

HapticBots: Distributed Encountered-type Haptics for VR with Multiple Shape-changing Mobile Robots

Ryo Suzuki
University of Calgary
Calgary, AB, Canada
ryo.suzuki@ucalgary.ca

Eyal Ofek
Microsoft Research
Redmond, WA, USA
eyalofek@microsoft.com

Mike Sinclair
Microsoft Research
Redmond, WA, USA
sinclair@microsoft.com

Daniel Leithinger
University of Colorado Boulder
Boulder, CO, USA
daniel.leithinger@colorado.edu

Mar Gonzalez-Franco
Microsoft Research
Redmond, WA, USA
margon@microsoft.com

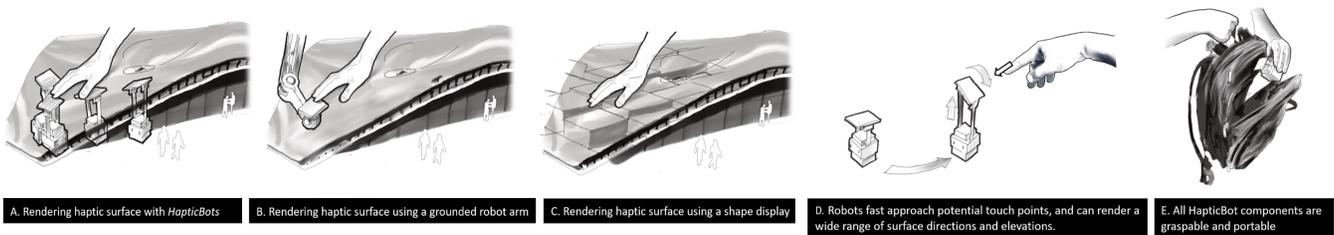


Figure 1: (a) *HapticBots* enable to support multiple concurrent touch points of a virtual surface, giving the illusion of a large physical object. An existing approach such as using a grounded robot arm (b) to bring a touchable surface piece to encounter the user hand [3, 55] is limited to render a single touch point. (c) shape displays [13, 34] may render the entire shape simultaneously but are limited in fidelity and the area they can cover (display geometry resolution is displayed coarser for visualization). (d) The *HapticBots* are designed to coordinate and move fast to encounter the hand when it reaches possible touch points and render 5 degrees of freedom (3D position & 2 angles of surface normal). (e) all the system's components are light, portable, and easy to deploy.

ABSTRACT

HapticBots introduces a novel encountered-type haptic approach for Virtual Reality (VR) based on multiple tabletop-size shape-changing robots. These robots move on a tabletop and change their height and orientation to haptically render various surfaces and objects on-demand. Compared to previous encountered-type haptic approaches like shape displays or robotic arms, our proposed approach has an advantage in deployability, scalability, and generalizability—these robots can be easily deployed due to their compact form factor. They can support multiple concurrent touch points in a large area thanks to the distributed nature of the robots. We propose and evaluate a novel set of interactions enabled by these robots which include: 1) rendering haptics for VR objects by providing just-in-time touch-points on the user's hand, 2) simulating continuous surfaces with the concurrent height and position change, and 3)

enabling the user to pick up and move VR objects through graspable proxy objects. Finally, we demonstrate *HapticBots* with various applications, including remote collaboration, education and training, design and 3D modeling, and gaming and entertainment.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; **Haptic devices**.

KEYWORDS

virtual reality; encountered-type haptics; tabletop mobile robots; swarm user interfaces

ACM Reference Format:

Ryo Suzuki, Eyal Ofek, Mike Sinclair, Daniel Leithinger, and Mar Gonzalez-Franco. 2021. *HapticBots: Distributed Encountered-type Haptics for VR with Multiple Shape-changing Mobile Robots*. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*, October 10–14, 2021, Virtual Event, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3472749.3474821>

1 INTRODUCTION

Effective haptic feedback promises to enrich Virtual Reality (VR) experiences in many application domains [17], but supporting general-purpose haptic feedback is still a difficult challenge. A common

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
UIST '21, October 10–14, 2021, Virtual Event, USA

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8635-7/21/10...\$15.00
<https://doi.org/10.1145/3472749.3474821>

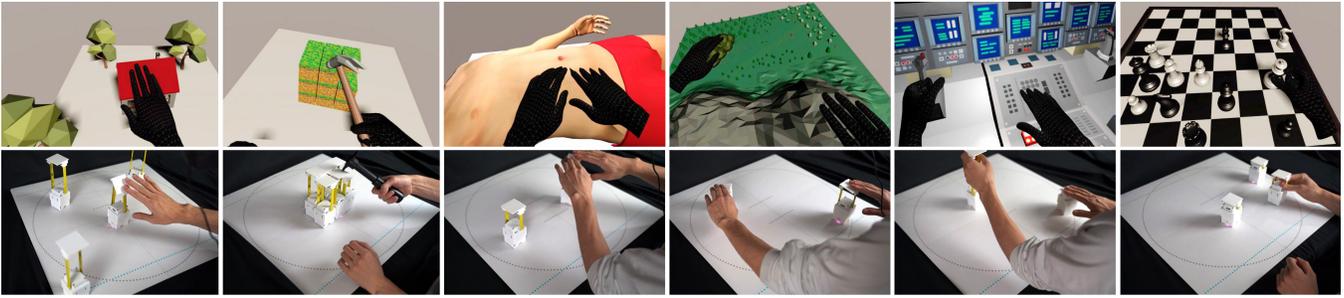


Figure 2: HapticBots render encountered-type haptics for a range of VR applications.

approach to providing haptic feedback for VR is to use a hand-held or wearable device [9, 10, 19, 30, 32]. However, these wearable hand-grounded devices are inherently limited in their ability to render a world grounded force, such as surfaces that can be touched or pushed with the user’s hand.

To fill this gap, encountered-type haptics [36, 59] are introduced as an alternative approach. In contrast to hand-held or wearable devices, the encountered-type haptics provide haptic sensations through *actuated physical environments* by dynamically moving physical objects [3, 18, 46] or transforming the physical shape [37, 43] when the user encounters the virtual object.

Different approaches have been developed for encountered-type haptics: from grounded robotic arms (e.g., Snake Charmer [3], VR-Robot [55]) to shape displays (e.g., shapeShift [43], Feelex [25], inForce [37]). However, the current approaches still face a number of challenges and limitations. For example, shape displays (Figure 1 c) often require large, heavy, and mechanically complex devices, reducing reliability and deployability of the system for use outside research labs. Also, the resolution fidelity and the display’s size are still limited, making it difficult to render smooth and continuous surfaces across a large interaction area. Alternately, robotic arms (Figure 1 b) can bring a small piece of a surface to meet the user hand on demand, but the speed at which humans move challenges the ability to cover just in time large interaction spaces with a single device. Scaling the number of robotic arms is also a challenge as complex 3D path planning is required to avoid unnecessary collision with both the user and the other arms.

The goal of this paper is to address these challenges by introducing a novel encountered-type haptics approach, which we call *distributed encountered-type haptics* (Figure 1). Distributed encountered-type haptics employ multiple shape-changing mobile robots to simulate a consistent physical object that the user can encounter through hands or fingers. By synchronously controlling multiple robots, these robots can approximate different objects and surfaces distributed in a large interaction area.

Our proposed approach enables *deployable*, *scalable*, and *general-purpose* encountered-type haptics for VR, providing a number of advantages compared to the existing approaches, including shape displays [1, 25, 37, 43], robotic arms [3, 55], and non-transformable mobile robots [14, 18]. **1) Deployable:** Each mobile robot is light and compact, making the system portable and easy to deploy (Figure 1 e). **2) Scalable:** Since each robot is simple and modular, it can scale to increase the number of touch-points and covered area.

Moreover, the use of multiple robots can reduce the average distance that a robot needs to travel, which reduces the robots’ speed requirements. **3) General-purpose:** Finally, the shape-changing capability of each robot can significantly increase the expressiveness of haptic rendering by transforming itself to closely match with the virtual object on-demand and in real-time. This allows for greater flexibility needed for general-purpose applications.

The main contribution of this paper is a concept of this novel encountered haptic approach and a set of distributed interaction techniques, outlined in Section 3 (e.g., Figure 3-4). To demonstrate this idea, we built *HapticBots*, an open source ¹ tabletop shape-changing mobile robots that are specifically designed for distributed encountered-type haptics. HapticBots consists of off-the-shelf mobile robots (Sony TOIO), and custom height-changing mechanisms to haptically render general large surfaces with varying normal directions (-60 to 60 degrees). It can cover a large space (55 cm × 55 cm) above the table (a dynamic range of 24 cm elevation) at high speed (24 cm/sec and 2.8 cm/sec for horizontal and vertical speed, respectively). Each robot is compact (4.7 × 4.7 × 8 cm, 135 g) and its tracking system consists of an expandable, pattern-printed paper mat; thus, it is portable and deployable.

