Survey on Hand-Based Haptic Interaction for Virtual Reality

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

(Tutorial paper)

***Abstract*—Interacting with virtual objects with haptic feedback directly using the user's hand (hand-based haptic interaction) has attracted increasing attention. Due to the high degrees of freedom of the hand, compared with tool-based interactive simulation using a pen-like haptic proxy, hand-based haptic simulation faces greater challenges, mainly including higher motion mapping and modeling difficulty of deformable hand avatars, higher computational complexity of contact dynamics, and nontrivial multi-modal fusion feedback. In this article, we aim to review key computing components for hand-based haptic simulation, and draw out major findings in this direction while analyzing the gaps toward immersive and natural hand-based haptic interaction. To this end, we investigate existing relevant studies on hand-based interaction with kinesthetic and/or cutaneous display in terms of virtual hand modeling, hand-based haptic rendering, and visuo-haptic fusion feedback. By identifying current challenges, we finally highlight future perspectives in this field.**

***Index Terms*—Hand-based haptic interaction, virtual hand, hand-based haptic rendering, virtual reality.**

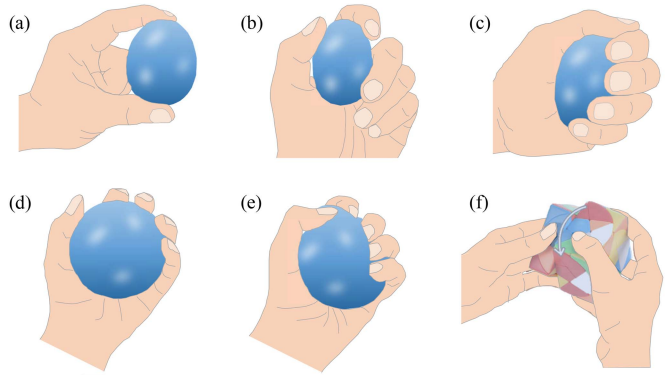

Fig. 1. Hand-based interaction examples with diverse hand-object contact states, bringing challenges of high computational complexity and time-consumption to hand-based haptic simulation. (a) Pinch an ellipsoid with two fingers. (b) Hold an ellipsoid with three fingers. (c) Grasp an ellipsoid with the whole hand. Grasp a deformable ball loosely (d) and tightly (e). (f) Manipulate a hinge-constrained object (such as a Rubik's cube) with two hands.

I. Introduction

DIRECTLY using the user's hand to manipulate virtual objects provides an intuitive and direct interactive mode for virtual reality (VR) technology. With the aid of technological advancements in VR hardware devices, such as head-mounted displays and hand tracking devices, hand-based interaction has become a compelling subject in the VR field. Haptics [1], [2] play a significant role in enhancing the user experience while interacting with virtual environments. Over the past decades, the emergence of haptic systems for fingertip and/or hand [3], [4], [5] has promoted the immersive sensation of hand-based interaction. Further, the simulation of hand-based haptic interaction (hand-based haptic simulation) that enables a user to explore virtual environments through these haptic systems has gained increasing attention.

Hand-based haptic simulation refers to the modeling of virtual hands, virtual objects, and hand-object contacts, as well as the computation of haptic information that is required to provide users with an exact haptic experience. Different from 3-DoF/6-DoF pen-like tools used in tool-based haptic interaction, the human hand has a complex kinematic structure with more than 20 DoFs [6]. As shown in Fig. 1, the high-DoF characteristic of the human hand makes hand-based interaction involve diverse manipulation behaviors and multiple contact points/regions with various distributions, which makes the computational issues of hand-based haptic simulation more complicated and time-consuming than that of 3-DoF/6-DoF tool-based haptic simulation. Specifically, it is challenging to compute a plausible hand configuration to ensure the non-penetration simulation,

Manuscript received 26 June 2022; revised 17 December 2022, 3 March 2023, and 30 March 2023; accepted 7 April 2023. Date of publication 11 April 2023; date of current version 20 June 2023. This work was supported in part 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 paper was recommended for publication by Associate Editor M. Bianchi and Editor-in-Chief D. Prattichizzo upon evaluation of the reviewers' comments. *(Corresponding author: Dangxiao Wang.)*

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

Wenxuan Wei and Yuru Zhang are with the State Key Laboratory of Virtual Reality Technology and Systems, 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 Robotics Engineering Department, Worcester Polytechnic Institute, Worcester, MA 01609 USA (e-mail: jxiao2@wpi.edu).

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

Digital Object Identifier 10.1109/TOH.2023.3266199

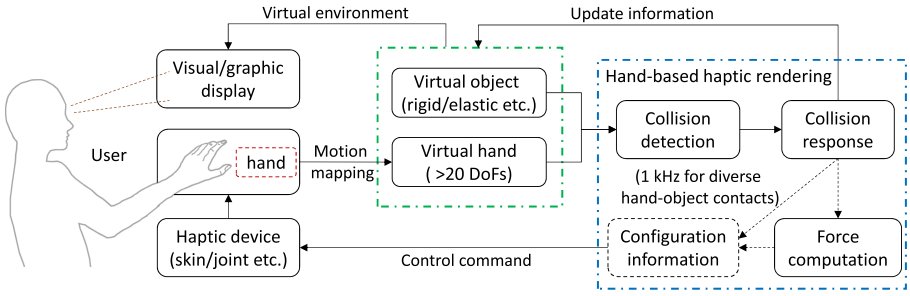

Fig. 2. General computational framework for the simulation of hand-based haptic interaction. This survey focuses on virtual hand modeling, hand-based haptic rendering, and visuo-haptic fusion feedback.

and it is difficult to ensure both accuracy and efficiency in hand-based haptic simulation due to the high computational complexity.

This survey aims to review existing approaches for hand-based haptic simulation from the perspectives of its computing components, including virtual hand modeling, hand-based haptic rendering (kinesthetic and tactile rendering), and visuo-haptic fusion feedback. To focus on the theme of hand-based haptic simulation, we excluded hand/finger modeling methods for computer animation that deal only with visual realism (e.g., skin color), and subtle haptic experience caused by skin characteristics, such as skin wrinkles, fine hairs on the skin, and palm creases, as well as hand motion tracking methods. Readers may refer to previous surveys on relevant research [7]. Further, this paper focuses on hand-based haptic simulation algorithms and software rather than relevant hardware technology that can be referred to in existing reviews [3], [4], [5].

II. Framework of Hand-Based Haptic Simulation

The simulation of hand-based haptic interaction can be conceptualized in a general framework. As shown in Fig. 2, the user's hand is first tracked in real time by a hand motion capture system. With the mapped hand motion information, and the predefined geometric and/or physical models of the virtual hand and virtual objects, collision detection, collision response, and force computation are sequentially performed for visuo-haptic display. The updated information for visual/graphic display mainly includes the non-penetration configuration of the virtual hand, the dynamic response (e.g., motion and deformation) of the virtual hand/objects, etc. Meanwhile, depending on the control principle of the haptic device (e.g. impedance or admittance control), the force (and/or the relevant configuration information) that needs to be exerted on the user's hand is computed for haptic display. Finally, the user receives the desired haptic stimuli through a haptic device (e.g., a force-feedback glove), and simultaneously observes the corresponding graphic display.

From the aforementioned interaction process, there are two major computational issues for hand-based haptic simulation: (1) compute the non-penetration configuration of the virtual hand, and (2) compute the forces resulting from contact between the virtual hand and the VE and/or relevant configuration information for haptic display. The high-DoF kinematic structure of the human hand produces diverse manipulation behaviors of virtual objects and various hand-object contact distributions (Fig. 1), which attaches high requirements to hand-object contact handling, such as the computational efficiency and generalization ability for different contact distributions.

Virtual hand models have an impact on realistic visual display and efficient haptic simulation. Virtual hand modeling not only involves the geometric representation of the human hand, but also needs to characterize its physical attributes and kinematic properties. During hand-based interaction, accurate high-DoF hand motion mapping from the real world to the virtual environment is crucial to acquire accurate hand kinematics (pose and motion) for contact identification and force computation. With a realistic virtual hand, efficient hand-based haptic rendering facilitates stable and natural haptic display in various contact situations without penetration between the virtual hand and manipulated objects. Recent years have witnessed the advent of different types of haptic devices, and different hand-based haptic simulation methods have also been implemented to provide users' hands with force feedback or cutaneous feedback. Generally, natural hand-based haptic interaction follows three aspects:

1) Realistic Visual Representation: including realistic virtual hand models and diverse manipulated objects.

A realistic virtual hand involves geometric model, physical model, and kinematic model. Important metrics include the complexity (e.g., the number of geometric elements, such as triangles and spheres) and accuracy (e.g., the error of hand size in terms of width, length, circumference, etc.) of the geometric model, the fidelity of physical parameters (e.g., elastic modulus, Poisson's ratio, nonlinear stress-strain characteristics) for simulating different hand structures (bone, muscle, skin, etc.), and the number of DoFs obtained from hand tracking and mapping methods to describe the kinematic model of the palm and fingers.

Diverse manipulated objects refer to rigid bodies, deformable bodies, fluids [8], [9], etc. Important metrics concern the geometric complexity (e.g., the number of geometric elements, such as triangles and spheres), the supported fine-scale geometric details (e.g., edges, pits, and cracks), the stiffness range, as well as the heterogeneous and nonlinear stress-strain characteristics of the manipulated virtual objects.

2) High-Fidelity Haptic Feedback: Supporting diverse hand manipulation behaviors, high-fidelity contact handling, and diverse interaction forces.

Diverse hand manipulation behaviors come to single-point and multi-point interaction. Typical examples include single-point tapping, pressing, sliding, and multi-finger grasping/holding, squeezing and rotating. Important metrics include the supported manipulation types (e.g., holding an object still, squeezing a sponge, grasping an object) [10], as well as the motion range and motion resolution of the manipulation.

High-fidelity contact handling refers to penetration-free modeling for various hand-object contacts, and stable simulation of diverse manipulation behaviors. The various distribution of hand-object contacts is capable of invoking dexterous manipulations, e.g., rotating an object. Important metrics include the hand-object penetration depth, and the stability of the simulated manipulation behaviors.

Diverse interaction forces include fingertip normal force, tangential/friction force between finger/palm and the manipulated objects, gravity and inertia force of the virtual object. Important metrics involve the dimension, accuracy, stability and dynamic/transient characteristics of the desired haptic stimuli on the user's hand.

A high update rate (e.g., 1 kHz) is required for kinesthetic rendering to simulate high contact stiffness stably [11], [12], while the update rate can be appropriately reduced for the interaction with soft objects. Cutaneous rendering does not impose the update rate restriction for stability [13], [14].

3) Concurrent Visuo-Haptic Display: calling for spatial and temporal registration.

Haptics and vision are both important for humans to explore the virtual environment with their hands [2]. Effective integration of visual and haptic display is vital for natural hand-based interaction [1]. The main metric is the spatial and temporal consistency between the visual display and haptic display.

This survey summarizes key computational components currently available in hand-based haptic simulation depicted in Fig. 2, i.e., virtual hand modeling, hand-based haptic rendering, and visuo-haptic fusion feedback.

III. Virtual Hand Modeling

Fig. 3 shows the bone structure of the human hand, and the real hand consists of ligaments, tendons, and muscles. Each finger except for the thumb has three bones (distal, middle, and proximal phalanges), and three joints (MCP, PIP, and DIP joints). The thumb has two bones (distal and proximal phalanges) and two joints (MCP and DIP joints). For more details about hands, readers can refer to [15].

Virtual hand modeling is a prerequisite for hand-based haptic interaction, which represents external geometric attributes of the human hand, as well as characterizes its internal physical attributes. Here, instead of discussing how to construct a visually realistic hand avatar through graphics rendering, we focus on modeling methods that facilitate efficient hand-object contact handling and diverse manipulation behaviors, which is fundamental for high-fidelity haptic simulation.

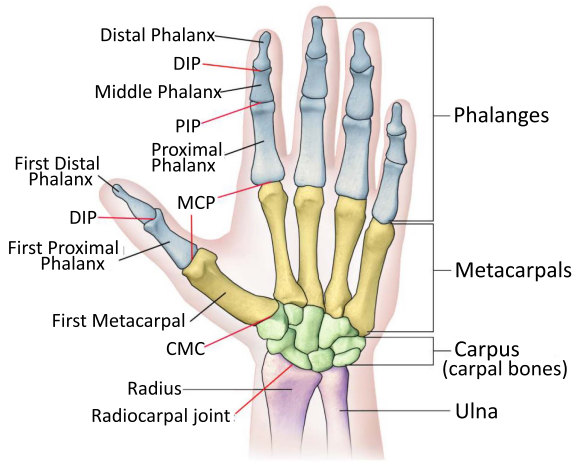

Fig. 3. The bone structure of the human hand including the forearm and wrist [7].[1] CMC: Carpometacarpal joint; MCP: Metacarpophalangeal joint; PIP: Proximal interphalangeal joint; DIP: Distal interphalangeal joint.

