82.

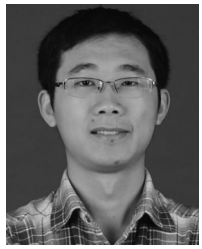
[24] D. Wang, X. Zhao, Y. Shi, Y. Zhang, and J. Xiao, "Six degree-of-freedom haptic simulation of a stringed musical instrument for triggering sounds," *IEEE Trans. Haptics*, vol. 10, no. 2, pp. 265–275, Apr.–Jun. 2017.

[25] D. Wang, X. Zhang, Y. Zhang, and J. Xiao, "Configuration-based optimization for six degree-of-freedom haptic rendering for fine manipulation," *IEEE Trans. Haptics*, vol. 6, no. 02, pp. 167–180, Apr.–Jun. 2013.
[26] D. Wang, Y. Shi, S. Liu, Y. Zhang, and J. Xiao, "Haptic simulation of organ deformation and hybrid contacts in dental operations," *IEEE Trans. Haptics*, vol. 7, no. 1, pp. 48–60, Jan.–Mar. 2014.
[27] J. Napier, J. R. Napier, and R. H. Tuttle, *Hands*, vol. 9. Princeton, NJ, USA: Princeton Univ. Press, 1993.
[28] N. Wheatland, Y. Wang, H. Song, M. Neff, V. Zordan, and S. Jörg, "State of the art in hand and finger modeling and animation," *Comput. Graph. Forum*, vol. 34, no. 2, pp. 735–760, 2015.
[29] G. Bradshaw and C . O' Sullivan, "Adaptive medial-axis approximation for sphere-tree construction," *ACM Trans. Graph.*, vol. 23, no. 1, pp. 1–26, 2004.
[30] G. Yu, D. Wang, Y. Zhang, and J. Xiao, "Simulating sharp geometric features in six degrees-of-freedom haptic rendering," *IEEE Trans. Haptics*, vol. 8, no. 1, pp. 67–78, Jan–Mar. 2015.
[31] D. L. James and D. K. Pai, "BD-Tree: Output-sensitive collision detection for reduced deformable models," in *Proc. ACM SIGGRAPH Papers*, 2004, pp. 393–398.
[32] J. Nocedal and S. Wright, *Numerical Optimization*. Berlin, Germany: Springer Science & Business Media, 2006.
[33] X. Gu, Y. Zhang, W. Sun, Y. Bian, D. Zhou, and P. O. Kristensson, "Dexmo: An inexpensive and lightweight mechanical exoskeleton for motion capture and force feedback in VR," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2016, pp. 1991–1995.
[34] F. Ryden and H. J. Chizeck, "A proxy method for real-time 3-DOF haptic rendering of streaming point cloud data," *IEEE Trans. Haptics*, vol. 6, no. 3, pp. 257–267, Jul–Sep. 2013.
[35] H. Ding, H. Mitake, and S. Hasegawa, "Continuous collision detection for virtual proxy haptic rendering of deformable triangular mesh models," *IEEE Trans. Haptics*, vol. 12, no. 4, pp. 624–634, Oct.–Dec. 2019.
[36] R. H. Osgouei, J. R. Kim, and S. Choi, "Data-driven texture modeling and rendering on electrovibration display," *IEEE Trans. Haptics*, vol. 13, no. 2, pp. 298–311, Apr.–Jun. 2020.
[37] A. Abdulali, I. R. Atadjanov, and S. Jeon, "Visually guided acquisition of contact dynamics and case study in data-driven haptic texture modeling," *IEEE Trans. Haptics*, vol. 13, no. 3, pp. 611–627, Jul–Sep. 2020.

Qianqian Tong (Member, IEEE) received the Ph.D. degree from the School of Computer Science, Wuhan University, Wuhan, China in 2019. She is currently a Postdoctoral Fellow with Peng Cheng Laboratory, Shenzhen, China. Her research interests include haptic rendering, haptic human-machine interface, and medical imaging analysis.

Qiong Wang received the Ph.D. degree from the Chinese University of Hong Kong, China, in 2012. She is currently a Researcher with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. Her research interests include VR applications in medicine, visualization, medical imaging, human-computer interaction, and computer graphics.

Yingkui Zhang received the M.S. degree from Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences in 2021. He is currently a Research Assistant with Hong Kong Polytechnic University, Hong Kong. His research interests include computer vision and AR/VR.



Xiangyun Liao is currently an associate researcher with the Guangdong Provincial Key Laboratory of Machine Vision and Virtual Reality Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Beijing, China. His research interests include virtual reality, physically-based simulation, and medical imaging.

Wenxuan Wei is currently working toward the M.S. degree with the School of Mechanical Engineering and Automation, Beihang University, Beijing, China. Her research interests include human-machine interaction, haptic device, and robotics.

Yuru Zhang (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from Beihang University, Beijing, China in 1987. She is currently a Professor with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. Her research interests include haptic human-machine interface, medical robotic systems, robotic dexterous manipulation, and virtual prototyping. She is a Member of ASME.

Jing Xiao (Fellow, IEEE) received the Ph.D. degree in computer, information, and control engineering from the University of Michigan, Michigan, MI, USA. She is currently the Deans' Excellence Professor, William B. Smith Distinguished Fellow in Robotics Engineering, Professor of Computer Science, and the Director of the Robotics Engineering Program, Worcester Polytechnic Institute (WPI), Worcester, MA, USA. She joined WPI from the University of North Carolina, Charlotte, North Carolina, NC, USA. She has co-authored a monograph, holds one patent, and published extensively in major robotics journals, conferences, and books. Her research interests include robotics, haptics, and intelligent systems. She was the recipient of the 2015 Faculty Outstanding Research Award of the College of Computing and Informatics, University of North Carolina at Charlotte.

Dangxiao Wang (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from Beihang University, Beijing, China in 2004. He is currently a Professor with the State Key Laboratory of Virtual Reality Technology and Systems in Beihang University. From 2006 to 2016, he was an Assistant and Associate Professor with the School of Mechanical Engineering and Automation, Beihang University. His research interests include haptic rendering, NeuroHaptics and medical robotic systems. He was an Associate Editor for IEEE TRANSACTIONS ON HAPTICS from 2015 to 2018. He was the Chair of Executive Committee of the IEEE Technical Committee on Haptics from 2014 to 2017.