Our HapticBots’ hardware design is inspired by ShapeBots [50], but as far as we know, our system is the first exploration of using multiple tabletop shape-changing robots for VR haptics. Applying to VR haptics introduces a set of challenging requirements, which led to a new distributed haptics system design as well as to new hardware for each of the robots: **1) Efficient path planning integrated with real-time hand tracking:** The system coordinates the movements of all robots with the user’s hand. We track and anticipate potential touch points at a high frame rate (60 FPS) and guide the robots to encounter the user’s hands in a just in time fashion. **2) Precise height and tilt control:** In contrast to ShapeBots’ open-loop system, HapticBots enables more precise height and tilt control with embedded encoders and closed-loop control system to render surfaces with varying normal angles. **3) Actuator robustness:** We vastly improved actuator force by around 70x (21.8N vs. 0.3N holding force of [50]) to provide meaningful force feedback. In addition to these technical contributions, we developed various VR applications to demonstrate the new possibilities for encountered haptics, including remote collaboration, medical training, 3D modeling, and entertainment.

¹<https://github.com/ryosuzuki/hapticbots>

To evaluate how HapticBots can create plausible haptic sensations, we conducted two user studies. In the first task, participants were asked to encounter and explore the surfaces of different virtual objects and report the encounters' level of realism. In the second task, we measured the capabilities of HapticBots to deliver continuous touch sensation when exploring surfaces, and the level of granularity that it can provide by asking participants to distinguish among surfaces of different tilt angles. We further compared the haptic sensations between HapticBots and a static shape display. Our study validates the effectiveness of the proposed approach.

Finally, the contributions of this paper are:

- (1) A novel concept of distributed encountered-type haptics
- (2) A hardware and software implementation of HapticBots and its applications
- (3) User evaluations that investigate the effects of HapticBots haptic illusions.

2 RELATED WORK

2.1 Haptic Interfaces for VR

2.1.1 Hand-held and Wearable Haptic Interfaces. In recent years, various haptic devices have been explored to enhance the user immersion of VR. One of the most common types of haptic devices is the hand-held or wearable haptic approach. Most hand-held haptic devices render touch sensations of virtual objects by applying differential forces of the fingertips against the opposing or thumb [6, 30, 45]. For realizing the dynamic range of the device movement, the fingertip is usually pushed back to stay outside the virtual object until interaction [9, 10, 42]. Only a few devices, such as Haptic Revolver [57], can also render forces such as texture and cutaneous shear force of the virtual surface. However, one inherent limitation of such *body grounded* devices is the lack of generating a convincing *world grounded* sensation as no perceived force stops the body from moving into virtual objects. Grounding the hands to the body using exoskeletons or strings [12] can aid the grounding perception but are cumbersome and complex.

2.1.2 Passive Haptics. Alternatively, *passive haptics* [23] approach uses a physical artifacts such as a haptic proxy for VR, so that a VR user can touch and interact with a real object. For example, Annexing Reality [20] employs static physical objects as props in an immersive environment by matching and adjusting the shape and size of the virtual objects. Haptic Retargeting [5, 8] leverages a mismatch in hand-eye coordination in order to guide the user's touch toward the position of physical objects. Similarly, by combining passive objects with redirected walking [40], Kohli et al. [29] explored haptics that can go beyond the scale of human hands. Using passive objects, one can generate very reasonable haptic sensations. However, as the shape and position of the proxy object in this case is fixed, it has a limited degree of haptic expression. For example, when the position or geometry of the proxy object differs from the displayed virtual object, it can break the illusion [41]. Manual reconfiguration of proxy objects has been also explored [4, 61], but lacks the capability of dynamically simulating various shapes on demand.

2.1.3 Robotic Encountered-type Haptics. To overcome this limitation, McNeely proposed *robotic encountered-type haptics* by integrating a passive haptic proxy with mechanical actuation [36]. Encountered-type haptics dynamically positions or transform the haptic props when the user "encounters" the virtual object. Overall, there are three different approaches that have been explored for tabletop encountered-type haptics: robotic arms, shape displays, and mobile robots.

Robotic arms [21], such as SnakeCharmer [3] and VRRobot [55] simulate surfaces by bringing a small patch of a surface to the user's hand wherever she may touch the virtual surface. Since the virtual object is invisible to the VR user, it can potentially generate the perception that the entire geometry exists in the physical world. However, the need for a robot arm to cover a large interaction space requires a large arm with a long reach which may be heavy and less portable. Also, the requirement for moving the large robotic arm in a volume while the user is blind to it, may limit the speed or movement space of the robot for safety reasons.

The second approach is shape displays [13, 34, 44]. Systems like Feelex [25], inFORCE [37], and shapeShift [1, 43] simulate dynamic surfaces and shapes by constructing the encountered geometry using an array of actuated pins. However, the large number of actuators that are needed to render a shape limits these devices' resolution and makes them complex, expensive, heavy, power hungry, and limited in coverage area.

The third approach uses mobile robots [14, 18, 24, 46, 56, 58] or even drones [2, 22] to move or actuate passive proxy objects. For example, PhyShare [18] and REACH++ [14] employ the tabletop mobile robots to dynamically reposition the attached passive haptic proxy. However, these mobile robots can only render a single predefined object due to the lack of transformation capability. Zhao et al. [60] explored assembled haptic proxies of objects using swarm robots. While they assemble the required geometry on demand, it requires significant time to assemble a large object, which limits real-time interaction.

As we can see, the existing encountered-type approaches still have many challenges in terms of deployability (portable and deployable form factor), scalability (both an interaction area and the number of touch-points), and generalizability (the variety of shapes and surfaces the system can support). Our contribution is to address these problems by introducing a new class of encountered-type haptics with distributed shape-changing robots.

2.2 Swarm User Interfaces

Our work is also built on recent advances in swarm user interfaces, which leverage a swarm of robots for tangible and shape-changing interactions (e.g., Zooids [31], UbiSwarm [27], HERMITS [38], ShapeBots [50], Reactile [47], PICO [39]). Some of the previous systems have demonstrated the haptic and tactile sensation with swarm interfaces. For example, SwarmHaptics [28] demonstrate the use of swarm robots for everyday, non VR, haptic interactions (e.g., notification, communication, force feedback) and RoboGraphics [15] and FluxMarker [49] explores the tactile sensation for people with visual impairments. More recently, several works have been introduced to augment the capability of each robot to enhance interactions.

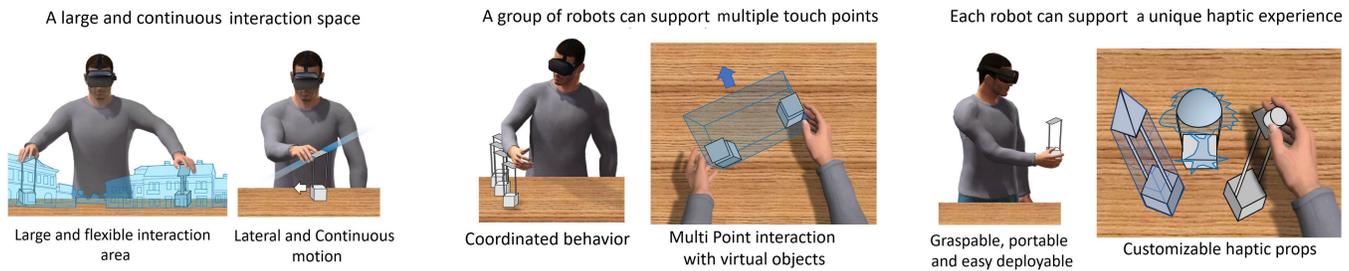


Figure 3: Unique features of distributed encountered-type haptics. First, our freely moving robots can reach and render touch points in a large interaction space. Accurate rendering of 3D locations and orientations of surfaces (left). The use of multiple robots can generate multiple touch events concurrently (middle). Finally, each robot can be regarded as a separate object that may be picked up, moved around, and the payload that it carries can be uniquely fitted to the application

For example, HERMITS [38] augment the robots with customizable mechanical add-ons to expand tangible interactions.

Particularly, our work is inspired by the idea of “*shape-changing swarm robots*” introduced by ShapeBots [50]. ShapeBots demonstrates the idea of combining a small table-top robot with a miniature reel actuator to greatly enhance the range of interactions and expressions for tangible user interfaces.

However, none of these works are aimed at rendering haptics of general large geometries in VR. As far as we know, our system is the first exploration of using tabletop-size shape-changing swarm robots for VR haptics. Applying swarm UIs to VR haptics introduces a set of challenges and opportunities. For example, the prior work [50] explicitly articulated that support for AR/VR haptics is their limitation and future work due to a number of technical challenges, including the robustness of the actuation and VR software integration. On the other hand, in VR, the user is blind to the real world, thus it is possible to render larger geometries with only a smaller number of just in time robots. Our goals are to explore this previously unexplored design space, introduce a set of haptic interactions, and address these software and hardware challenges for VR haptics applications.