A. Geometric Model

Geometric models are used to represent the 3D shape of the virtual hand, and geometric constraints of the palm and multiple fingers. Each element of the virtual hand can be represented by using mainstream discrete models, e.g., mesh model [16], [17], [18], [19], [20], [21], [22], [23], voxel model [24], [25], point-shell model [26], [27], and sphere model [28] [29]. The mesh model (such as Fig. 4(a)), adopted in the field of haptics by Zilles et al. [30] from computer graphics, can provide a compelling display of the virtual hand because of its fine surface mesh, but this model tends to produce penetration effect during force rendering [31]. When combined with voxel model, the point-shell model [27], [32] can facilitate efficient hand-object collision detection.

Simplified models (e.g., skeleton model and sphere model) are usually used in geometric modeling of a virtual hand. Wan et al. [27] constructed a virtual hand with a metaball model, which was tessellated into a triangular mesh [33] to facilitate both the motion control and the visual enhancement of the virtual hand. Some researchers used a set of skeletons attached with meshes to represent the virtual hand, and the attached meshes were deformed by changing the skeleton poses, such as Zhao et al. [34] and Kim et al. [35], [36], [37].

B. Physical Model

Physical models are used to describe the physical properties of different hand structures (bone, muscle, skin, etc.), including elastic modulus, Poisson's ratio, nonlinear stress-strain characteristics, etc. The human hand is anisotropic and non-linear, which is connected by multiple bones and joints covered with soft tissue, bringing great challenges to construct a physically realistic virtual hand.

[1][Online]. Available: https://commons.wikimedia.org/wiki/File:Blausen_0440_HandBones.png

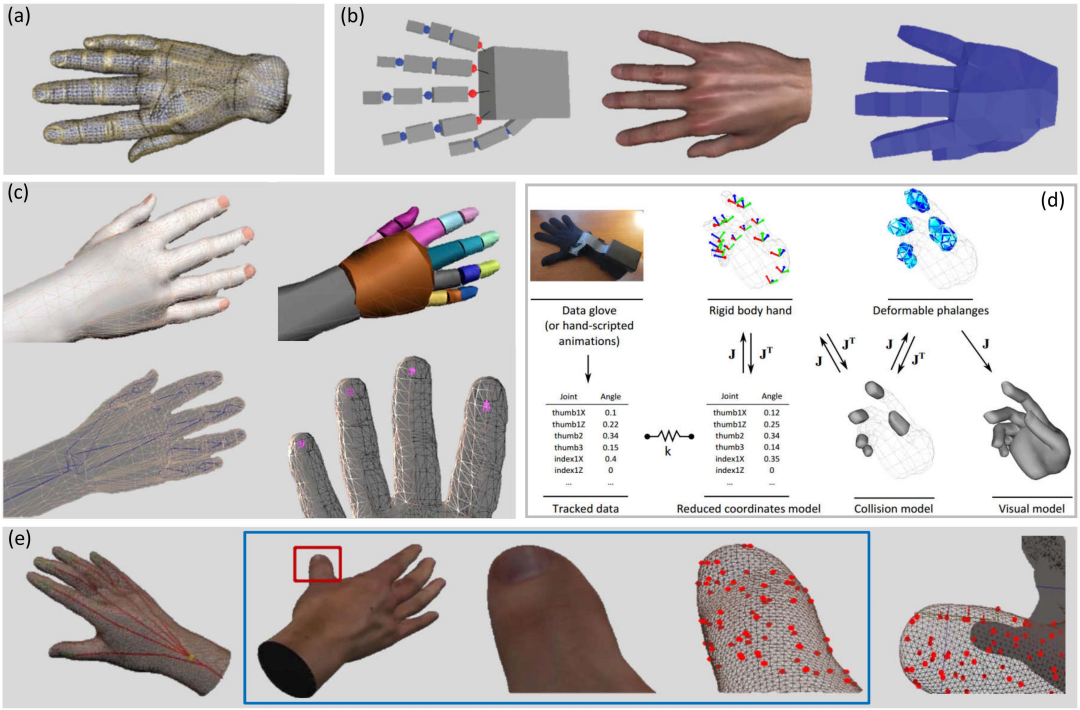

Fig. 4. Examples of virtual hand models. (a) A meshed virtual hand model [19]. (b) A virtual hand including the skeletal bone and a deformable flesh [41] (left: articulated hand skeleton with 16 bones, middle: surface model for visual rendering, and right: embedding tetrahedral mesh to model flesh elasticity). (c) A 4-layer flexible virtual hand model [24]. (d) A 5-layer hand model [42]. (e) Deformable hand models and physical particles [36] (left: global deformation model, i.e., shape deformation by the hand skeleton, middle (blue box): sparse physics particles, and right: local deformation model for contacted skin).

Articulating multiple rigid hinged links is the most straightforward method for constructing a virtual hand avatar. Borst et al. [16] built an articulated virtual hand model that was driven by a spring model. This method was shown to be efficient for dexterous interaction with haptic feedback on the palm and fingers [38]. Ott et al. [39] extended such method for bimanual haptic manipulation. Jacobs et al. [40] developed a Jacobian matrix to represent their God-hand model consisting of a palm and separately connected phalanxes for each finger.

The deformable simulation of the virtual hand is not only necessary for a more realistic-looking simulation [43], but also for more accurate simulation of contact friction required for stable manipulation [42]. Finite element model (FEM) and its extensions are usually used to simulate deformable hands [17].

To maintain intuitive hand-object interaction in real time, Kim et al. [36] simulated deformable hands using a number of physics particles. As shown in Fig. 4(e), the global hand deformation is driven by hand skeleton poses. A small number of physics particles are attached to skin meshes for physical simulation. Local deformation occurs when the skin meshes collide with a virtual object, providing stable manipulation results. This deformable hand was utilized for the handover manipulation [37] of a virtual object as well as the direct hand manipulation of constrained virtual objects [35].

C. Kinematic Model

Kinematic models are generally driven by captured data using a hand tracking system to control virtual hand movements. The bones and the joints that link those bones together are two key components that constitute the basic structure of a hand. For virtual hand modeling, it is challenging to simulate joints between bone components so that its adjacent components do not separate or penetrate each other. Jacobs et al. [40] used two-DoF spherical joints to limit the fingers' movement in relation to the palm, and used revolute joints that allow for one rotational DoF to connect the phalanxes.

The number of degrees of freedom that determines the complexity of the virtual hand model is an important variable that affects the kinematic dexterity of the virtual hand. Some researchers modeled the virtual hand using 33 DoFs for maintaining the anatomical articulation [34], [44]. Such a high-resolution anatomical model is computationally expensive for haptic rendering. Studies in neuroscience demonstrated that a substantial reduction from the high number of DoFs is possible to describe hand posture [45]. Liu et al. [46] used 22 DoFs to model their virtual hand: 2 DoFs MCP joints, 1 DoF for proximal PIP joints, 1 DoF for DIP joints, and 3 DoFs for the palm. Mulatto et al. [6] animated a virtual hand avatar with 20 DoFs based on postural synergies, and the hand posture was represented using fewer variables.

The skeleton structure of the virtual hand determines the hand kinematic behaviors. Vision-based [47], [48] and glove-based (optical fibers, inertial sensors and strain gauges) methods, referring to previous surveys [3], [7], [49] for more details, are commonly used for hand motion tracking. There are two main ways to map the motion of the user's hand to the kinematic model: fingertip mapping and joint space mapping.

Fingertip mapping is to establish the mapping relationship between the real-fingertip position and the virtual-fingertip position. Inverse kinematics (IK) [50] is used to solve joint angles for the knuckles of each finger. Fingertips and the palm are more frequently used for manipulating an object by a hand [17], [51]. Some studies developed kinematic inversion algorithms to estimate the kinematic configuration of the whole hand for limited tracking information (e.g., the motion of the index finger and thumb tips [6]).

Joint space mapping builds the mapping relationship between real hand joint angles with the virtual ones. With the tracked hand posture, the corresponding gesture can be determined according to the relationship between the angle of finger joint and the set threshold [52]. The joint angles of the virtual hand can also be computed according to the motion constraint relationship among knuckles for each finger with obtained bending angles [53] or by introducing calibration parameters (e.g., joint gains and offsets) [54], [55].

When applying the motion of the tracked hand directly to the virtual hand, some problems may occur. For example, sudden displacements of the hand may turn into excessive forces if the tracking output is noisy or discontinuous. Verschoor et al. [56] produced the configuration of intermediate bones by taking as input the tracked hand configuration and the full skeletal simulation, and coupling energy was modeled to ensure that the intermediate bones follow the tracked configuration. Benefiting from the ability to limit the tracking forces acting on the hand's tissue, this elastic tracking method ensures robust and natural interactive simulation.

D. Multi-Layer Hand Model

The human hand has a complex anatomical structure consisting of skin, soft tissues, bones, etc. The simulation of its deformable parts is vital for realistic display, and also necessary for stable hand manipulations, especially for interaction tasks with cutaneous feedback. Researchers have explored multi-layer modeling strategies for deformable hand simulation and hand-based haptic interaction.

Jacobs et al. [57] developed a virtual hand model by coupling soft bodies (15 soft pads) to a rigid hand skeleton (16 rigid finger bones), and they extended the lattice-shape matching algorithm [58] with adaptive stiffness of the soft bodies for robust interaction. Similarly, Pouliquen et al. [59] modeled fingers consisting of a rigid articulated skeleton and deformable finger pads, and FEM was used to model deformation tissues.

Garre et al. [41] built a rigid-soft coupled hand that includes skeletal bone and a deformable flesh based on layered deformable models [60], [61]. As shown in Fig. 4(b), the skeleton was formulated using an articulated model and the deformable flesh (embedded tetrahedral mesh) was modeled following a linear co-rotational FEM [59]. Linear blend skinning (LBS) [62] was used for the skinning implementation. The skeleton and flesh were bidirectionally coupled using zero-length springs between nodes of the deformable hand and rest positions skinned from the rigid body configuration.

Multi-layer strategies of virtual hand modeling have also been explored to achieve a compromise between accuracy and efficiency for hand-based haptic interaction. Wan et al. [24] proposed a 4-layer flexible virtual hand model (Fig. 4(c)) including a skin layer, a kinematics layer, a collision detection layer and a haptic layer. The deformation of the skin layer was driven by the kinematics layer using the skeletal subspace deformation (SSD) method [62]. The collision layer built with simplified geometries was used to implement efficient collision detection. The haptic layer was composed of many line segments for force computation. Talvas et al. [42] built a 5-layer virtual hand model (Fig. 4(d)): the tracked hand data, a reduced coordinates model, a mapped rigid body skeleton, deformable phalanges, and a surface collision/visual model. The rigid skeleton was mapped to the reduced coordinates model following the tracked data with stiff springs. The visual model was skinned from the deformable phalanges simulated with linear co-rotational FEM [63]. The collision model was mapped to both the rigid-body model and deformable phalanges using a global co-rotational approach [64].

IV. Hand-Based Haptic Rendering

Haptic rendering refers to the process of conveying haptic information about virtual environments to the user through desired sensory stimuli [65]. With the advent of multiple kinesthetic haptic devices and cutaneous haptic devices [4], [5], corresponding hand-based haptic rendering methods are required to provide users with kinesthetic feedback on muscles, tendons and joints (kinesthetic rendering), as well as cutaneous feedback on the skin (tactile rendering). This section discusses the key computational components of these two haptic rendering technologies.

A. Kinesthetic Rendering

For hand-based interaction, kinesthetic rendering can be formulated as the simulation of hand-object contacts (Fig. 5 and Table I), and feedback is usually displayed by computing coupling forces between the hand avatar and the kinesthetic haptic device. Similar to tool-based kinesthetic feedback [11], high frequency is required for hand-based force display. As described earlier, it is not trivial to realize a high enough update rate in hand-based haptic simulation due to its high computational complexity. In the following, we describe what efforts existing research has made in collision detection, collision response and force computation to achieve stable and efficient kinesthetic rendering.

1) Collision Detection: Collision detection is used to identify whether and how a virtual tool is in contact with virtual objects in a virtual environment at a certain moment. To provide precise contact information to collision response and force computation, efficient strategies of collision detection have been concerned for hand-based interaction.[2]

Voxel-based collision detection, implemented as the Voxmap PointShellTM (VPS) software [67], facilitates a high haptic

[2][Online]. Available: http://www.cyberglovesystems.com/

TABLE I
TYPICAL APPROACHES FOR HAND-BASED INTERACTION WITH KINESTHETIC FEEDBACK

Method	Features			Performance
	Hand model	Contact handling	Force model	
Borst et al.2006 [54]	(1) Articulated hand (rigid fingers and deformable palm) (2) 22 DoFs (fingers) with 2270 triangles	Novodex-based implementation	Spring-damper force/torque model	(1) Scenarios: grasping/manipulating various objects (teacup, bunny, rock, etc.) (2) Update rate: a few hundred Hertz for triangle meshes with several thousand vertices (3) Stability: providing good grasp stability with high friction constants (4) Fidelity: the hand model visually remains outside the object
Wan et al. 2009 [24]	(1) 4-layer model: skin, kinematics, collision detection, and haptic layer (2) 23 DoFs (fingers and palm) with more than 6500 triangles	Software toolkit: RAPID	Hooke's law based on line segments and voxelized model	(1) Scenarios: grasping a sphere and a cube (2) Update rate: over 60 fps (overall update)
Ott et al. 2010 [39]	(1) Articulated hand (rigid) (2) Mass-spring hand	Axis-aligned bounding boxes for collision detection Forward and Inverse Dynamics for non-penetration	Spring-damper force/torque model	(1) Scenarios: grasping a sphere and a cube (2) Update rate: near 1 kHz (force refresh) for single-handed interaction, around 900 Hz (haptic thread) for two-handed interaction
Garre et al. 2011 [41]	(1) Deformable hand coupling skeleton and flesh with skinning (2) 16 rigid bodies and 15 joints	Constraint-based response	Bidirectional viscoelastic coupling	(1) Scenarios: grasping rigid objects (car toy) and deformable objects (bread) (2) Update rate: 18.87 Hz (full simulation for the bread scene), 10.5 kHz (haptic loop) (3) Stability: guaranteeing stability by admissible viscoelasticity parameters (4) Fidelity: producing visually realistic hand avatar by LBS
Mulatto et al. 2012 [6]	(1) Deformable skin (2) 20-DoF kinematic skeleton	Ray-triangle test algorithm for collision detection NVIDIA PhysX library for dynamic simulation	God-object algorithm Friction-cone	(1) Scenarios: pinch grasping (2) Update rate: 30 Hz (graphics and physics simulation), about 1 kHz (Thumb and Index) (3) Stability: stable and free from vibrating effects (4) Fidelity: visually smooth and coherent for the avatar animation
Perez et al. 2013 [109]	(1) Deformable finger (2) Nonlinear elasticity with strain-limiting constraints	Constraint-based optimization problem solved by projected Gauss-Seidel relaxation	Virtual coupling	(1) Scenarios: sliding under frictional contact with a table (2) Update rate: 100 Hz for normal interactions, 40 Hz for highly constrained situations (3) Fidelity: enabling rich and robust deformations of the fingertip
Lobo et al. 2017 [66]	Soft finger model in [109]	Subspace-proxy-based method	Computing forces directly based on the deviation in the device actuated state	(1) Scenarios: enclosing a virtual object (cylinder) using a soft finger (2) Fidelity: offering superior passivity and transparency properties
An et al. 2020 [90]	Rigid hand with 24 joints	Proxy-based method for contact points, and determining projective direction for non-contact points	Pendulum-based force model	(1) Scenarios: grabbing different shapes of beakers (2) Fidelity: maintaining visual non-penetration for various grasps, and grasps resembling real-life grasping
Tong et al. 2022 [29]	Rigid hand with a sphere-tree palm and five cone-frustum fingers	Configuration-based optimization in a decoupled-and-progressive framework	Computing forces based on the position deviation of the tracked and graphic fingertips	(1) Scenarios: single-finger touching, two-finger pinching, and five-finger grasping with palm-object contacts (2) Update rate: over 1 kHz for the whole-hand interaction with about 250 contacts (3) Stability: the participants didn't feel perceptible penetration and instability during the above experiments

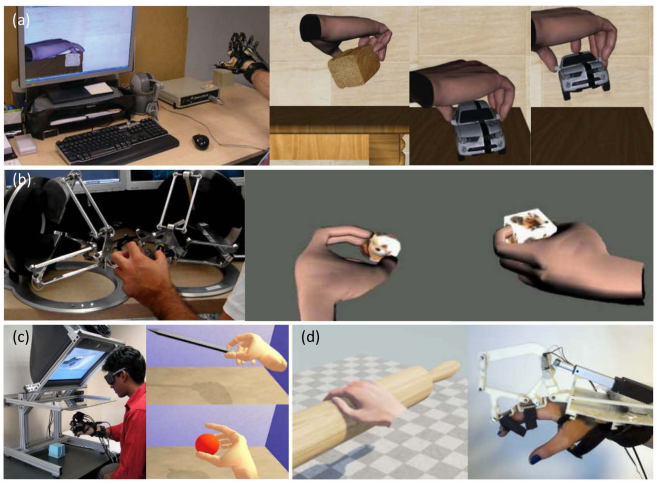

Fig. 5. Examples of kinesthetic simulation results. (a) Haptic interaction with a loaf of bread and a rigid car using a CyberGrasp^2 device [41]. (b) Virtual objects grasped by a left-handed operator using two 3-DoF haptic devices [6]. (c) Grasping various objects using the method in [54]. (d) Contact with a soft hand being rendered through an underactuated exoskeleton [66].

update rate due to its approximate nature, and it can be used for arbitrarily complex geometry. Wan et al. [27] used the collision proxy concept to remedy the problem of dynamically constructing point shells, implementing a VPS method [26] for the deformable virtual hand. The collision proxy, a collection of points on the virtual hand surface, was attached onto the fingertip for each finger and used as the finger's PointShell. The computation of the distance field can support fast collision detection, which is independent of the geometric complexity of objects. Nevertheless, the performance of distance-field-based collision detection depends on the storage structure and generation time of the distance field. To speed up the collision detection evaluation for hand-based interaction, Zhao et al. [34] precomputed the signed distance transformation of the object and simplified the runtime evaluation process as lookup table operations. Similarly, Verschoor et al. [56] computed a signed distance field for each virtual object at initialization to accelerate the hand-object collision detection, and thus collisions of vertices on the hand surface could be quickly checked by querying the distance field.

A two-stage strategy that includes coarse detection and fine detection [67] (e.g., spatial partitioning algorithms and bounding volume hierarchies-BVH) is commonly used to improve the efficiency of collision detection. Tzafestas et al. [68] developed a real-time collision detection algorithm for whole-hand kinesthetic feedback based on the spherical octree structure proposed in [69]. Jacobs et al. [57] performed collision detection between the deformable finger-pad and objects using the optimized spatial hashing described by Teschner et al. [70]. BVH, commonly using spheres [71] and boxes as an envelope for the shape of each virtual object, is capable of accelerating collision detection for dynamic scenes of rigid bodies. As shown in Fig. 6, when modeling a virtual hand by a sphere-tree model [28], low-hierarchy levels (such as level 1 and level 2) provide coarse detection, and the accuracy of collision detection grows higher with the level of hierarchy increases. Tong et al. [29] used a sphere tree to model the palm and achieved over 1 kHz collision detection for the whole hand interaction with five cone-frustum fingers. In addition, OBB-trees were usually used for fast collision detection

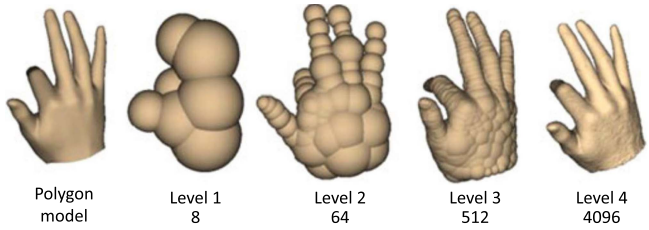

Fig. 6. Different-level sphere trees generated for a virtual hand [28]. The left-most figure is a polygonal model and the rest are sphere-tree models with different levels.

by building simplified structures for the virtual hand [19], [72], and Tian et al. [72] further sped up the hand-object collision detection by a queue-based technique [73] for parallel traversal of OBB-based BVH on GPUs.

The aforementioned collision detection methods used for hand-object interaction sample objects' trajectories at discrete times. Visual interpenetration can occur due to large discrete detection time steps. Moreover, details of hand-object interactions can be missed when objects are too thin or the virtual hand moves too fast. To handle virtual assembly simulation that often requires highly detailed models, Jacobs et al. [40] made a trade-off between the simulation quality and interactive simulation rate, and undesirable interpenetrations were inhibited based on the resulting accelerations of all bodies and continuous collision detection. Höll et al. [23] used the Continuous Dynamic mode of Unity for collision detection to support fast hand movements.

2) Collision Response: As one core component of haptic rendering, collision response is required to determine contact information (such as penetration distance, contact points, contact normals, etc.) for force computation. Heuristic-based approaches depend on the predefined heuristic or a priori information about the hand and/or object. These methods are mostly used for object grasping [74], [75], [76], and perform well for predefined manipulation behaviors. To realize more realistic and natural hand-based interaction, physics-based approaches including penalty-based and constraint-based approaches have attracted much attention for haptic simulation.

Penalty-based methods usually model contact constraints as springs. The elastic energy of springs increases when the haptic tool (i.e., haptic hand for hand-based interaction) penetrates into a manipulated virtual object. Penalty forces are computed to push the graphic tool (i.e., graphic hand for hand-based interaction) toward a non-penetration configuration [77]. In the early stage, single-point penalty-based models were built to compute the contact information between the virtual hand and virtual objects in VEs [78]. To address the problem of possible hand-object visual interpenetration for classical point-based contact handling approaches, Borst et al. [54] coupled a dynamic articulated hand model to the tracked hand configuration through a system of virtual linear and torsional spring-dampers. Hirota et al. [17] constructed a deformable hand using FEM. A co-rotation method was introduced to mitigate the effect of geometric nonlinearity caused by bending fingers. The friction contact between the virtual hand and an object was introduced by the penalty method [79]. The damping factor was integrated into Newton's and Euler's equations of motion for each of the contact points, ensuring the simulation stability during the motion computation. In addition, to ensure stable haptic rendering, Garre et al. [18], [41] used a viscoelastic coupling that was extended from the virtual coupling algorithm [80]. As stated in [41], such methods suffer from excessive smoothing under moderately slow visual thread updates due to the use of virtual coupling. Verschoor et al. [56] resorted to approximating the true behavior of skin deformation and friction to balance accuracy and interactivity. An energy-based formulation was designed for all the relevant mechanical and interaction elements: coupling between the hand's skeleton and the user's motion, constraints at skeletal joints, non-linear soft skin deformation, coupling between the hand's skeleton and the soft skin, frictional contact between the skin and virtual objects, and coupling between a grasped object and other virtual objects. Benefiting from the use of the efficient energy-minimization solver, this work enables natural and intuitive interactions between the virtual hand and VEs in real time.

Penalty-based collision response methods are favored in hand-based interaction because of their simple implementation, and many improvements have been made to obtain acceptable performance. When the contact state (e.g., the position of the contact point or the normal of the contact surface) changes, for example, a finger rapidly slides on an edge of virtual objects, contact forces may not be accurately solved. This phenomenon may lead to inconsistent visuo-haptic feedback (such as possible hand-object interpenetration), unstable contact forces, and an unrealistic sticking effect. Constraint-based methods aim to handle all contacts by solving a single computational problem, attempting to constrain the configuration of the graphic tool to be free of penetration. Compared with penalty-based methods, constraint-based methods can achieve accurate contact simulation. An issue worth paying attention to is that its computational complexity brings great challenges to accurately simulate contact mechanics at a haptic update rate using constraint-based methods, even for a linearized system.

Garre et al. [18] coupled the modeled deformable hand and the rigid handle by setting stiff damped springs between their corresponding nodes, and implicit integration methods were used to integrate the dynamics equations to ensure stiff yet stable hand-handle coupling. To efficiently solve the coupled dynamic problem, they proposed a constrained dynamics solver based on the computation of a Schur complement [81]. The exploration toward full-hand haptic interaction in [18] did not account for skeletal constraints. Garre et al. [41] further used Newton-Euler equations to describe the skeleton dynamics, and a linear co-rotational FEM approach for the deformable flesh. The coupled skeleton-flesh dynamics was solved based on constraint-based response in two steps: compute the bone velocities that satisfy joint constraints without varying the positions and velocities of the flesh nodes, and refine the bone velocities and compute the flesh velocities, accounting fully for the skeleton-flesh couplings with no joint constraints. Given that the God-object approach [82] can enable efficient computation of penetration-free interactions, Jacobs et al. [40] extended this

approach to support multi-finger interactions. An equation system was derived to simulate multiple God objects using Gauss' principle of least constraint. The phalanxes of each finger were tied together and limited in their local movement by constraints. A generalized acceleration vector was calculated by finding an optimal solution for the dynamics equations, and on this basis the constrained target positions were computed in Euler integration. This God-hand method stably supports multiple God objects with no undesirable interpenetrations based on a continuous collision detection [82] and the resulting accelerations of all bodies.