3 DISTRIBUTED ENCOUNTERED-TYPE HAPTICS

3.1 Concept and Unique Features

This section introduces a novel encountered-type haptic approach, which we call *distributed encountered-type haptics*. Distributed encountered type haptics employ multiple coordinated robots that can move their position and transform their shape to haptically render various objects and surfaces distributed in the space. Our approach has the following unique features: 1) support for large and flexible interaction areas, and 2) portable and deployable form factor.

3.1.1 Large and Flexible Interaction Area. One of the unique advantages of our approach is the ability for distributed and fast moving robots to cover a large and flexible interaction space (Figure 3) and leverage two-handed interactions. Since each robot is simple and modular, it is easy to scale the number of touch points and covered area.

3.1.2 Portable and Deployable Form Factor. Distributed robots are composed of compact and small components, and are not bound to preset locations. Our implementation particularly leverages recent advantages of tracking systems in both the VR headset and the robot’s location. For example, HapticBots uses a lightweight mat, printed with a dot pattern viewed by the robots, as a tracking mechanism. Since the setup of this tracking mechanism is fairly simple (only placing a mat), the system can work in any flat location without dedicated external tracking devices (e.g., a transparent desk and camera in [50] or optical tracking systems in [14, 18, 46]). Also, a standalone Oculus Quest HMD enables inside-out hand tracking, where the system does not require any calibrated setup. Since each robot is compact, it is possible to put the entire system in a small carrying case and quickly deploy it to another horizontal surface.

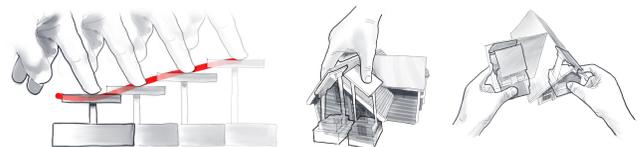


Figure 4: Left: HapticBots supports lateral movements of the touch surface (shown as a red curve). The robot may also tilt its surface to better reflect the surface normal. Right: Coordinated behaviors with multiple robots. Multiple touch points can enable the user to move, rotate or scale virtual objects.

3.2 Haptic Interactions and Unique Affordances

In addition to these unique features, our approach enables the following haptic interactions and unique affordances.

3.2.1 Rendering Continuous Surfaces with Concurrent Lateral Motion. Pin-based shape displays approximate a shape of a surface by a set of discreet pins, which generates spatial aliasing when rendering diagonal or curved objects [1]. Particularly when sliding ones fingers over a non-horizontal surface of the shape display, the abrupt shape change of the individual pins are very noticeable. On

the other hand, our approach can closely render the continuous smooth surface by leveraging the concurrent lateral motion. By using real-time tracking of the user’s hands to guide the robots, it helps enable constant contact with the user’s fingers. Robots can follow the movement of the user’s finger and move or rotate in azimuth with the hand while changing its surface height and orientation to follow the virtual geometry (Figure 4).

3.2.2 Rendering Large Objects with Coordinated Multiple Robots. The coordination of multiple robots can extend their rendering capabilities. For example, Figure 4 illustrates an example of multi-point interaction with coordinated robots to simulate the haptic sensation of an object that is much larger than each robot. A group of robots can mimic a small shape display with additional degrees of freedom such as tilting and relative rotations. For example, two tilted robots can work in tandem to simulate a corner between two surfaces, such as the tip of a roof of a virtual house (Figure 4)

3.2.3 Rendering a Large Number of Objects with a Smaller Number of Robots. These robots can also give the illusion that the user can touch more objects than the actual number of robots. Even with a smaller number of robots, by leveraging the locomotion capability and anticipation of hand movement, the robots can position themselves to the object which the user will most likely encounter in the next moment. With that, the user may perceive that the entire scene is haptically rendered with these robots.

3.2.4 Rendering Graspable Objects for Tangible Interaction. Another unique affordance of the mobile robot is its graspability. (Figure 3). A compact robot can be picked and moved to another location by the user. This ability opens new interaction possibilities such as positioning robots as input (e.g. as proxies to virtual objects).

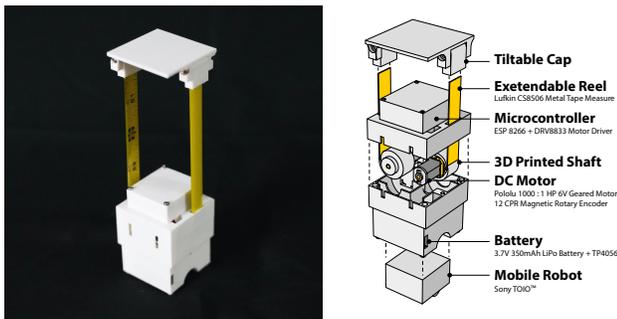


Figure 5: Mechanical design of the reel actuator.

4 HAPTICBOTS: SYSTEM AND IMPLEMENTATION

To demonstrate our concept, we built HapticBots, a system that consists of multiple coordinated height-changing robots and the associated VR software. Each robot is made of 1) custom-built shape-changing mechanisms with reel-based actuators, and 2) an off-the-shelf mobile robot (Sony Toio) that can move on a mat printed with a pattern for position tracking. For the VR system, we used Oculus Quest HMD and its hand tracking capability for interaction.

The software system synchronizes virtual scenes with physical environment (e.g., each robot’s position, orientation, and height), so that the robots can provide a haptic sensation in a timely manner. This section describes the design and implementation of the both hardware and software systems, then provides technical evaluation of HapticBots prototype.

4.1 Reel-based Linear Actuator

4.1.1 Mechanical Design. To enable a large dynamic range of heights with a compact form factor, HapticBots employs an extendable reel-based linear actuator, inspired by prior works (e.g., ShapeBots [50], KineReels [51], and G-Raff [26]). However, we need to accommodate for: 1) mechanical stability of the actuator, essential to provide meaningful force feedback, 2) compact form factor, and 3) fast transformation speed for real-time interactions. For example, the vertical load-bearing capability of ShapeBots is only 0.3N with extended states. Also, the linear actuator can easily buckle with hand pressure [50].



Figure 6: The actuator maximum holding load is 1379 g at the extended state (25 cm height)

Our HapticBots linear actuator is designed to achieve all of these requirements with reasonable force capabilities. Figure 5 illustrates the mechanical design of a linear actuator. In our design, the two retractable metal tapes on motorized reels occupy a small footprint but extend and hold their shape while resisting modest loads in certain directions. Our reel-based linear actuator uses compact DC motors (Pololu 1000:1 Micro Metal Gearmotor HP 6V, Product No. 2373). This motor has a cross-section of 1.0×1.2 cm and its length is 2.9 cm. The no-load speed of the geared motor is 31 rpm, which extends the metal tape at 2.8 cm/sec. The motor’s maximum stall torque is 12 kg·cm. We accommodate two motors placed side by side to minimize the overall footprint size.

For the reel, we use an off-the-shelf metal tape measure reel (Crescent Lufkin CS8506 1/2 x 6 inch metal tape measure). The material choice of this reel is one of the key design considerations as it determines the vertical load-bearing capability. On the other hand, a strong material makes it more difficult for this small DC motor to successfully rotate and rotate the reel. After the test of eight different tape measure devices with various materials, stiffnesses, thicknesses, and widths, we determined the Crescent Lufkin CS8506 tape measure to work most reliably in our setting. The tape has 0.15 mm thickness and is 1.2 cm (1/2 inch) width wide, and slightly curved to avoid buckling. We cut this tape measure to 36 cm and drilled a 3 mm hole at the end to fix it to the shaft with an M3 screw.

Two DC motors + gears individually rotate to extend and retract the reels. Each reel is connected to a tilttable planar cap. The cover cap is made of 3D printed parts (4.7×4.7 cm) and has a shaft on each

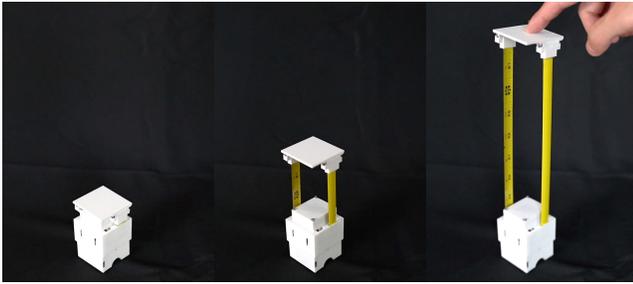


Figure 7: The reel actuator extends the tape at 2.8 cm/sec from a minimum height of 8cm to a maximum height of 32 cm.

side fastened with an M3 screw (2.6 cm in length) and nut to make each end rotatable. By individually controlling the extension length of each tape, the top surface can tilt between -60 to 60 degrees. We use a rotary encoder (Pololu Magnetic Encoder Pair Kit, 12 CPR, 2.7-18V, Product No. 4761) connected to the motor shaft to continuously measure the position of each reel and hence extension of the tape and tilt of the cap.