The simulation of a deformable hand is capable of facilitating stable hand-based interaction due to the potential of stable friction simulation. The softness of the fingerpads can produce a fast expansion of the contact surface at the initial contact, leading to intensive computation because the number of contact points increases. Some strategies, such as constraint separation [83], contact reduction/clustering methods [84], and volume constraints [85], have been reported to effectively handle complex contact scenarios involving large numbers of contact points. Talvas et al. [42] introduced aggregate constraints based on volume contact constraints [85] for dexterous grasping with a soft finger. All contacts on each phalanx were aggregated into a single set of separation and friction constraints, reducing the number of constraints. To accurately simulate the contact response at sharp edges of objects, a weighting method was introduced for simulating the non-uniform pressure distribution within aggregate constraints. Ultimately, a minimal set of constraints was obtained for accurately simulating dexterous manipulation of virtual objects with soft fingers at an update rate of tens of Hertz. Tong et al. [29] decoupled the high-dimensional optimization problem to several simple ones and solved configuration for each finger independently, achieving an update rate of higher than 1 kHz.

3) Force Computation: There are two main methods used to compute feedback force: direct mapping method and virtual coupling method. The direct mapping method regards the penalty force as the feedback force, which is directly output to haptic devices. A jump on the contact position or the contact normal may result in jumps of force and torque, and thus it is difficult to guarantee the stability of the feedback force using a direct mapping method. The virtual coupling method [80] connects the haptic device and a VE by a virtual spring and a virtual damper in parallel, which can simplify the problem of ensuring interactive stability.

Ott et al. [39] first introduced a direct mapping method to compute the force feedback. The force direction could be discontinuous for some specific examples. They then built a mass-spring hand model, and obtained the feedback force by computing the distance between the position of the tracked hand and the position of the mass-spring system. Borst et al. [54] extended the virtual spring-damper coupling commonly used in single-point devices (e.g., PHANToM) to hand-based haptic interaction using force-feedback gloves. Considering the limited degrees of freedom of these gloves, they reduced the forces and torques of the spring model for mapping to the glove actuators. They assumed one degree of freedom per fingertip

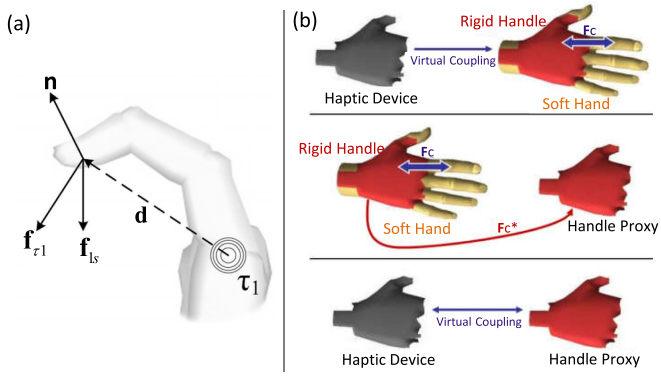

Fig. 7. (a) Components of force-feedback rendering in [54]. (b) Basic steps of haptic rendering based on handle-space force linearization [18].

and computed a scalar force at each fingertip as $\mathbf{f}_{feedback} = -(\alpha_1\mathbf{f}_{\tau 1} + \alpha_2\mathbf{f}_{\tau 2} + \alpha_3\mathbf{f}_{\tau 3} + \alpha_4\mathbf{f}_{ls}) \cdot \mathbf{n}$, where $\alpha_1, \alpha_2, \alpha_3$ and α_4 are user-chosen coefficients, $\mathbf{f}_{\tau 1}$,$\mathbf{f}_{\tau 2}$ and $\mathbf{f}_{\tau 3}$ are three forces related to flexion torques, $\mathbf{f}_{ls}$ is a force computed using a linear spring-damper by not considering palm base configuration during forward kinematics, and $\mathbf{n}$ is a unit vector describing the force direction of the actuator. The above computation equation of feedback force blends four force components (see Fig. 7(a)).

Garre et al. [18] computed the feedback force through a two-way virtual coupling (Fig. 7(b)-bottom) between the haptic device and the proxy handle. As shown in Fig. 7(b)-top, the haptic device was connected to a rigid handle on the hand model. In Fig. 7(b)-middle, $\mathbf{Fc}$ denotes the total coupling force between the handle and the virtual hand, and $\mathbf{Fc}^*$ denotes the actual coupling forces between the handle and the rest of the hand. To ensure stable haptic rendering, Garre et al. [41] extended the virtual coupling method [80], and they set a six-dimensional viscoelastic coupling (i.e., linear and torsional viscoelastic springs) [26] for each pair of tracked and proxy bones. Joint constraints for proxy bones were not modeled in a haptic thread to ensure computational efficiency. Tong et al. [29] calculated forces according to the position deviation between the tracked fingertip and graphic fingertip for each finger, and obtained force feedback using a Dexmo haptic glove [86].

Underactuated haptic devices possess the advantages of simplicity and affordability [66], [87]. In recent years, specific rendering algorithms have been explored for underactuated haptic devices. Lobo et al. [66] extended the typical proxy-based haptic rendering strategy for underactuated devices by defining two instances of the proxy. After obtaining displacements in the actuated subspace, the subspace proxy allows computing force commands that are optimally constrained to actuated DoFs. This method is highly favorable for multi-dimensional environments, it is however not efficient when the user's finger is modeled through rigid links and joints and the dimensionality of the system is reduced to 2-3 DoFs [88]. To tackle the issue of lacking controllability for underactuated exoskeletons, Sarac et al. [88] minimized the error between the desired and the actual output force in the force space and replaced the desired values with proxy values that the devices can satisfy, improving

the accuracy of perceived forces. When ignoring the user's intent during the proxy-based rendering process, underactuated devices may produce unwanted ghost forces. Lobo et al. [89] modeled the user's intent and designed an optimal rendering strategy based on anisotropic tracking impedance. The forces rendered by an underactuated device using their proposed strategy matched well with those rendered by a fully actuated device.

For the simulation of picking up an object, two stages need to be considered, i.e. in contact and in lifting while holding the object. Instead of using the direct mapping method and virtual coupling method, An et al. [90] explored the appropriate force that should be given in these two situations. By analyzing the virtual force acting on the user's fingers when in contact as well as manipulating (i.e., moving and tilting while holding) the object, they formulated a pendulum-based force model that includes the center of mass (CoM), the tilted angle, and the grip location, which are the main factors influencing grasping forces. This study considered manipulated objects with different shapes, and different grip locations in the experiment, and the user could successfully manipulate virtual objects in lifting and tilting actions through the computed forces.

B. Tactile Rendering

In recent years, the advent of cutaneous devices opens the door to direct hand interaction with cutaneous feedback. Different from kinesthetic feedback that requires external support, cutaneous devices convey cutaneous feedback to users by stimulating skin mechanoreceptors directly. In this section, we first briefly discuss current cutaneous display techniques and then review the main rendering methods.

1) Cutaneous Display Technologies: Recent years have witnessed the advent of cutaneous devices, which impose haptic stimuli directly on the hand to convey haptic information, such as local contact surface modulation, skin stretch, and ultrasound feedback. Current prevalent technologies include active surfaces, wearable cutaneous devices, and mid-air haptic interfaces. Active surfaces enable direct exploration and palpation of dynamically varying shapes, such as deformable crusts [91], [92]. We leave out these devices, which aim to reproduce predefined geometric shapes for the user's active touch.

Wearable cutaneous devices, varying in terms of their control strategy based on position control or force control, provide tactile feedback by stimulating skin directly with miniature electromechanical actuators. Such devices modulate contact pressure against the skin, typically at the finger pad. Some devices are built as a pin array that enables the display of surface shape to an area of the fingertip [93], others operate on the finger pad by translating and orienting a small mobile platform [94], [95], or using two translational DoFs tangent to the finger pad to produce lateral skin stretch by pulling [96]. When contact takes place on the skin, both contact force and contact area vary. The relationship between these two variables provides information about object softness [97].

Mid-air haptic interfaces enable both direct-touch and mid-air interaction, without the need to hold or wear any device. Air jets and ultrasound modulation are two main strategies for the mid-air haptic display. AIREAL [98] is designed to use a vortex, a ring of air that can travel large distances to hit a user's skin. Ultrasound devices [99], [100], [101] modulate the activation of an array of ultrasound transducers to create focal pressure around the user, which can give the user tactile cues of gestures, textures, and virtual objects.

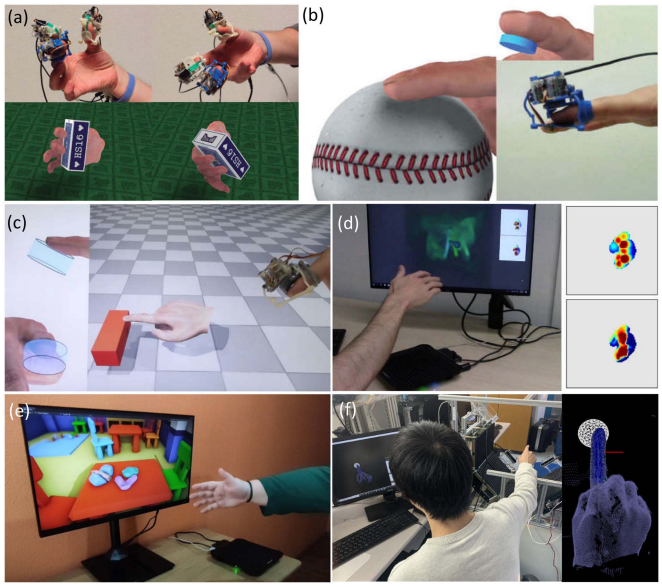

Fig. 8. Examples of tactile simulation results. (a) Tactile rendering for dexterous manipulation of a virtual object [103]. (b) Tactile rendering for the exploration of a ball [14]. (c) Tactile rendering based on skin stress optimization [102]. (d) Left: a user interacts with a fluid simulation on an Ultrahaptics device (left). Right: the extracted pressure field on the virtual hand (bottom), and the rendered pressure field to the user in the left image (top) [9]. (e) Arts & crafts in the virtual classroom with a tangible interactive experience of highly realistic deformations, merging, and splitting [104]. (f) A user discerns the softness of the virtual object via an ultrasound transducer array [105].

Currently, some tactile rendering methods have been proposed for haptic display using wearable cutaneous devices and ultrasound haptic devices (Fig. 8 and Table II). For these two types of haptic devices, haptic stimulation is limited to the passive outer layer of the hand skin. Different from kinesthetic rendering, tactile rendering is therefore not subject to stability problems due to latency [102]. For such tactile rendering, the update rate does not affect stability as in kinesthetic rendering, but rather affects the bandwidth of the interactions that can be rendered. The following will describe how to perform haptic rendering for wearable cutaneous devices and ultrasound haptic devices.

2) Rendering for Wearable Cutaneous Devices: To achieve high fidelity tactile rendering, it is vital to accurately simulate contact between a skin model and the virtual object. In early work, the contact model was commonly simplified to each finger pad. Barbagli et al. [106] proposed a one-dimensional deformation point-contact model with friction capable of supporting moments about the contact normal. Kawasaki et al. [107] presented a point contact model to compute the static and/or dynamic friction moment. Such contact models ignored the high-resolution mechanoreceptor density of finger skin, and simplified complex force fields perceived by the finger pad as

TABLE II
TYPICAL APPROACHES FOR HAND-BASED INTERACTION WITH CUTANEOUS FEEDBACK

Method	Features			Performance
	Hand model	Rendering method	Haptic device	
Perez et al. 2015 [108]	Skin: linear co-rotational FEM with strain-limiting constraints	LCSM: Levenberg-Marquardt nonlinear least-squares optimization	3-DoF wearable thimble, parallel mechanism	(1) Scenarios: a variety of contact interactions against different surfaces, such as an edge (2) Update rate: average 60 Hz (16ms for simulation and <1ms for optimization) (3) Fidelity: the simulation (single finger) is coherent with the real-world scenario
Perez et al. 2016 [103]	Skin: linear co-rotational FEM with strain-limiting constraints	LCSM: nonlinear Jacobi solver with Cholesky factorizations and parallelism	3-DoF wearable thimble [94], parallel mechanism	(1) Scenarios: grasping and rotating a virtual object with a deformable hand (2) Update rate: achieving more than 10× speed-up, 119 fps for free motion, 89 fps for <100 active constraints, and 54 fps for > 100 active constraints (3) Fidelity: enabling full-hand (nonlinear skin) interactive simulation
Perez et al. 2017 [14]	Skin: linear co-rotational FEM with strain-limiting constraints	LCSM: SLSQP sequential quadratic programming routine in NLopt	3-DoF wearable thimble [95], open-chain and parallel mechanisms	(1) Scenarios: exploring virtual surfaces with various properties (flat surface and edge) (2) Update rate: average 50 Hz (16ms for contact simulation, <1ms for unconstrained and 4ms for constrained device optimization) (3) Fidelity: outperforming plane-fitting and unconstrained optimization in force error
Verschoor et al. 2020 [102]	Soft hand in [56] with a data-driven skin model (distal phalanx of the index finger)	Skin stress optimization: nonlinear least-squares optimization by an active-set Gauss-Newton method	3-DoF wearable thimble [95], parallel mechanism	(1) Scenarios: evaluating the ability to render diverse virtual interactions, including smooth rolling, contact with edges, and frictional stick-slip motion (2) Update rate: 80 Hz (optimization plus friction update) (3) Fidelity: the stimuli produced by the device match closely the recorded target stimuli
Hirota et al. 2022 [110]	A finger model (the tetrahedral mesh is generated by the TetMesh program [111]) from BodyParts3d [112]	Deformation matching: minimizing the error between the target skin deformation and the skin deformation caused by the device	128-pin finger-mounted tactile device [113]	(1) Scenarios: comparing deformation matching with force matching in terms of the perception of the friction coefficient and of the perception of the friction direction (2) Update rate: non-real-time and driving the device with offline data (3) Fidelity: deformation matching performs better on the perception of the friction coefficient and friction direction than force matching
Barreiro et al. 2019 [9]	A hand (signed-distance representation with velocity field) driven by Leap Motion	AM: pressure field optimization including focal points (Lloyd's algorithm) and pressure magnitudes (quadratic function)	Ultrahaptics STRATOS Explore (USX) device	(1) Scenarios: smoke stirring scene and the smoke jet scene (2) Update rate: 90 Hz for simulation and 30 Hz for haptic rendering (3) Fidelity: the reconstruction depends on the number of focal points and their radii
Jang and Park 2020 [119]	A hand (point cloud) driven by Leap Motion	Adaptive AM: SPH-based pressure distribution computation, and the optimization of focal points using a hill-climbing algorithm	UHEV-1 [100]	(1) Scenarios: four spray conditions for qualitative and two for quantitative evaluation (2) Update rate: 90 Hz ~ 100 Hz for their haptic application (3) Fidelity: producing reliable estimates by increasing the smoothing length and number of focal points, and enhancing the reality of the fluid simulation with haptic feedback
Barreiro et al. 2020 [120]	A hand (voxelization) driven by Leap Motion	PRO-STM: the optimization of focal point paths including path initialization (2-opt algorithm) and path refinement (gradient descent)	Ultrahaptics STRATOS Explore (USX) device	(1) Scenarios: interaction with 1 and 4 plumes of smoke (2) Update rate: 50 Hz for fluid simulation on GPU and 25 Hz for PRO-STM on CPU (3) Fidelity: providing more details about the target to the users than AM
Barreiro et al. 2021 [104]	Soft hand in [56]	Clustering-based AM approach with PBD simulation of clay and a perceptual weight map	Ultrahaptics STRATOS Explore (USX) device	(1) Scenarios: a kindergarten table with colorful clay and a potter's wheel with clay (2) Update rate: approximately 25 fps (2ms for rendering algorithm) (3) Fidelity: enabling rich creative experiences with unprecedented realism
Matsubayashi et al. 2022 [125]	A non-rigid mesh model driven by four depth cameras [123]	Solving a minimization problem to optimize the sound field amplitude using Levenberg-Marquardt method	Ultrasound tactile display proposed by Inoue et al. [124]	(1) Scenarios: reproducing a randomly determined amplitude distribution, and controlling the scattered sound field for a hand with a certain grasping shape (2) Update rate: the computation time is 1.784 s on CPU (Core i7-10700K CPU 3.80 GHz) and 0.4793 s on GPU (NVIDIA GeForce RTX 3090) (3) Fidelity: this work reproduces the amplitude distribution of the sound field with higher accuracy than conventional algorithms

a single point. The force computed for this point is directly rendered by a device. Due to the inability to represent the relative orientation between the finger pad and virtual objects, a point contact cannot convey curvature information.

It is the potential to improve the accuracy of tactile rendering by simulating a high-dimensional contact between an accurate skin model and virtual objects. Perez et al. [108] explored contact surface matching as an approach for tactile rendering with local contact surface modulation (LCSM) devices. They simulated the skin using a strain-limiting deformation model [109] to characterize the extreme nonlinearities in human skin. After computing the contact surface to be rendered through the skin contact model, they computed the device configuration by solving an optimization problem-minimizing the deviation between the contact surface in the virtual environment and the contact surface rendered by the device. Benefiting from the surface contact matching strategy, this approach can display smooth and edge contact surfaces, and the orientation of the platform can successfully approximate the relative orientation between the finger pad and the surface under contact.

It is computationally complex to simulate nonlinear skin, which hampered the cutaneous stimulation for full-hand/multi-finger interaction. The work [108] implemented finger interaction instead of full-hand interaction for computational efficiency. Later, Perez et al. [103] incorporated an efficient solver based on a Jacobi relaxation scheme, which linearizes the nonlinear constraints on each Jacobi iteration and performs line-search optimization per iteration to guarantee a steady error reduction. This solver achieved more than 10× over previous approaches, and enabled multi-finger manipulation of a virtual object by adopting the contact surface matching tactile rendering algorithm [108].

Based on an accurate and efficient simulation of contact between a finger model and the virtual environment [103], [109], Perez et al. [14] further extended their precious contact surface matching approach [108] by incorporating device workspace constraints for both open-chain and parallel mechanisms, establishing a formal optimization-based framework for cutaneous rendering, such that the contact surface between the device and the actual finger matches as close as possible the contact surface in the virtual environment. Through force error analysis, it is verified that this constrained optimization method outperforms the unconstrained optimization method and plane-fitting heuristic method. As stated in this work, the contact surface matching approach is suitable for virtual objects that are rigid or stiffer than the finger pad. Due to the fast extension of contact area with soft objects even for very low forces, an LCSM device with a rigid mobile platform would fail to render such effects correctly.

The aforementioned LCSM methods did not account for the mechanical interaction between the haptic device and skin, which approximated the local contact geometry around a simulated finger. Verschoor et al. [102] formulated a constrained optimization to search the device configuration space while

producing the best-matching stress. A device-skin simulation model was built to estimate the skin stress distribution, which accounted for trajectory-dependent effects efficiently by staggering the computation of friction state and device configuration. To produce a smooth rendering output, they designed a data-driven model of skin mechanics that aims at avoiding the high computational cost of full device-skin simulations. The quantitative validation shows the close match between the stimuli produced by the device and the target stimuli recorded when interacting with real-world objects.

Different from the aforementioned studies that addressed tactile rendering methods for 3-DoF wearable devices, Hirota et al. [110] proposed a deformation matching method that minimized the error between the target skin deformation and the skin deformation caused by the device. They compared this deformation matching method with the force matching method that minimizes the error between the target force and the acting force of the device. These two methods rendered frictional contacts between the finger [111], [112] and a plane using a pin-array tactile display [113] with the ability of simulating 128 points, and the results show that deformation matching performs better on the perception of the friction coefficient and friction direction than force matching.

3) Rendering for Ultrasound Devices: As a mid-air haptic display technology, ultrasound haptics [114], [115] employs an array of ultrasound transducers as actuators to produce high-frequency pressure waves, which enables direct touch sensations on the skin. Ultrasound haptics possesses the capabilities of modulating multiple tactile focal points at the same time, which promises the possibility of a more immersive and scalable virtual touch experience.

Long et al. [116] proposed Amplitude Modulation (AM) to control the pressure intensity at focal points by solving an eigenproblem. Spatiotemporal Modulation (STM) [117], [118] was also studied to produce target shapes by moving a single point at high speed, in which parameters (e.g., the velocity of the focal point) are required to be tuned for optimal sensitivity. Barreiro et al. formulated the computation of the rendered pressure as an optimization problem, and optimized the location and intensity of focal points based on AM for the interaction with fluid [9] (Fig. 8(d)) and clay material [104] (Fig. 8(e)). Jang and Park [119] computed the pressure distribution of the rigid-fluid interaction, and reconstructed a target pressure field by solving their proposed AM as an optimization using a hill-climbing method. To address the AM's issue of perceivable vibration on the skin (typically at 200 Hz), Barreiro et al. [120] further proposed path routing optimization for STM (PRO-STM) to render the force distribution resulting from a dynamic virtual interaction, which provides larger and smoother coverage compared with AM rendering method [9].

Matsubayashi et al. [121], [122] focused on manipulating 3D objects with ultrasound haptic feedback. They estimated the cross section of the finger in contact with the virtual object and enabled the user to recognize the object surface for quick grasping through the pressure distribution on the finger [121]. This method was extended for any 3D polygon mesh model by generating time-averaged pressure distribution through the moving of the ultrasound focus along the points of the finger surface near the polygon surface with a high speed [122]. Matsubayashi et al. then formulated the rendering of pressure distribution as an optimization problem of phases of transducers or together with phases of target pressure distribution, reproducing the tactile sensation of touching a soft object [105] and accurate pressure pattern on the hand surface [123] via an ultrasound transducer array [124]. Matsubayashi et al. [125] further evaluated the performance of the Levenberg-Marquardt method for optimizing the transducer phase, showing its higher accuracy in reproducing the amplitude distribution of the sound field than conventional algorithms. Due to the high computational cost, this method is however difficult to control the pressure distribution. Plasencia et al. [126] proposed a GPU phase retrieval algorithm (GS-PAT), which achieved 17 kHz computing for up to 32 simultaneous points of ultrasound fields on NVIDIA GTX 1660, enabling multi-point spatio-temporal modulation of drawing speed and frequency independently or multi-frequency stimulation.

Howard et al. [115] proposed a robotic solution to increase the usable workspace of ultrasound mid-air haptic devices, thus to enable the adequate synchronization and co-location of ultrasound tactile feedback with other stimuli for VR interaction. An open-source software framework [127] was also proposed for the stimulus design of the ultrasound tactile rendering and perceptual evaluation. Some studies have evaluated the perception of focused ultrasound haptic stimuli [127], [128]. Perquin et al. [129] studied the discrimination of motion direction and evidenced the directional bias for dynamic STM patterns. Howard et al. [130] investigated the perception of continuity or gaps between neighboring AM focal focused ultrasound stimuli.

V. Perspectives for Research

The existing studies reviewed above have produced delightful visuo-haptic interaction experiences. This section will elaborate future research topics by identifying the gaps between existing technology and natural hand-based haptic interaction.

A. Realistic Visual Display of Hand-Object Interaction

Tables I and II summarize typical studies for hand-based interaction with kinesthetic and tactile feedback. Existing studies support hand-based interaction with rigid objects well, and usually involving a single object. For the interaction with complex-material objects, such as clay-like objects [104], and fluids [9], the update rate of full simulation and haptic rendering is tens of Hertz. Such a low update rate results from the high complexity of modeling complex-material objects such as nonlinear deformations and fluids, which also brings challenges to interactive realism. In particular, when the user interacts with a virtual environment with multiple objects of different properties, such as grasping a fish in a fish tank, it is necessary to integrate different models. The coupling between different physical properties requires to be performed in an efficient manner for a realistic experience. Subspace methods facilitate efficient computation through model reduction [131], providing a potential way for modeling complex scenarios.

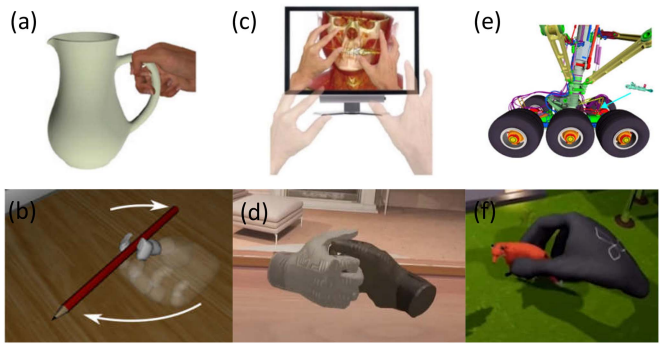

Fig. 9. Examples of complex manipulation behaviors. (a) Grasping a vase with multiple fingers [72]. (b) Rotating a pencil with three fingers [42]. (c) Palpating and identifying the stiffness, the location and the geometric boundary of a suspicious lump in a surgery [138]. (d) The interaction between two hands.[3] (e) Assembling small-size parts in virtual assembly/maintenance [139]. (f) Stroking a furry animal.[4]

For virtual hand modeling, multi-layer hand avatars including skin, soft tissue and skeletal bone have been constructed to mimic the structure of the human hand [41], [56]. These hand avatars enable kinesthetic ability or cutaneous features for hand-based interaction with haptic feedback. In the real world, the movement of palm bows allows the human hand to manipulate objects of different sizes and shapes. Currently, the palm is usually represented as a rigid body without metacarpals, which results in a lack of freedom of motion in palm bows, making it difficult to manipulate diverse objects. To improve the realism of virtual hands for manipulating various virtual objects, it is necessary to build a deformable palm with a high degree of freedom in addition to deformable fingers and map the motion of the real hand to drive the virtual hand accurately and efficiently. There are two constraints to achieving the above goals, one is the high complexity of virtual hand modeling, and the other is the limitation of hand tracking and mapping, such as high-DoF motion matching between the real hand and virtual hand, especially for the thumb and palm, and the adaptability to different hand sizes (e.g., width, length, and circumference). Data-driven methods possess advantages of high efficiency and accuracy for contact-free static poses, and subspace simulation methods provide a fast way to approximate soft-tissue deformations [131], [132]. A hybrid of such two approaches has the potential to enhance the realism of hand-based interactions.