The overall footprint of our actuator has a cross-section of 4.7×4.7 cm and 3.0 cm in height. The robot's height can change from 8 cm in minimum to 32 cm at maximum. The no-load extension/retraction speed is 2.8 cm/sec. The vertical load-bearing capability is approximately 13.53 N (at the extended state), which is strong enough to withstand a modest human touch force (describe more detail in the technical evaluation section).



Figure 8: The cap can tilt up to 60 degrees.

4.1.2 Design Rationale. During our initial prototyping phase, we have considered and prototyped various different mechanisms, including lead screws (similar to shapeShift [43]), telescopic pole (similar to motorized telescopic car antenna), and inflatable structure (similar to TilePoP [53], LiftTiles [48], PuPoP [52], and Pneumatic Reel Actuator [16]). These designs are suitable for haptic devices as they can withstand the strong vertical force (e.g., pushing force with a hand), but they often require a longer vertical travel and lower extension ratio (e.g., lead screw, telescopic pole) or slow transformation speed and complex control system (e.g., pneumatic actuation). Therefore, we decided to explore and develop the current design to meet our design requirements.

4.1.3 Electronic Design. Figure 9 illustrates the schematic of HapticBots' electronic components. We use an ESP8266 microcontroller (Wemos D1 mini, 3.4×2.5 cm) to control two motors, read signals from two rotary encoders, and communicate with the main computer through Wi-Fi communication with a user datagram protocol (UDP). Each module connects to a private network and is assigned a unique IP address. The computer sends a target height to each IP address, and the microcontroller controls the rotation of the motor by measuring the rotation count based on the rotary encoder. The microcontroller controls one dual motor driver (Pololu DRV8833 Dual Motor Driver Product No. 2130), which can control two DC motors independently. The operating voltage of all modules is 3.5V and the microcontroller is connected to 3.7V LiPo battery (350mAh 652030) through a recharging module (TP4056). The module is rechargeable through an external micro-USB cable.

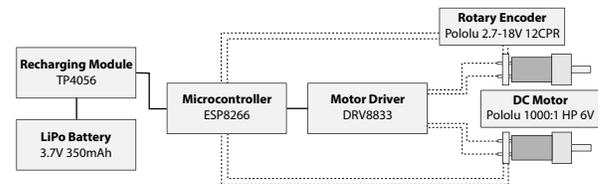


Figure 9: System schematics of the actuator electronics.

The entire linear actuator component measures 4.7×4.7 cm \times 8.0 cm. Our prototype costs are approximately \$70 USD for each actuator, including the microcontroller (\$3 USD), motors (\$20 USD \times 2), a motor driver (\$4 USD), rotary encoders (\$7 USD \times 2), a battery (\$6 USD), a custom PCB (\$1 USD), and a tape measure (\$6 USD), but the cost could be significantly decreased with mass production and using alternative low-cost motors.

4.2 Mobile Robot Base

As a mobile robotic base, we use an off-the-shelf mobile robot (Sony ToioTM) that features two-wheeled robots and Bluetooth control. The main reason we chose the Toio robot is its sophisticated and easily deployable tracking system. In addition, Toio robots have numerous advantages in 1) off-the shelf availability, 2) light and compact, 3) fast, 4) fairly strong, and 5) safe. For tracking and localization, Toio has a built-in look-down camera at the base of the robot to track the position and orientation on a mat by identifying unique printed dot patterns, similar to the Anoto marker [11]. The built-in camera reads and identifies the current position of the robot, enabling easy 2D tracking of the robots with no external hardware (Figure 10).

We use a publicly available JavaScript API to programmatically track and control the robots from a computer². The size of each robot has a cross-section of 3.2×3.2 cm and 2.5 cm in its height. The tracking mat (Toio Tracking MatTM) has 55×55 cm of covered area, but the interaction area can be easily extended with multiple mats.

²<https://github.com/toio/toio.js>

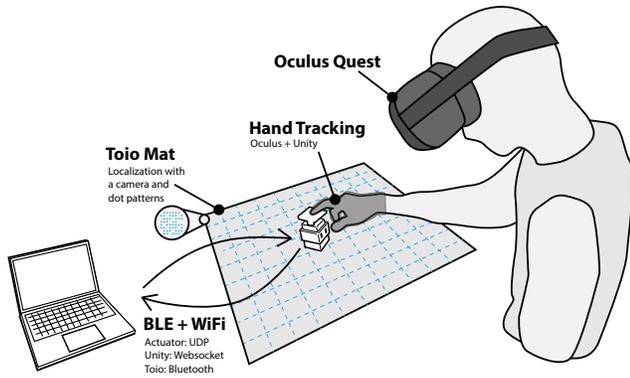


Figure 10: System Setup: The Oculus Quest VR Headset is calibrated to work in the same coordinate space as the tracked robots.

The resolution of 2D position and orientation detection are 1.42 mm and 1 degree, respectively. The Toio robot alone (without the shape changing module) can drive and rotate with the maximum speed of 35 cm/sec and 1500 degrees/sec, respectively. The maximum load-bearing capacity of Toio robot is 200g, but can move heavier objects with an appropriate caster base that can evenly distribute the weight. Each robot costs approximately \$40 USD.

4.2.1 Path Planning and Control Mechanism. Based on Toio’s internal tracking mechanism, the software reads and controls the position and orientation of the multiple robots simultaneously (see Figure 11). To achieve this, we adapt the Munkres algorithm for target assignment and Reciprocal Velocity Obstacles (RVO) algorithm [54] for collision avoidance.

Given the current position of each robot and its target position, the algorithm first assigns each target by minimizing the total travel distance of all robots. Once a target is assigned, the system navigates the robot until all of the robots reach to their target positions, while avoiding the collisions with RVO. The driving speed dynamically changes based on the distance between the current and target position, to maximize speed when the target is far and slows it down when approaching to the target to reduce the braking distance and avoid overshooting. The distance to stop a moving robot is 2 mm and 5 degrees in orientation.

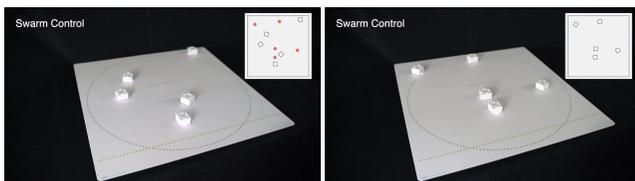


Figure 11: Our path planning and control algorithm coordinates multiple robots.

4.3 Software Implementation

4.3.1 Software Architecture. For the virtual reality environment and gestural tracking, we use an Oculus Quest HMD with its built-in hand tracking mechanism (see Figure 10). The main computer (MacBook Pro 16 inch, Core i9, 32GB RAM) runs as the Node.js server. The server controls and communicates with the linear actuators and the Unity application on an Oculus Quest through Wi-Fi (with UDP and WebSocket protocol, respectively) and Toio robots through Bluetooth. The host computer communicates with seven Toio robots through Bluetooth V4.2 (Bluetooth Low Energy). The HapticBots computer operates at 60 FPS for tracking and control.

4.3.2 Approximating a Virtual Surface in Unity. We use the Unity game engine to render a virtual environment. As each robot moves along the planar surface, it constantly changes its height and orientation to best fit the virtual surface above it. To obtain the target height and surface normal of the robot, the system uses a vertical ray casting to measure the height of the virtual contact point given its position on the desk.

Given the virtual object or surface, and the robot position in Unity space (x, y, z_0) , where z_0 is the height of the table, we cast a ray vertically from a high elevation (above the height of the virtual geometry) $(x, y, z_0 + H)$ where H is 1 meter, down until it intersects the virtual surface at (x, y, z) . We obtain the distance until the ray hits to the surface $0 \leq d_{(x,y)} \leq H$, thus we get the height at which the HapticBot needs to render the virtual surface, with $height_{(x,y)} = z - z_0 = H - d_{(x,y)}$. The ray casting is performed from above and not from the robot height, to avoid culling effects, as the virtual geometry is facing up toward the user. To obtain the tilting angle of the robot’s top surface (e.g., to render a tilted roof of a house), we cast two rays, each at the locations of each actuator attachment to the plane, and change the actuator heights accordingly.

4.3.3 Target Assignment for Haptic Interactions. We perform this height measuring at each 0.5 cm grid point of the mat (55 cm × 55 cm) at every 0.5 seconds. Based on this, the system obtains the height map of the playground area. With this, the system can identify regions that the robot should provide a touchable plane (i.e., the region that has a surface higher than a certain height, in our setting where 1 cm is the threshold). The system sets these regions as target objects or surfaces, then moves the robot within this target region while the user moves her finger laterally. When the number of regions exceeds the number of robots, we optimize the target position based on the finger position. For example, when the robots need to cover four separate target regions, the robots move across the region that is closest to the current finger position. Multiple robots can also support a single large continuous region when there is enough of robots. This way, we can assign the target position to each of virtual objects (e.g., virtual houses, buildings, and cars on a map), either they are static (e.g., buildings) or dynamic (e.g., moving cars). By leveraging the combination of hand tracking and dynamic target assignment, the smaller number of robots can successfully simulate the virtual haptic proxy.

4.3.4 Finger Tracking and Continuous Positioning. To enable the robot to encounter the user’s finger whenever she wishes to touch a virtual surface, a camera tracks the position of the user’s finger, and

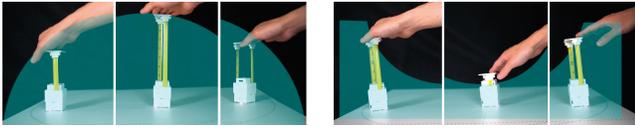


Figure 12: A single robot simulates a virtual surface.

moves the robot to minimize the robot’s distance to the finger. We use the built-in hand tracking of the Oculus Quest that generates the user’s finger position inside the Unity virtual space. To do that we need to be able to generate a reference between Unity coordinate system, and Toio’s mat reference. We use a simple calibration process to match the coordinate systems between Toio mat and the virtual floor in the Unity. The user calibrates the mat through the following two steps. First, the user places her index finger at the center of the mat and presses a space key then the center of the virtual mat becomes the location of the index finger. Second, the user places her index finger on the left bottom corner and presses the space key again, then the virtual mat rotates to match with the physical mat. Although this brief calibration routine is needed to synchronize the Toio and Oculus coordinate systems, it does not limit the deployability and portability—the user can pack all parts of the system in a backpack, go to a new room, and set it up there with minimal effort.

The Oculus tracks the user’s hands at 60Hz and the target position is transmitted to the corresponding robot. The system has enough bandwidth to support multiple robots (we use seven robots in total). By leveraging the combination of hand tracking and dynamic target assignment, a small number of robots can simulate a large haptic surface.

4.4 Technical Evaluation

We evaluated the technical capabilities of HapticBots in terms of the following aspects. The table summarizes the results of measurements for each of them.

| | |
|---|----------|
| 1-a) Maximum Vertical Load Holding (at 12 cm height) | 21.8 N |
| 1-b) Maximum Vertical Load Holding (at 25 cm height) | 13.5 N |
| 1-c) Maximum Vertical Load Holding (tilted at 25 cm height) | 9.31 N |
| 2) Maximum Vertical Load Lifting | 10.8 N |
| 3) Maximum Horizontal Pushing Force | 0.31 N |
| 4) Speed of Vertical Extension | 2.8 cm/s |
| 5) Speed of Horizontal Movement | 24 cm/s |
| 6-a) Average Target Reach Time (One Robot) | 2.0 sec |
| 6-b) Average Target Reach Time (Three Robots) | 1.7 sec |
| 6-c) Average Target Reach Time (Seven Robots) | 1.6 sec |
| 7) Average Vertical Extension Error | 3 mm |
| 8) Average Tilting Angle Error | 5 deg |
| 9-a) Average Position Error | 3 mm |
| 9-b) Average Orientation Error | 3 deg |
| 10) Latency | 80 ms |

4.4.1 Method. 1-3) We employed a Powlaken’s food scale (accurate to 0.1g) for force measurements and analyzed the number via video recordings. For the load-bearing, we put the robot on top of the

scale and pushed down with hands until the reel buckles. For the maximum force of vertical extension, we place a fixed surface above the robot so that it can push against it while sitting on on the scale that measures the vertical force plus weight of the robot (which is subtracted). Then, the robot extends the actuator and measure the force to push the object through the reaction force measured with the scale on the bottom. For the horizontal pushing force, we place the scale horizontally and let the robot bump and push the scale. We measured the maximum impact force against the scale. 4-5) We measured the speed of the robot and linear actuator via video recordings of the robot moving next to a ruler. 6) We measured the time to reach the target via video recordings. The target points are randomly assigned and measured the time until the last robot reaches the target. We evaluated this with one, three, and seven robots. 7-8) For the extension and tilting accuracy of the linear actuator, we took three measurements of the extension length given the same motor activation duration for three different duration values. 9) For position and orientation accuracy, we logged the distance and angle deviation between the robot and its target with our control system, given a certain duration. 10) We measured the latency of each step (i.e., Wi-fi communication, tracking, and computation of path planning) by comparing timestamps from the start of each event to its conclusion with four attempts for five robots.

4.4.2 Results. The robot’s vertical load-bearing capability is 13.53 N (at 25 cm height) and load lifting capability is 10.08 N. This is a significant improvement of the existing similar work (e.g., 0.3 N in [50]), which is important for haptic devices. The maximum speed of the vertical extension was 2.8 cm/sec. The average target reach time ranges from 1.6 to 2.0 sec with the maximum speed of the robot’s horizontal movement was 24 cm/sec. This is slightly slower than an unmodified Toio robot due to the added weight. We found that the moving robot can deliver a maximum force of 0.31 N for horizontal pushing force. The measured average positional error for vertical extension and horizontal movement were both 3 mm. Finally, the measured latency was 80 ms for the total latency of the control loop with three robots.

5 USER EVALUATION

We conducted a user study to evaluate the HapticBots ability to simulate: (1) haptic encounters of different types, (2) continuous touch on different tilts. In our evaluation we aim to compare the performance of HapticBots rendering continuous surfaces to a shape display. To do so we designed 2 tasks and recruited 6 participants for each task. Totalling 12 participants ages 18 to 50 (8 female). Most of our participants did not have any prior experience with haptic devices. After their corresponding task, participants completed an offline questionnaire.

5.1 Task 1: Haptic Encounters

In this task, we evaluate the ability to render various objects of different size, form, and texture by a shape changing robot. We are interested in how HapticBots can “approximate” some of the presented objects. To do so, we introduce a larger range of objects with different convexity and/or texture, including a mag cup, a rubik’s cube, a tennis ball, a wrench, and a sea urchin.

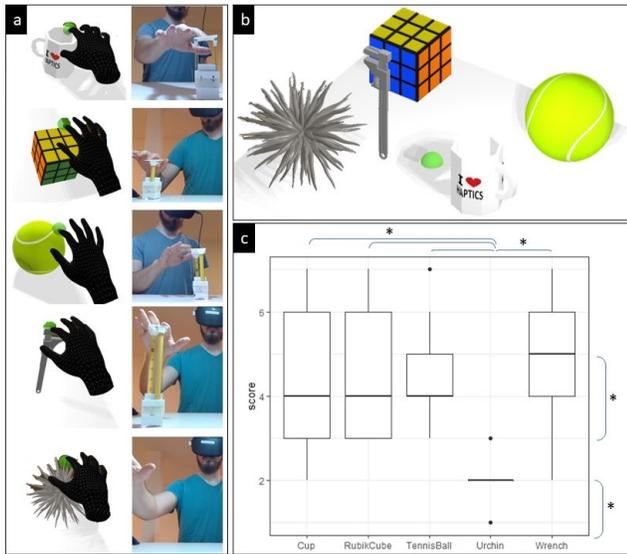


Figure 13: a) Participant in the haptic encounter study reaching for the 5 different objects. b) The five objects that they encounter in different areas of the study table and sizes. c) Boxplot of the realism score results. Marked with an asterisk significant differences. The thick lines are the medians, and the boxes are the interquartile ranges (IQR). The whiskers follow the standard convention of extending to 1.5 times the IQR or the maximal/minimal data point.

We ask participants to rate their experience in response to the question: "How realistic did the touch encounter feel?" from 1 to 7. Participants in this task were requested to only touch the top of the appearing object for 10 seconds with one finger, but once touched, they were free to move their finger over the surface and the robot which moved to be under the finger tip and adapt the height to the virtual object (see supplementary video). Each of the five objects appeared two times and the average score was computed for each object (Figure 13).

5.1.1 Results. Overall, the results (Figure 13) show that all objects could be realistically rendered by HapticBots during the encounters. And all of them, minus the spiky urchin, scored significantly higher than 3 (Wilcoxon signed rank test with continuity correction $V=26$, $p<0.05$), the mean score was 4.55 ± 1.4 SD. The spiky urchin that was significantly lower than all the rest ($V=36$, $p<0.01$), which scored 2 ± 0.7 SD, and despite the very non-compatible shape of the object scored significantly higher than 1 ($V=28$, $p=0.01$).

5.2 Task 2: 3D Continuous Touch

In this task, we evaluate the ability to render a large continuous surface of the different tilt angles. To evaluate this, we presented participants with a total of five different tilted virtual surfaces of 0, 15, 30, 50 and 70 degrees of inclination. Then, we measured how precisely the participants can recognize each tilt angle.

The second task's motivation is to evaluate the capability of large surface rendering with a single robot (e.g., Section 3.2.1),

particularly compared to the other state-of-the-art encountered-type approach. To date, the only approach that can dynamically render a large haptic surface is shape displays (see Related Work); thus we aimed to compare the performance with the shape display approach. However, most large pin-based shape displays are rare due to their high cost, making it infeasible to replicate or prepare the fully functional shape display only for this task. Thus, we built a wooden mock-up shape display for our base condition to simulate the shape display's haptic sensation. Our mock-up have a stair shape instead of a slope, is to mimic the discrete pins of the shape display. The resolution of our mock up shape display is 1.5 cm, following prior works [13, 25, 37]. We only used the HapticBots' lateral motion for fair comparison and did not use the tilt functionality.