B. Haptic Simulation With Multiple Dynamic Contacts

Figs. 5 and 8 show typical results of existing hand-based haptic simulation methods. We can see that the typical grasping and touching of rigid objects by one hand or a finger can be simulated and rendered with haptic feedback. It however remains challenging to mimic complicated contact situations in a natural manner, including multi-point contacts during multi-finger cooperative manipulations (Fig. 9(a)—a soft hand grasps a vase with multiple fingers, and Fig. 9(b)—three fingers rotate a pencil), and contacts between the virtual hand and virtual objects with special shapes (thin-wall objects and thin lines [28], [133], etc.) or complex physical properties (deformable objects [104], [134], [135], etc.). The former challenge results primarily from the current limited computational and handling ability for various distributed contacts due to the complexity of contact modeling, and thus hand-object penetration and interaction instability might occur. The latter requires focusing on the simulation of complex physical properties and contact dynamics, which is time-consuming for natural interaction, leading to possible missing of contact determination when the hand moves fast. These challenges will result in providing erroneous information for the computation of force and/or corresponding configuration information, affecting stable haptic display and concurrent visuo-haptic interaction experience.

For bimanual/multi-hand cooperative manipulations (e.g., bimanual manipulation — Figs. 1(d) and 9(c)), haptic simulation involves the coupling of mechanical behavior of multiple hand avatars (e.g., force coordination of these hands) to maintain the manipulation stability. Compared with single-hand interactions, multi-hand interactions involve more complex contact modeling, and coupled dynamics constraints needs to be constructed under distributed contacts with more contacts point/areas to keep stable cooperative manipulation. In addition, two or more hand avatars may collide with each other while interacting with virtual objects, or perhaps multiple users communicate with each other through hand-hand interactions (e.g., Fig. 9(d)). Under such circumstances, the visual effects and haptic sensation induced by hand-hand contacts need to be concerned for the interactive haptic simulation. Hand-hand interaction involves the contact modeling of multiple high-DoF objects (i.e., hand avatars), and high-dimensional computation should perform at high update rates for haptic display.

The human hand is skilled in performing fine manipulations, such as palpating and identifying the stiffness, location and geometric boundary of a suspicious lump in medical procedures (e.g., Fig. 9(c)), assembling small-size parts in assembly/maintenance tasks (e.g., Fig. 9(e)), and stroking a furry animal (e.g., Fig. 9(f)). For such fine manipulations, multi-modal (e.g., coupling kinesthetic and cutaneous feedback) and distributed haptic feedback are crucial for stable and realistic simulation, which requires not only multi-modal haptic devices that can provide multi-point/region haptic stimuli but also efficient handling of distributed contacts. Due to the need for complex and time-consuming computation, nonlinear skin modeling and constrained dynamics solver are two main bottlenecks for simulating a rich haptic experience. Recent years have witnessed the rapid development of data-driven haptic modeling and rendering, such as sliding and sticking contact with deformable objects [136], and the processing of high-dimensional haptic interaction signals [137], providing possible solutions for multi-modal haptic feedback.

As elaborated above, contact modeling is the core of haptic simulation [140]. The exploration of human hand behaviors in manipulating objects [141], [142], [143] helps to reveal hand manipulation characteristics, and the understanding of contact characteristics for real-world interactions will help in modeling

[3][Online]. Available: https://www.youtube.com/watch?v=h5WzF1ch3ww
[4][Online]. Available: https://www.youtube.com/watch?v=OK2y4Z5IkZ0

hand-object and hand-hand contact. A statistical model including fingertip deformations and contact forces can be created using the data captured during the real finger-surface interaction [144]. Due to the nonlinear behavior of the human hand, it is however challenging to obtain the spatial distribution of the mechanical properties in real interaction scenarios. With the limited real-world contact data, the differences in visuo-haptic interaction experience between the virtual and real manipulations may provide guidelines to design novel yet simple paradigms for contact modeling.

C. Validation of Visuo-Haptic Interaction Experience

Tables I and II show some metrics used to evaluate the performance of hand-based haptic simulation. In reality, most studies focus on the display of visual realism, including the non-penetration simulation between the virtual hand and virtual objects (such as Fig. 5 and Fig. 8) and realistic deformable effects of the virtual soft hand/fingers [41], [56]. It is challenging to define meaningful metrics that can be quantitatively measured for haptic display, which has been one significant bottleneck that plagues the quantitative validation of the haptics effect [102].

As mentioned earlier, a high update rate is required for kinesthetic rendering to simulate high contact stiffness stably, and cutaneous rendering does not impose high impedance or update rate restrictions for stability and is safer since its actions do not directly affect the input of the system (i.e., the position of the hand). When providing these two-type haptic stimuli at the same time through multi-modal haptic rendering, does the user's nervous system integrate the haptic information in a similar manner to real interactions? It is necessary to evaluate the user's experience and corresponding neural activities for the case of multi-modal haptic feedback.

Studies showed that the integration of visuo-haptic inputs can aid in object recognition and perception [145], and concurrent visuo-haptic inputs can aid in a more accurate perception of stimuli, over solely visual information [146]. With the advent of virtual reality technology, it is an interesting and important problem that should be explored: do the enhancements of the realistic visuo-haptic display necessarily lead to an improved subjective experience for users? Such a problem is inspired by the notion of the uncanny valley from the field of humanoid robotics [147]. Berger et al. [148] explored whether an uncanny valley also exists for human haptic perception that might be rendered during human-robot interaction, teleoperation, or other virtual manipulation tasks in virtual environments, and found that enhanced haptic feedback inconsistent with other sensory cues can reduce subjective realism, producing an uncanny valley of haptics. Gehrke et al. [149] automatically detected visuo-haptic mismatches in VR using event-related brain potentials. D'Alonzo et al. [150] found discordant levels of visualization of sight and touch elicited revulsion, which extended the concept of the uncanny valley to avatar embodiment.

In addition, humans inherently interact with the real world through multiple senses, primarily including sight, hearing, touch, smell, and taste. The exploration of multi-modal interaction, such as the integration of multiple sensory channels, the creation of natural and compelling multi-modal interfaces remains a grand challenge [151], [152].

D. Software Platform for Hand-Based Haptic Interaction

Developing a software platform for hand-based haptic simulation is beneficial in many aspects. An extensible software platform allows researchers to conveniently develop hand-based haptic applications, as well as objectively compare different haptic simulation approaches on a unified platform, which helps to boost the rapid development of hand-based haptic interaction by sharing and comparing. Currently, there is still a lack of accessible software platforms to support hand-based haptic interaction. A few pioneering works for hand-based interaction such as CLAP[5] [56], [104] aims to provide a realistic hand simulator for real-time applications, and it features real-time hand simulation with soft skin, supporting hand touching and grasping of virtual objects. Haptic devices and haptic rendering algorithms are two key components for the development and application of a haptic interactive software platform. Due to the shortage of high-performance haptic gloves and high-fidelity haptic rendering algorithms that can provide users' hands with robust and realistic sensations of interactions with various virtual environments, it is still a challenge to develop a hand-based haptic interactive software platform for academic exchange and commercial development.

To facilitate the development of hand-based interaction applications with visuo-haptic feedback, a software platform should satisfy the following requirements: (1) support multi-modal haptic feedback, including softness, friction, texture, vibration, temperature, etc.; (2) allow convenient access to different devices, including various visual display devices, haptic devices, and hand motion tracking devices; (3) support spatiotemporal consistency of multi-modal haptic and immersive visual feedback; (4) support the easy development of 3D interactive scenes for different application fields, such as medicine, business, and entertainment.

A three-layer architecture consisting of a low update rate physical engine layer (such as Unity, the update rate is about 60 Hz), a high update rate (e.g. 1 kHz) haptic rendering layer, and a hardware control layer might be a potential solution to meet the above requirements. The physical engine layer is used for the construction of interaction scenarios and immersive visual display. The haptic rendering layer performs contact handling at a haptic update rate, and computing the desired control command for the haptic device. The hardware control layer needs to be compatible with different devices. The software platform can be developed by exploiting GPU acceleration to obtain a high update rate for haptic display.

VI. Conclusion

In this survey, we focused on the interactive simulation between the virtual hand and virtual objects with haptic feedback

[5][Online]. Available: https://clapxr.com/

in terms of its computing components that mainly includes virtual hand modeling, as well as hand-based haptic rendering for kinesthetic and tactile feedback. Most of existing studies implement plausible hand models with limited DoFs of hand motion, and support the hand-object interaction either with kinesthetic feedback or tactile feedback that does not involve complex dynamic contacts.

There are still gaps between existing technologies and natural interaction between the hand and virtual environments with high-fidelity haptic feedback, including the lack of realistic hand avatars, the inability to support complicated interactive scenarios (e.g., hand-based haptic interaction with deformable and fluid objects, and bimanual/multi-hand cooperative manipulations), the lack of objective validation of visuo-haptic fusion experience, and the lack of extensible software platform for hand-based haptic interaction. Future research to fill these gaps toward natural hand-based interaction with richer haptic feedback and concurrent visuo-haptic display will bring rapid development for potential applications, such as medicine, business, and entertainment.

REFERENCES

[1] M. O. Ernst and M. S. Banks, "Humans integrate visual and haptic information in a statistically optimal fashion," *Nature*, vol. 415, no. 6870, pp. 429–433, 2002.

[2] J. F. Norman et al., "Solid shape discrimination from vision and haptics: Natural objects (Capsicum annuum) and Gibson's 'feelies'," *Exp. Brain Res.*, vol. 222, no. 3, pp. 321–332, 2012.

[3] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," *IEEE Trans. Syst., Man, Cybern., Part C. (Appl. Rev.)*, vol. 38, no. 4, pp. 461–482, Jul. 2008.

[4] C. Pacchierotti, S. Sinclair, M. Solazzi, A. Frisoli, V. Hayward, and D. Prattichizzo, "Wearable haptic systems for the fingertip and the hand: Taxonomy, review, and perspectives," *IEEE Trans. Haptics*, vol. 10, no. 4, pp. 580–600, Oct.–Dec. 2017.

[5] D. Wang, M. Song, A. Naqash, Y. Zheng, W. Xu, and Y. Zhang, "Toward whole-hand kinesthetic feedback: A survey of force feedback gloves," *IEEE Trans. Haptics*, vol. 12, no. 2, pp. 189–204, Apr.–Jun. 2019.

[6] 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, First Quarter 2013.

[7] 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.

[8] G. Cirio, M. Marchal, A. Lécuyer, and J. R. Cooperstock, "Vibrotactile rendering of splashing fluids," *IEEE Trans. Haptics*, vol. 6, no. 1, pp. 117–122, First Quarter 2013.

[9] H. Barreiro, S. Sinclair, and M. A. Otaduy, "Ultrasound rendering of tactile interaction with fluids," in *Proc. IEEE World Haptics Conf.*, 2019, pp. 521–526.

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

[11] 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.

[12] D. Wang, K. Ohnishi, and W. Xu, "Multimodal haptic display for virtual reality: A survey," *IEEE Trans. Ind. Electron.*, vol. 67, no. 1, pp. 610–623, Jan. 2020.

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

[14] A. G. Perez et al., "Optimization-based wearable tactile rendering," *IEEE Trans. Haptics*, vol. 10, no. 2, pp. 254–264, Apr.–Jun. 2017.

[15] J. Napier, J. R. Napier, and R. H. Tuttle, *Hands*, vol. 9. Princeton Univ. Press, 1993.

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

[17] K. Hirota and K. Tagawa, "Interaction with virtual object using deformable hand," in *Proc. IEEE Virtual Reality*, 2016, pp. 49–56.

[18] C. Garre and M. A. Otaduy, "Toward haptic rendering of full-hand touch," in *Proc. CEIG*, 2009, pp. 19–26.

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

[20] M. Feng and J. Li, "Real-time deformation simulation of hand-object interaction," in *Proc. IEEE 5th Int. Conf. Robot., Automat. Mechatron.*, 2011, pp. 154–157.

[21] D. Chamoret, M. Bodo, and S. Roth, "A first step in finite-element simulation of a grasping task," *Comput. Assist. Surg.*, vol. 21, no. sup1, pp. 22–29, 2016.