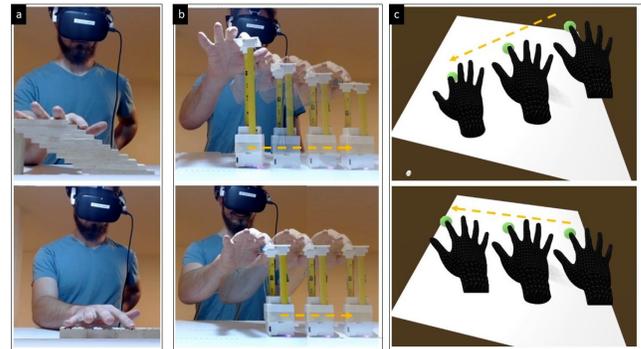


Figure 14: a) Participant assessing the tilt on a static shape display made of wooden bricks. b) Participant assessing the tilt using HapticBots. c) First Person Perspective inside the HMD during the tilt estimation task.

The users, while in VR, could not see the tilt but only a green dot representing the position they needed to touch with their fingers during the task. This green dot appeared in their HMD that they needed to keep following and touching in order to advance in the user test. At the end of each tilted surface, participants had to guess the inclination and then answer the question "How much did it feel like you were touching a continuous line?" from 1 to 7.

5.2.1 Results. We compare the results for two scenarios: using HapticBots and using a mock-up shape display made of wood (Figure 14).

Tilt Estimation: Regarding the tilt estimation, neither the repeated measures ANOVA nor the pairwise Wilcoxon signed rank test with continuity correction showed significant differences between the responses with HapticBots or the static shape display ($V=7$, $p=0.5$). Post-hoc analysis per each of the different tilts also revealed no differences per any particular tilt, and none of the errors was greater than 15° . Indeed, the error was quite low with both devices and participants only got 3.6 ± 0.6 SD errors out of the 15 tilts in the static shape display condition, and 3.9 ± 0.75 SD in the HapticBots condition. This shows the task was equivalently difficult to perform in both the static shape display and the HapticBots and that it was well designed to not produce ceiling effects as there were still some errors in perception for about 20% of the trials. Furthermore, the fact that the HapticBots tilt perception error wasn't

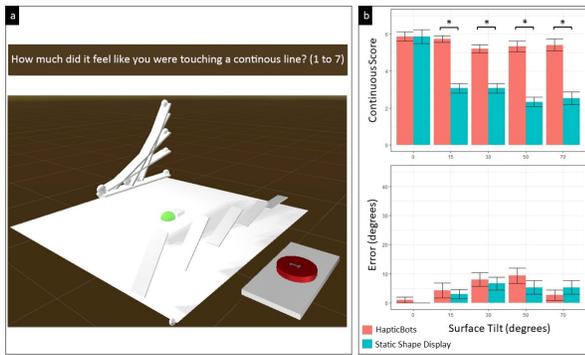


Figure 15: a) VR evaluation. After performing the continuous touch participants had to select the tilt from a menu of tilt options and then respond to the continuity question. b) Bar charts with the scores for the perceived continuity and tilt error plots with HapticBots and static shape display (error bars represent Standard Deviation).

different than the error perceived in a real physical tilt shows the quality of control achieved on the HapticBots for the task.

Continuity Score: For the flat surfaces (0 degrees), both conditions performed very high on the continuity score. However, the continuity score rapidly deteriorated for the static shape display whereas HapticBots maintained a good performance across the different tilts. A repeated measures ANOVA showed that both condition and tilt were significant main effects ($F(1,4)=73.9, p=0.001$ and $F(4,16)=12.5, p<0.0001$ respectively). And that there was an interaction effect between condition and tilt ($F(4,16)=6.031, p=0.004$) on the score. In the post-hoc analysis we confirmed the interaction, there was a significant drop in continuity scores for the static shape display (Wilcoxon signed rank test $V = 0, p = 0.001$). While, no significant drop was found for the HapticBots tilt estimation.

5.2.2 Discussion. The results show that HapticBots can be used in dynamic continuous mode without reduced slope detection compared to a solid slope of a static shape display. The correct classification of tilts for the HapticBots condition also shows that the presented slopes were overall distinguishable between them. However, there are two limitations in this study. First, since the sample size of our user study is relatively small, a further in-depth user study may be required to fully evaluate our approach. Second, our user study only used the mock-up, static shape display and did not compare with fully-functional shape displays or robotic arms. Since the development of these functional systems goes beyond the scope of this paper, we are interested in further evaluating our approach as future work.

6 APPLICATION SCENARIOS

6.1 Education and Training

VR is an accessible way to create realistic training setups to improve skills or prepare for complex situations before they happen in real life. With its fast encounter-type approach, users of HapticBots can train their muscle memory to learn where different physical

elements such as the interface of a flight cockpit are located (Figure 2). HapticBots can simulate continuous surfaces, and the robots can follow the user’s fingers as they move and even elevate them during palpation diagnostics. These features could be relevant for medical education and surgery training (Figure 2).

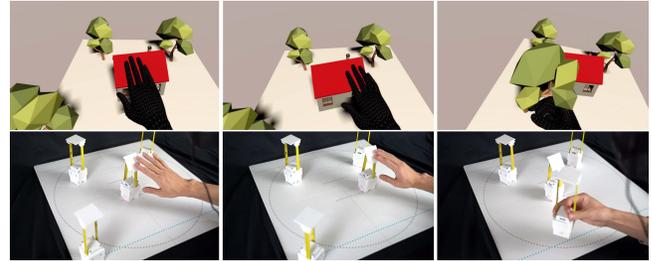


Figure 16: Modeling a virtual environment: Users touch and rearrange 3D models by handling the robots.

6.2 Design and 3D Modeling

In addition to its continuous shape rendering capabilities, the design of HapticBots being based on dual actuators makes the system robust to lateral bending and provides the ability to control different tilts to render topography of a terrain surface. This enables activities like map and city exploration or terrain simulation, which can be necessary for architectural design or virtual scene/object modeling (Figure 17).

6.3 Remote Collaboration

Tangible interfaces can enrich remote collaboration through shared synchronized physical objects [7]. Using two connected HapticBots setups, we can reproduce remote physical objects, or introduce shared virtual objects. Figure 17 shows an example of a chess game application where the user moves the chess figures physically through robots. As a user is replacing an opponent piece from the board, she can feel the robots at the correct place on the board. This interaction could extend to multiple end points to create shared, distributed multi-user spaces.

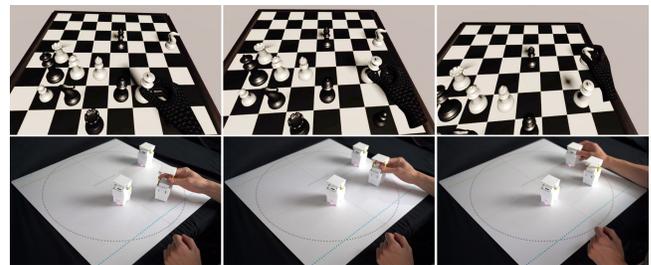


Figure 17: Playing chess: When a user moves a physical piece, the virtual piece moves accordingly and is transmitted to the opponent.

Through its encountered-type haptic rendering approach, HapticBots physically renders information about sizes, locations and

heights of objects on-demand where the user touches them. HapticBots also enables direct interaction with the 3D models, where users can pick up and move the robots to modify objects in the terrain and to redesign the environment (Figure 17).

6.4 Gaming and Entertainment

World-building games like Minecraft often rely on players constructing terrains and objects. However the lack of haptics distracts from the immersive experience. HapticBots can augment the game experience during construction or game play in these VR games. Apart from the previously mentioned interactions to grab, push, and encounter, multiple robots can act in coordinated ways to simulate larger objects. They can also provide proxy objects that interact with additional props and game controllers, such as an axe in Minecraft (see Figure 18).

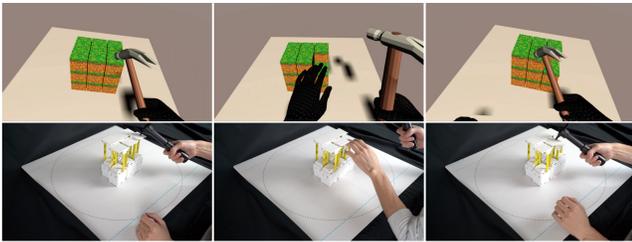


Figure 18: Playing Minecraft: The robots render the virtual terrain. When the player hammers with a pickaxe on a real robot, it changes height and the virtual terrain is chipped away.

7 LIMITATIONS AND FUTURE WORK

HapticBots is our first functional prototype to demonstrate the concept of a distributed encountered-type haptic device. The following future work could improve this approach further.

Number of Robots and Coordinated Swarm Behaviors. Our system employs relatively a small number of robots (e.g., 2-7 robots in the demonstrated applications). This was partly because we noticed that the small number of robots are often sufficient to render various virtual objects in our interaction space (55 cm × 55 cm), as a single robot can also render the continuous surface. However, by employing more robots, we could further leverage the swarm behavior of the robots and enrich the haptic interaction. When scaling the robots, there are a couple of technical challenges that need to be solved. For example, in our settings, the maximum number of Bluetooth pairing is limited with 7 bots on a single computer, but this can be scaled, for example, by using multiple Raspberry Pi as access points [38]. Path planning of the robots also introduces another challenge, as more sophisticated swarm controls would be required to avoid the collision. On the other hand, the large number of robots could benefit to support more expressive haptic experiences as discussed in Section 3. As future work, we will continuously explore interactions and applications that benefit from complex swarm behaviors.

Haptic Retargeting. While our prototype is relatively fast and accurate, the current prototype still has some technical limitations, such as the speed of vertical actuation, the maximum height of the actuator, and speed and accuracy of locomotion, which causes generates the physical-virtual discrepancy. For example, when the hand moves too fast, the robot cannot change its shape and/or position in time for a good fidelity haptic perception. Also, our robots may need to retract or extend the actuator before traveling, which may also negatively impact the response time. To mitigate this mechanical limitation, we are interested in incorporating haptic retargeting [5], specifically for the combination of continuous surface simulation which has not been explored in prior works [14]. Related to this, the better anticipation of potential touch events would also improve the system’s performance. For example, we could leverage the user’s behavior such as eye-gaze, or anticipate based on the context of the application. We will further investigate how these approaches can enable a better experience.

Combining with Customizable Passive Haptic Proxy. It is also interesting to see the possibilities to augment existing robots with custom haptic proxy. For example, the Hermits system [38] introduced the ability for robots to reconfigure their functionality through mechanical shell add-ons. This way it is possible to augment the robot with various geometries (e.g., attached haptic props), surface textures (e.g., different robots have different materials), and even functional objects (e.g., buttons, sliders) that the user can feel when touching the robots. Such ideas are partially explored in the prior works (e.g., adding a string for force feedback of fishing application [56]), but there should be broader design space for combining shape-changing mobile robots with external passive props. Also, the coordinated robots can actuate an existing environment, materials, and objects (e.g., similar to RoomShift [46], Sublimate [35] or Molebot [33]). We are interested in exploring the design space of this *inter-material interaction* [13] between HapticBots and external environments.

User Force. The stability of the robots is another technical limitation. We noticed that the robot’s movement becomes less stable—like the robot could be knocked over—while being touched, especially when the actuator is tilted or fully extended (e.g., over 25-30 cm). This limitation can be alleviated by using a magnetic sheet and attaching a magnet to the robot. Our current prototype also does not have the force-sensing capability, thus the input and interaction capability is limited. An additional force sensing can expand the tangible interaction modality. For example, the user could push the actuator to lower the height or stretch it to extend. These interactions can bring additional rich embodied and tangible interactions into VR. We are interested in further exploring this aspect.

Size and Resolution. Finally, there is a trade-off between high resolution contact points and complexity of the position control. Currently, the footprint of our prototype robot is 5 × 5 cm. Scaling down to finger-tip size (e.g., 1-2 cm) would enable the higher resolution rendering and more portable system, but also introduces a number of technical challenges in actuation robustness. For example, as we discussed in the implementation section, there is a trade-off between the size of the motor, the speed of actuation, and the active or holding force against the user’s hand. The actuators’ motor and

gear torque also affect the stiffness of the tape—as the tape becomes stiffer, a stronger motor will be required to roll and unroll, which makes the overall size large. However, unlike using swarm robots for shape-changing displays [50], the use for haptics in VR may not always require such a high-resolution contact point because we can leverage the visual illusion of the touch points. Therefore, there might be a certain threshold of a minimum required size and a maximum number of robots. Similar to visuo haptic illusion for shape displays [1], investigating the scalability and trade-off of swarm robots for haptics in VR would be also an interesting research question in the future.

8 CONCLUSION

We introduced distributed encountered type haptics, a novel encountered type haptic concept with tabletop shape-changing robots. Compared to previous encountered type haptics, such shape displays and robotic arm, our proposed approach improves the deployability and scalability of the system, while maintaining generalizability for general-purpose haptic applications. This paper contributes to this concept, a set of haptic interaction techniques, and its demonstration with HapticBots system. The design and implementation of HapticBots were described and the results of our user evaluation show the promise of our approach to simulating various shapes rendering. Finally, we demonstrated various haptic applications for different VR scenarios, including training, gaming and entertainment, design, and remote collaboration.

REFERENCES

- [1] Parastoo Abtahi and Sean Follmer. 2018. Visuo-haptic illusions for improving the perceived performance of shape displays. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [2] Parastoo Abtahi, Benoit Landry, Jackie Yang, Marco Pavone, Sean Follmer, and James A Landay. 2019. Beyond the force: Using quadcopters to appropriate objects and the environment for haptics in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [3] Bruno Araujo, Ricardo Jota, Varun Perumal, Jia Xian Yao, Karan Singh, and Daniel Wigdor. 2016. Snake Charmer: Physically Enabling Virtual Objects. In *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 218–226.
- [4] Jatin Arora, Aryan Saini, Nirmita Mehra, Varnit Jain, Shwetank Shrey, and Aman Parnami. 2019. Virtualbricks: Exploring a scalable, modular toolkit for enabling physical manipulation in vr. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [5] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D Wilson. 2016. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *Proceedings of the 2016 chi conference on human factors in computing systems*. ACM, 1968–1979.
- [6] Hrvoje Benko, Christian Holz, Mike Sinclair, and Eyal Ofek. 2016. Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 717–728.
- [7] Scott Brave, Hiroshi Ishii, and Andrew Dahley. 1998. Tangible interfaces for remote collaboration and communication. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*. 169–178.
- [8] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D Wilson. 2017. Sparse haptic proxy: Touch feedback in virtual environments using a general passive prop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3718–3728.
- [9] Inrak Choi, Elliot W Hawkes, David L Christensen, Christopher J Ploch, and Sean Follmer. 2016. Wolverine: A wearable haptic interface for grasping in virtual reality. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 986–993.
- [10] Inrak Choi, Eyal Ofek, Hrvoje Benko, Mike Sinclair, and Christian Holz. 2018. Claw: A multifunctional handheld haptic controller for grasping, touching, and triggering in virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [11] Christer Fähræus, Johan Lindgren, and Stefan Burström. 2007. Electronic pen. US Patent 7,239,306.
- [12] Cathy Fang, Yang Zhang, Matthew Dworman, and Chris Harrison. 2020. Wireality: Enabling Complex Tangible Geometries in Virtual Reality with Worn Multi-String Haptics. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–10. <https://doi.org/10.1145/3313831.3376470>
- [13] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: dynamic physical affordances and constraints through shape and object actuation. In *Uist*, Vol. 13. 2501988–2502032.
- [14] Eric J Gonzalez, Parastoo Abtahi, and Sean Follmer. 2020. REACH+: Extending the Reachability of Encountered-type Haptics Devices through Dynamic Redirection in VR. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM.
- [15] Darren Guinness, Annika Muehlbradt, Daniel Szafir, and Shaun K Kane. 2019. RoboGraphics: Dynamic Tactile Graphics Powered by Mobile Robots. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*. 318–328.
- [16] Zachary M Hammond, Nathan S Usevitch, Elliot W Hawkes, and Sean Follmer. 2017. Pneumatic reel actuator: Design, modeling, and implementation. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 626–633.
- [17] Vincent Hayward, Oliver R Astley, Manuel Cruz-Hernandez, Danny Grant, and Gabriel Robles-De-La-Torre. 2004. Haptic interfaces and devices. *Sensor review* (2004).
- [18] Zhenyi He, Fengyuan Zhu, and Ken Perlin. 2017. Physhare: sharing physical interaction in virtual reality. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 17–19.
- [19] Seongkook Heo, Christina Chung, Geehyuk Lee, and Daniel Wigdor. 2018. Thor's hammer: An ungrounded force feedback device utilizing propeller-induced propulsive force. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–11.
- [20] Anuruddha Hettiarachchi and Daniel Wigdor. 2016. Annexing reality: Enabling opportunistic use of everyday objects as tangible proxies in augmented reality. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1957–1967.
- [21] Koichi Hirota and Michitaka Hirose. 1995. Simulation and presentation of curved surface in virtual reality environment through surface display. In *Proceedings Virtual Reality Annual International Symposium '95*. IEEE, 211–216.
- [22] Matthias Hoppe, Pascal Knierim, Thomas Kosch, Markus Funk, Lauren Futami, Stefan Schneegass, Niels Henze, Albrecht Schmidt, and Tonja Machulla. 2018. VRHapticDrones: Providing Haptics in Virtual Reality through Quadcopters. In *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 7–18.
- [23] Brent Edward Insko, M Meehan, M Whitton, and F Brooks. 2001. *Passive haptics significantly enhances virtual environments*. Ph.D. Dissertation. University of North Carolina at Chapel Hill.
- [24] Hiroo Iwata. 1999. Walking about virtual environments on an infinite floor. In *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*. IEEE, 286–293.
- [25] Hiroo Iwata, Hiroaki Yano, Fumitaka Nakaizumi, and Ryo Kawamura. 2001. Project FEELEx: adding haptic surface to graphics. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 469–476.
- [26] Chang-Min Kim and Tek-Jin Nam. 2015. G-raff: an elevating tangible block for spatial tabletop interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 4161–4164.
- [27] Lawrence H Kim and Sean Follmer. 2017. Ubiswarm: Ubiquitous robotic interfaces and investigation of abstract motion as a display. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 66.
- [28] Lawrence H Kim and Sean Follmer. 2019. Swarmhaptics: Haptic display with swarm robots. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.
- [29] Luv Kohli, Eric Burns, Dorian Miller, and Henry Fuchs. 2005. Combining passive haptics with redirected walking. In *Proceedings of the 2005 international conference on Augmented tele-existence*. ACM, 253–254.
- [30] Robert Kovacs, Eyal Ofek, Mar Gonzalez-Franco, Alexa F Siu, Sebastian Marwecki, Christian Holz, and Mike Sinclair. 2020. Haptic PIVOT: On-Demand Handhelds in VR. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA.
- [31] Mathieu Le Goc, Lawrence H Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. 2016. Zooids: Building blocks for swarm user interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 97–109.
- [32] Jaeyeon Lee, Mike Sinclair, Mar Gonzalez-Franco, Eyal Ofek, and Christian Holz. 2019. TORC: A virtual reality controller for in-hand high-dexterity finger interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [33] Narae Lee, Juwhan Kim, Jungsoo Lee, Myeongsoo Shin, and Woohun Lee. 2011. Molebot: mole in a table. In *ACM SIGGRAPH 2011 Emerging Technologies*. 1–1.