[22] K. Nasim and Y. J. Kim, "Physics-based interactive virtual grasping," in *Proc. HCI Korea*, 2016, pp. 114–120.

[23] M. Höll, M. Oberweger, C. Arth, and V. Lepetit, "Efficient physics-based implementation for realistic hand-object interaction in virtual reality," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2018, pp. 175–182.

[24] H. Wan, F. Chen, and X. Han, "A 4-layer flexible virtual hand model for haptic interaction," in *Proc. IEEE Int. Conf. Virtual Environ., Hum.-Comput. Interfaces Meas. Syst.*, 2009, pp. 185–190.

[25] M. Rosenberg and J. M. Vance, "Virtual hand representations to support natural interaction in immersive environments," in *Proc. Int. Des. Eng. Tech. Conf.s Comput. Inf. Eng. Conf.*, 2013, Art. no. V02BT02A028.

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

[27] H. Wan, Y. Luo, S. Gao, and Q. Peng, "Realistic virtual hand modeling with applications for virtual grasping," in *Proc. ACM SIGGRAPH Int. Conf. Virtual Reality Continuum Its Appl. Ind.*, 2004, pp. 81–87.

[28] 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.

[29] Q. Tong et al., "Configuration-based optimization for virtual hand haptic simulation," *IEEE Trans. Haptics*, vol. 15, no. 3, pp. 613–625, Jul.–Sep. 2022.

[30] C. B. Zilles and J. K. Salisbury, "A constraint-based god-object method for haptic display," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Hum. Robot Interact. Cooperative Robots*, 1995, pp. 146–151.

[31] D. E. Johnson, P. Willemsen, and E. Cohen, "Six degree-of-freedom haptic rendering using spatialized normal cone search," *IEEE Trans. Vis. Comput. Graph.*, vol. 11, no. 6, pp. 661–670, Nov./Dec. 2005.

[32] Y. Ding, J. Bonse, R. Andre, and U. Thomas, "In-hand grasping pose estimation using particle filters in combination with haptic rendering models," *Int. J. Humanoid Robot.*, vol. 15, no. 01, 2018, Art. no. 1850002.

[33] A. Guy and B. Wyvill, "Controlled blending for implicit surfaces using a graph," in *Proc. Implicit Surfaces*, 1995, pp. 107–112.

[34] W. Zhao, J. Zhang, J. Min, and J. Chai, "Robust realtime physics-based motion control for human grasping," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–12, 2013.

[35] J.-S. Kim and J.-M. Park, "Direct hand manipulation of constrained virtual objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 357–362.

[36] J.-S. Kim and J.-M. Park, "Physics-based hand interaction with virtual objects," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 3814–3819.

[37] J.-S. Kim and J.-M. Park, "Direct and realistic handover of a virtual object," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 994–999.

[38] R. Ott, V. De Perrot, D. Thalmann, and F. Vexo, "MHaptic: A haptic manipulation library for generic virtual environments," in *Proc. Int. Conf. Cyberworlds*, 2007, pp. 338–345.

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

[40] 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.

[41] 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.

[42] 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.

[43] S. Jain and C. K. Liu, "Controlling physics-based characters using soft contacts," in *Proc. SIGGRAPH Asia Conf.*, 2011, pp. 1–10.

[44] W. Tsang, K. Singh, and E. Fiume, "Helping hand: An anatomically accurate inverse dynamics solution for unconstrained hand motion," in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, 2005, pp. 319–328.

[45] M. Santello, M. Flanders, and J. F. Soechting, "Postural hand synergies for tool use," *J. Neurosci.*, vol. 18, no. 23, pp. 10105–10115, 1998.

[46] H. Liu et al., "High-fidelity grasping in virtual reality using a glove-based system," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 5180–5186.

[47] R. Wang, S. Paris, and J. Popović, "6D hands: Markerless hand-tracking for computer aided design," in *Proc. 24th Annu. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 549–558.

[48] J. L. Raheja, A. Chaudhary, and K. Singal, "Tracking of fingertips and centers of palm using kinect," in *Proc. IEEE 3rd Int. Conf. Comput. Intell., Model. Simul.*, 2011, pp. 248–252.

[49] A. Ahmad, C. Migniot, and A. Dipanda, "Tracking hands in interaction with objects: A review," in *Proc. IEEE 13th Int. Conf. Signal-Image Technol. Internet-Based Syst.*, 2017, pp. 360–369.

[50] R. Fujiki, D. Arita, and R.-I. Taniguchi, "Real-time 3D hand shape estimation based on inverse kinematics and physical constraints," in *Proc. Int. Conf. Image Anal. Process.*, 2005, pp. 850–858.

[51] A. Chabrier, F. Gonzalez, F. Gosselin, and W. Bachta, "Analysis of the directions in which forces are applied on the hand during manual manipulation and exploration," in *Proc. IEEE World Haptics Conf.*, 2015, pp. 280–285.

[52] P. Kumar, J. Verma, and S. Prasad, "Hand data glove: A wearable real-time device for human-computer interaction," *Int. J. Adv. Sci. Technol.*, vol. 43, pp. 15–26, 2012.

[53] H. Du, W. Xiong, and Z. Wang, "Modeling and interaction of virtual hand based on virtools," in *Proc. IEEE Int. Conf. Multimedia Technol.*, 2011, pp. 416–419.

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

[55] G. Zachmann, "Real-time and exact collision detection for interactive virtual prototyping," in *Proc. ASME Des. Eng. Tech. Conf.*, 1997, pp. 14–17.

[56] 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.

[57] J. Jacobs and B. Froehlich, "A soft hand model for physically-based manipulation of virtual objects," in *Proc. IEEE Virtual Reality Conf.*, 2011, pp. 11–18.

[58] A. R. Rivers and D. L. James, "FastLSM: Fast lattice shape matching for robust real-time deformation," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 82–87, 2007.

[59] M. Pouliquen, C. Duriez, C. Andriot, A. Bernard, L. Chodorge, and F. Gosselin, "Real-time finite element finger pinch grasp simulation," in *Proc. IEEE 1st Joint Eurohaptics Conf. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst. World Haptics Conf.*, 2005, pp. 323–328.

[60] J. E. Chadwick, D. R. Haumann, and R. E. Parent, "Layered construction for deformable animated characters," in *Proc. 16th Annu. Conf. Comput. Graph. Interactive Techn.*, 1989, pp. 243–252.

[61] J.-P. Gourret, N. M. Thalmann, and D. Thalmann, "Simulation of object and human skin formations in a grasping task," in *Proc. 16th Annu. Conf. Comput. Graph. Interactive Techn.*, 1989, pp. 21–30.

[62] N. Magnenat-Thalmann, R. Laperrire, and D. Thalmann, "Joint-dependent local deformations for hand animation and object grasping," in *Proc. Graphics Interface*, 1988, pp. 26–33.

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

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

[65] K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles, "Haptic rendering: Programming touch interaction with virtual objects," in *Proc. Symp. Interactive 3D Graph.*, 1995, pp. 123–130.

[66] D. Lobo, M. Saraç, M. Verschoor, M. Solazzi, A. Frisoli, and M. A. Otaduy, "Proxy-based haptic rendering for underactuated haptic devices," in *Proc. IEEE World Haptics Conf.*, 2017, pp. 48–53.

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

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

[69] C. Tzafestas and P. Coiffet, "Real-time collision detection using spherical octrees: Virtual reality application," in *Proc. IEEE 5th Int. Workshop Robot Hum. Commun.*, 1996, pp. 500–506.

[70] M. Teschner, B. Heidelberger, M. Müller, D. Pomerantes, and M. H. Gross, "Optimized spatial hashing for collision detection of deformable objects," in *Proc. VMV*, 2003, pp. 47–54.

[71] P. M. Hubbard, "Approximating polyhedra with spheres for time-critical collision detection," *ACM Trans. Graph.*, vol. 15, no. 3, pp. 179–210, 1996.

[72] H. Tian, C. Wang, D. Manocha, and X. Zhang, "Realtime hand-object interaction using learned grasp space for virtual environments," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 8, pp. 2623–2635, Aug. 2019.

[73] J. Pan and D. Manocha, "GPU-based parallel collision detection for fast motion planning," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 187–200, 2012.

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

[75] M. Moehring and B. Froehlich, "Pseudo-physical interaction with a virtual car interior in immersive environments," in *Proc. IPT/EGVE*, 2005, pp. 181–189.

[76] 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.

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

[78] M. de Pascale, G. Sarcuni, and D. Prattichizzo, "Real-time soft-finger grasping of physically based quasi-rigid objects," in *Proc. IEEE 1st Joint Eurohaptics Conf. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst. World Haptics Conf.*, 2005, pp. 545–546.

[79] C. A. Felippa, "A systematic approach to the element-independent corotational dynamics of finite elements," Dept. Aerosp. Eng. Sci. Center Aerosp. Struct., Tech. Rep. CU-CAS-00-03, 2000.

[80] 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 Interaction Cooperative Robots*, 1995, pp. 140–145.

[81] G. Golub and C. Van Loan, *Matrix Computations*. Baltimore, MD, USA: JHU Press, 1996.

[82] 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.

[83] E. Miguel and M. A. Otaduy, "Efficient simulation of contact between rigid and deformable objects," *ECCOMAS-Multibody Dyn.*, vol. 4, pp. 25–35, 2011.

[84] Y. J. Kim, M. A. Otaduy, M. C. Lin, and D. Manocha, "Six-degree-of-freedom haptic rendering using incremental and localized computations," *Presence: Teleoperators Virtual Environ.*, vol. 12, no. 3, pp. 277–295, 2003.

[85] 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, pp. 1–10.

[86] 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.

[87] J. Iqbal, N. Tsagarakis, and D. Caldwell, "Human hand compatible underactuated exoskeleton robotic system," *Electron. Lett.*, vol. 50, no. 7, pp. 494–496, 2014.

[88] M. Sarac, M. Solazzi, M. A. Otaduy, and A. Frisoli, "Rendering strategies for underactuated hand exoskeletons," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 2087–2092, Jul. 2018.

[89] D. Lobo and M. A. Otaduy, "Rendering of constraints with underactuated haptic devices," *IEEE Trans. Haptics*, vol. 13, no. 4, pp. 699–708, Oct.–Dec. 2020.

[90] Y. An, H. Cho, and J. Park, "Virtual whole-hand grasping feedback for object manipulation with a two-finger haptic interface," *J. Comput. Sci. Eng.*, vol. 14, no. 2, pp. 41–51, 2020.

[91] A. A. Stanley and A. M. Okamura, "Controllable surface haptics via particle jamming and pneumatics," *IEEE Trans. Haptics*, vol. 8, no. 1, pp. 20–30, Jan.–Mar. 2015.