- [34] Daniel Leithinger, Sean Follmer, Alex Olwal, and Hiroshi Ishii. 2015. Shape displays: Spatial interaction with dynamic physical form. *IEEE computer graphics and applications* 35, 5 (2015), 5–11.
- [35] Daniel Leithinger, Sean Follmer, Alex Olwal, Samuel Luescher, Akimitsu Hogge, Jinha Lee, and Hiroshi Ishii. 2013. Sublimate: state-changing virtual and physical rendering to augment interaction with shape displays. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1441–1450.
- [36] William A McNeely. 1993. Robotic graphics: a new approach to force feedback for virtual reality. In *Proceedings of IEEE Virtual Reality Annual International Symposium*. IEEE, 336–341.
- [37] Ken Nakagaki, Daniel Fitzgerald, Zhiyao Ma, Luke Vink, Daniel Levine, and Hiroshi Ishii. 2019. inFORCE: Bi-directionalForce'Shape Display for Haptic Interaction. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 615–623.
- [38] Ken Nakagaki, Joanne Leong, Jordan L Tappa, Joao Wilbert, and Hiroshi Ishii. 2020. HERMITS: Dynamically Reconfiguring the Interactivity of Self-Propelled TUIs with Mechanical Shell Add-ons. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM.
- [39] James Patten and Hiroshi Ishii. 2007. Mechanical constraints as computational constraints in tabletop tangible interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 809–818.
- [40] Sharif Razzaque, Zachariah Kohn, and Mary C Whitton. 2005. *Redirected walking*. Citeseer.
- [41] Adalberto L Simeone, Eduardo Velloso, and Hans Gellersen. 2015. Substitutional reality: Using the physical environment to design virtual reality experiences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3307–3316.
- [42] Mike Sinclair, Eyal Ofek, Mar Gonzalez-Franco, and Christian Holz. 2019. CapstanCrunch: A Haptic VR Controller with User-Supplied Force Feedback. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (New Orleans, LA, USA) (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 815–829. <https://doi.org/10.1145/3332165.3347891>
- [43] Alexa F Siu, Eric J Gonzalez, Shenli Yuan, Jason B Ginsberg, and Sean Follmer. 2018. Shapeshift: 2D spatial manipulation and self-actuation of tabletop shape displays for tangible and haptic interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 291.
- [44] Anthony Steed, Eyal Ofek, Mike Sinclair, and Mar Gonzalez-Franco. 2021. A Mechatronic Shape Display based on Auxetic Materials. In *Nature Communications*.
- [45] Yuqian Sun, Shigeo Yoshida, Takuji Narumi, and Michitaka Hirose. 2019. PoCoPo: A handheld vr device for rendering size, shape, and stiffness of virtual objects in tool-based interactions. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [46] Ryo Suzuki, Hooman Hedayati, Clement Zheng, James L Bohn, Daniel Szafir, Ellen Yi-Luen Do, Mark D Gross, and Daniel Leithinger. 2020. RoomShift: Room-scale Dynamic Haptics for VR with Furniture-moving Swarm Robots. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–11.
- [47] Ryo Suzuki, Jun Kato, Mark D Gross, and Tom Yeh. 2018. Reactile: Programming Swarm User Interfaces through Direct Physical Manipulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 199.
- [48] Ryo Suzuki, Ryosuke Nakayama, Dan Liu, Yasuaki Kakehi, Mark D. Gross, and Daniel Leithinger. 2020. LiftTiles: Constructive Building Blocks for Prototyping Room-scale Shape-changing Interfaces. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM.
- [49] Ryo Suzuki, Abigale Stangl, Mark D Gross, and Tom Yeh. 2017. FluxMarker: Enhancing Tactile Graphics with Dynamic Tactile Markers. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*. 190–199.
- [50] Ryo Suzuki, Yasuaki Zheng, Clement Kakehi, Tom Yeh, Ellen Yi-Luen Do, Mark D Gross, and Daniel Leithinger. 2019. ShapeBots: Shape-changing Swarm Robots. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM.
- [51] Shohei Takei, Makoto Iida, and Takeshi Naemura. 2011. Kinereels: extension actuators for dynamic 3d shape. In *ACM SIGGRAPH 2011 Posters*. ACM, 84.
- [52] Shan-Yuan Teng, Tzu-Sheng Kuo, Chi Wang, Chi-huan Chiang, Da-Yuan Huang, Liwei Chan, and Bing-Yu Chen. 2018. Pupop: Pop-up prop on palm for virtual reality. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 5–17.
- [53] Shan-Yuan Teng, Cheng-Lung Lin, Chi-huan Chiang, Tzu-Sheng Kuo, Liwei Chan, Da-Yuan Huang, and Bing-Yu Chen. 2019. TilePoP: Tile-type Pop-up Prop for Virtual Reality. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM.
- [54] Jur Van den Berg, Ming Lin, and Dinesh Manocha. 2008. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*. IEEE, 1928–1935.
- [55] Emanuel Vonach, Clemens Gatterer, and Hannes Kaufmann. 2017. VRRobot: Robot actuated props in an infinite virtual environment. In *2017 IEEE Virtual Reality (VR)*. IEEE, 74–83.
- [56] Yuntao Wang, Zichao Chen, Hanchuan Li, Zhengyi Cao, Huiyi Luo, Tengxiang Zhang, Ke Ou, John Raiti, Chun Yu, Shwetak Patel, et al. 2020. Movevr: Enabling multiform force feedback in virtual reality using household cleaning robot. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [57] Eric Whitmire, Hrvoje Benko, Christian Holz, Eyal Ofek, and Mike Sinclair. 2018. Haptic revolver: Touch, shear, texture, and shape rendering on a reconfigurable virtual reality controller. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 86.
- [58] Yan Yixian, Kazuki Takashima, Anthony Tang, Takayuki Tanno, Kazuyuki Fujita, and Yoshifumi Kitamura. 2020. ZoomWalls: Dynamic Walls that Simulate Haptic Infrastructure for Room-scale VR Worlds. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM.
- [59] Yasuyoshi Yokokohji, Nobuhiko Muramori, Yuji Sato, and Tsuneo Yoshikawa. 2005. Designing an encountered-type haptic display for multiple fingertip contacts based on the observation of human grasping behaviors. *The International Journal of Robotics Research* 24, 9 (2005), 717–729.
- [60] Yiwei Zhao, Lawrence H Kim, Ye Wang, Mathieu Le Goc, and Sean Follmer. 2017. Robotic assembly of haptic proxy objects for tangible interaction and virtual reality. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. ACM, 82–91.
- [61] Kening Zhu, Taizhou Chen, Feng Han, and Yi-Shiun Wu. 2019. HapTwist: creating interactive haptic proxies in virtual reality using low-cost twistable artefacts. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.