[92] A. A. Stanley and A. M. Okamura, "Deformable model-based methods for shape control of a haptic jamming surface," *IEEE Trans. Visual. Comput. Graph.*, vol. 23, no. 02, pp. 1029–1041, Feb. 2017.
[93] I. Sarakoglou, N. Garcia-Hernandez, N. G. Tsagarakis, and D. G. Caldwell, "A high performance tactile feedback display and its integration in teleoperation," *IEEE Trans. Haptics*, vol. 5, no. 3, pp. 252–263, Third Quarter 2012.
[94] D. Prattichizzo, F. Chinello, C. Pacchierotti, and M. Malvezzi, "Towards wearability in fingertip haptics: A 3-DoF wearable device for cutaneous force feedback," *IEEE Trans. Haptics*, vol. 6, no. 4, pp. 506–516, Oct.–Dec. 2013.
[95] F. Chinello, M. Malvezzi, C. Pacchierotti, and D. Prattichizzo, "Design and development of a 3RRS wearable fingertip cutaneous device," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatron.*, 2015, pp. 293–298.
[96] D. Leithinger, S. Follmer, A. Olwal, and H. Ishii, "Shape displays: Spatial interaction with dynamic physical form," *IEEE Comput. Graph. Appl.*, vol. 35, no. 5, pp. 5–11, Sep./Oct. 2015.
[97] M. A. Otaduy, A. Okamura, and S. Subramanian, "Haptic technologies for direct touch in virtual reality," in *Proc. ACM SIGGRAPH Courses*, 2016, pp. 1–123.
[98] R. Sodhi, I. Poupyrev, M. Glisson, and A. Israr, "Aireal: Interactive tactile experiences in free air," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–10, 2013.
[99] T. Hoshi, M. Takahashi, T. Iwamoto, and H. Shinoda, "Noncontact tactile display based on radiation pressure of airborne ultrasound," *IEEE Trans. Haptics*, vol. 3, no. 3, pp. 155–165, Jul.–Sep. 2010.
[100] T. Carter, S. A. Seah, B. Long, B. Drinkwater, and S. Subramanian, "UltraHaptics: Multi-point mid-air haptic feedback for touch surfaces," in *Proc. 26th Annu. ACM Symp. User Interface Softw. Technol.*, 2013, pp. 505–514.
[101] O. Georgiou, W. Frier, E. Freeman, C. Pacchierotti, and T. Hoshi, *Ultrasound Mid-Air Haptics for Touchless Interfaces*. Berlin, Germany: Springer, 2022.
[102] M. Verschoor, D. Casas, and M. A. Otaduy, "Tactile rendering based on skin stress optimization," *ACM Trans. Graph.*, vol. 39, no. 4, pp. 90–1, 2020.
[103] A. G. Perez, G. Cirio, D. Lobo, F. Chinello, D. Prattichizzo, and M. A. Otaduy, "Efficient nonlinear skin simulation for multi-finger tactile rendering," in *Proc. IEEE Haptics Symp.*, 2016, pp. 155–160.
[104] H. Barreiro, J. Torres, and M. A. Otaduy, "Natural tactile interaction with virtual clay," in *Proc. IEEE World Haptics Conf.*, 2021, pp. 403–408.
[105] A. Matsubayashi, T. Yamaguchi, Y. Makino, and H. Shinoda, "Rendering softness using airborne ultrasound," in *Proc. IEEE World Haptics Conf.*, 2021, pp. 355–360.
[106] F. Barbagli, A. Frisoli, K. Salisbury, and M. Bergamasco, "Simulating human fingers: A soft finger proxy model and algorithm," in *Proc. IEEE 12th Int. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst.*, 2004, pp. 9–17.
[107] H. Kawasaki, Y. Ohtuka, S. Koide, and T. Mouri, "Perception and haptic rendering of friction moments," *IEEE Trans. Haptics*, vol. 1, no. 4, pp. 28–38, Jan.–Mar. 2011.
[108] A. G. Perez et al., "Soft finger tactile rendering for wearable haptics," in *Proc. IEEE World Haptics Conf.*, 2015, pp. 327–332.
[109] A. G. Perez, G. Cirio, F. Hernandez, C. Garre, and M. A. Otaduy, "Strain limiting for soft finger contact simulation," in *Proc. IEEE World Haptics Conf.*, 2013, pp. 79–84.
[110] K. Hirota, Y. Ujitoko, S. Sakurai, and T. Nojima, "Deformation matching: Force computation based on deformation optimization," *IEEE Trans. Haptics*, vol. 15, no. 2, pp. 267–279, Apr.–Jun. 2022.
[111] S. Yoshimura, R. Shioya, H. Noguchi, and T. Miyamura, "Advanced general-purpose computational mechanics system for large-scale analysis and design," *J. Comput. Appl. Math.*, vol. 149, no. 1, pp. 279–296, 2002.
[112] N. Mitsuhashi, K. Fujieda, T. Tamura, S. Kawamoto, T. Takagi, and K. Okubo, "Bodyparts3d: 3D structure database for anatomical concepts," *Nucleic Acids Res.*, vol. 37, no. suppl_1, pp. D782–D785, 2009.
[113] Y. Ujitoko, T. Taniguchi, S. Sakurai, and K. Hirota, "Development of finger-mounted high-density pin-array haptic display," *IEEE Access*, vol. 8, pp. 145107–145114, 2020.
[114] T. Iwamoto, M. Tatezono, and H. Shinoda, "Non-contact method for producing tactile sensation using airborne ultrasound," in *Proc. Int. Conf. Hum. Haptic Sens. Touch Enabled Comput. Appl.*, 2008, pp. 504–513.
[115] T. Howard, M. Marchal, and C. Pacchierotti, "Ultrasound mid-air tactile feedback for immersive virtual reality interaction," in *Ultrasound Mid-Air Haptics for Touchless Interfaces*. Berlin, Germany: Springer, 2022, pp. 147–183.
[116] B. Long, S. A. Seah, T. Carter, and S. Subramanian, "Rendering volumetric haptic shapes in mid-air using ultrasound," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 1–10, 2014.
[117] G. Korres and M. Eid, "Haptogram: Ultrasonic point-cloud tactile stimulation," *IEEE Access*, vol. 4, pp. 7758–7769, 2016.
[118] B. Kappus and B. Long, "Spatiotemporal modulation for mid-air haptic feedback from an ultrasonic phased array," *J. Acoustical Soc. Amer.*, vol. 143, no. 3, pp. 1836–1836, 2018.
[119] J. Jang and J. Park, "SPH fluid tactile rendering for ultrasonicmid-air haptics," *IEEE Trans. Haptics*, vol. 13, no. 1, pp. 116–122, Jan.–Mar. 2020.
[120] H. Barreiro, S. Sinclair, and M. A. Otaduy, "Path routing optimization for STM ultrasound rendering," *IEEE Trans. Haptics*, vol. 13, no. 1, pp. 45–51, Jan.–Mar. 2020.
[121] A. Matsubayashi, Y. Makino, and H. Shinoda, "Direct finger manipulation of 3D object image with ultrasound haptic feedback," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–11.
[122] A. Matsubayashi, H. Oikawa, S. Mizutani, Y. Makino, and H. Shinoda, "Display of haptic shape using ultrasound pressure distribution forming cross-sectional shape," in *Proc. IEEE World Haptics Conf.*, 2019, pp. 419–424.
[123] A. Matsubayashi, Y. Makino, and H. Shinoda, "Rendering ultrasound pressure distribution on hand surface in real-time," in *Proc. Haptics: Sci., Technol., Appl.: 12th Int. Conf.*, 2020, pp. 407–415.
[124] S. Inoue, Y. Makino, and H. Shinoda, "Scalable architecture for airborne ultrasound tactile display," in *Haptic Interaction: Science, Engineering and Design 2*. Berlin, Germany: Springer, 2018, pp. 99–103.
[125] A. Matsubayashi, Y. Makino, and H. Shinoda, "Accurate control of sound field amplitude for ultrasound haptic rendering using the levenberg-marquardt method," in *Proc. IEEE Haptics Symp.*, 2022, pp. 1–6.
[126] D. M. Plasencia, R. Hirayama, R. Montano-Murillo, and S. Subramanian, "GS-PAT: High-speed multi-point sound-fields for phased arrays of transducers," *ACM Trans. Graph.*, vol. 39, no. 4, pp. 138–1, 2020.
[127] L. Mulot et al., "Dolphin: A framework for the design and perceptual evaluation of ultrasound mid-air haptic stimuli," in *Proc. ACM Symp. Appl. Percep.*, 2021, pp. 1–10.
[128] I. Rakkolainen, E. Freeman, A. Sand, R. Raisamo, and S. Brewster, "A survey of mid-air ultrasound haptics and its applications," *IEEE Trans. Haptics*, vol. 14, no. 1, pp. 2–19, Jan.–Mar. 2021.
[129] M. N. Perquin, M. Taylor, J. Lorusso, and J. Kolasinski, "Directional biases in whole hand motion perception revealed by mid-air tactile stimulation," *Cortex*, vol. 142, pp. 221–236, 2021.
[130] T. Howard, K. Driller, W. Frier, C. Pacchierotti, M. Marchal, and J. Hartcher-O'Brien, "Gap detection in pairs of ultrasound mid-air vibrotactile stimuli," *ACM Trans. Appl. Perceptions*, vol. 20, no. 1, pp. 1–17, 2023.
[131] J. Barbič, F. Sin, and E. Grinspun, "Interactive editing of deformable simulations," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–8, 2012.
[132] J. Tapia, C. Romero, J. Pérez, and M. A. Otaduy, "Parametric skeletons with reduced soft-tissue deformations," in *Computer Graphics Forum*, vol. 40, no. 6. Hoboken, NJ, USA: Wiley, 2021, pp. 34–46.
[133] J. P. Fritz, T. P. Way, and K. E. Barner, "Haptic representation of scientific data for visually impaired or blind persons," in *Proc. CSUN Conf. Technol. Disabil.*, 1996. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=dbb21d64d3bd2f640d99ee5045632fc55d0cd19d
[134] T. Cui, J. Xiao, and A. Song, "Simulation of grasping deformable objects with a virtual human hand," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 3965–3970.
[135] T. Cui, A. Song, and J. Xiao, "Modeling global deformation using circular beams for haptic interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 1743–1748.
[136] S. Yim, S. Jeon, and S. Choi, "Data-driven haptic modeling and rendering of viscoelastic and frictional responses of deformable objects," *IEEE Trans. Haptics*, vol. 9, no. 4, pp. 548–559, Oct.–Dec. 2016.
[137] A. Sianov and M. Harders, "Exploring feature-based learning for data-driven haptic rendering," *IEEE Trans. Haptics*, vol. 11, no. 3, pp. 388–399, Jul.–Sep. 2018.
[138] T. Tokuyasu, E. Maeda, T. Okamoto, K. Toshimitsu, K. Okamura, and K. Yoshiura, "Construction of virtual patient model for maxillofacial palpation training system," *Artif. Life Robot.*, vol. 19, no. 2, pp. 176–180, 2014.
[139] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Voxel-based 6-DoF haptic rendering improvements," *Haptics*, vol. 3, no. 7, pp. 1–12, 2006.

[140] C. Duriez, C. Andriot, and A. Kheddar, "Signorini's contact model for deformable objects in haptic simulations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, pp. 3232–3237.
[141] J. R. Flanagan, M. K. Burstedt, and R. S. Johansson, "Control of fingertip forces in multidigit manipulation," *J. Neuriophysiol.*, vol. 81, no. 4, pp. 1706–1717, 1999.
[142] A. Naceri, A. Moscatelli, M. Santello, and M. O. Ernst, "Coordination of multi-digit positions and forces during unconstrained grasping in response to object perturbations," in *Proc. IEEE Haptics Symp.*, 2014, pp. 35–40.
[143] T. Feix, I. M. Bullock, Y. Gloumakov, and A. M. Dollar, "Effect of number of digits on human precision manipulation workspaces," *IEEE Trans. Haptics*, vol. 14, no. 1, pp. 68–82, Jan.–Mar. 2021.
[144] D. Gueorguiev, D. Tzionas, C. Pacchierotti, M. J. Black, and K. J. Kuchenbecker, "Statistical modelling of fingertip deformations and contact forces during tactile interaction," in *Proc. Extended Abstr. Presented Hand, Brain Technol. Conf.*, 2018. [Online]. Available: https://ps.is.mpg.de/publications/gueorguiev18-hbtea-modelling
[145] T. Kassuba, C. Klinge, C. Hölig, B. Röder, and H. R. Siebner, "Vision holds a greater share in visuo-haptic object recognition than touch," *Neuroimage*, vol. 65, pp. 59–68, 2013.
[146] J. M. Hillis, M. O. Ernst, M. S. Banks, and M. S. Landy, "Combining sensory information: Mandatory fusion within, but not between, senses," *Science*, vol. 298, no. 5598, pp. 1627–1630, 2002.
[147] M. Mori, K. F. MacDorman, and N. Kageki, "The uncanny valley [from the field]," *IEEE Robot. Automat. Mag.*, vol. 19, no. 2, pp. 98–100, Jun. 2012.
[148] C. C. Berger, M. Gonzalez-Franco, E. Ofek, and K. Hinckley, "The uncanny valley of haptics," *Sci. Robot.*, vol. 3, no. 17, 2018, Art. no. eaar7010.
[149] L. Gehrke et al., "Detecting visuo-haptic mismatches in virtual reality using the prediction error negativity of event-related brain potentials," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–11.
[150] M. D'Alonzo, A. Mioli, D. Formica, L. Vollero, and G. Di Pino, "Different level of virtualization of sight and touch produces the uncanny valley of avatarhand embodiment," *Sci. Rep.*, vol. 9, no. 1, pp. 1–11, 2019.
[151] M. Turk, "Multimodal interaction: A review," *Pattern Recognit. Lett.*, vol. 36, pp. 189–195, 2014.
[152] S. M. Nizam, R. Z. Abidin, N. C. Hashim, M. C. Lam, H. Arshad, and N. Majid, "A review of multimodal interaction technique in augmented reality environment," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, no. 4–2, 2018, Art. no. 1460.

Qianqian Tong (Member, IEEE) received the Ph.D. degree in computer application technology from 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.

Wenxuan Wei is currently working toward the M.S. degree in mechanical engineering, 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 technical 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, Ann Arbor, MI, USA. She is the Deans' Excellence Professor, William B. Smith Distinguished Fellow of robotics engineering, a Professor of computer science, and the Head of the Robotics Engineering Department, Worcester Polytechnic Institute, Worcester, MA, USA. She joined WPI from the University of North Carolina at Charlotte, Charlotte, 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, 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. Since 2015, he has been an Associate Editor for IEEE TRANSACTIONS ON HAPTICS. From 2014 to 2017, he was the Chair of Executive Committee of the IEEE Technical Committee on Haptics (IEEE TCH).