

# RoomShift: Room-scale Dynamic Haptics for VR with Furniture-moving Swarm Robots

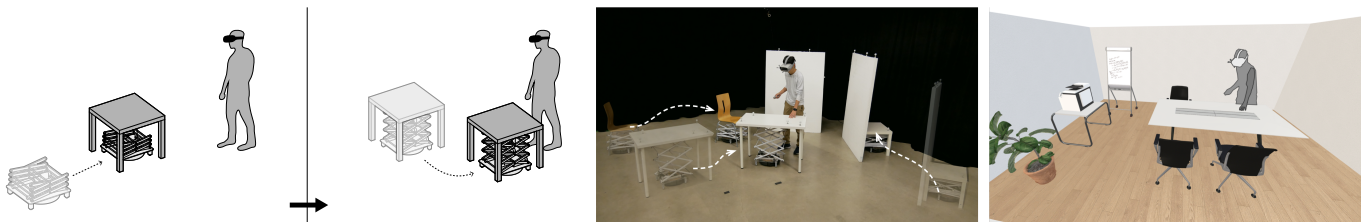
Ryo Suzuki<sup>1</sup>, Hooman Hedayati<sup>1</sup>, Clement Zheng<sup>2</sup>, James Bohn<sup>1</sup>,  
Daniel Szafir<sup>1,2</sup>, Ellen Yi-Luen Do<sup>1,2</sup>, Mark D. Gross<sup>1,2</sup>, Daniel Leithinger<sup>1,2</sup>

University of Colorado Boulder,

<sup>1</sup>Department of Computer Science, <sup>2</sup>ATLAS Institute

{ryo.suzuki, hooman.hedayati, clement.zheng, james.bohn

daniel.szafir, ellen.do, mdgross, daniel.leithinger}@colorado.edu



**Figure 1.** RoomShift is composed of a swarm of shape-changing robots for haptic feedback in VR. RoomShift robots move beneath a piece of furniture to lift, move and place it. Multiple robots move furniture to construct a physical haptic environment collectively. The corresponding virtual scene is shown, with a human silhouette added for a reference.

## ABSTRACT

RoomShift is a room-scale dynamic haptic environment for virtual reality, using a small swarm of robots that can move furniture. RoomShift consists of nine shape-changing robots: Roombas with mechanical scissor lifts. These robots drive beneath a piece of furniture to lift, move and place it. By augmenting virtual scenes with physical objects, users can sit on, lean against, place and otherwise interact with furniture with their whole body; just as in the real world. When the virtual scene changes or users navigate within it, the swarm of robots dynamically reconfigures the physical environment to match the virtual content. We describe the hardware and software implementation, applications in virtual tours and architectural design and interaction techniques.

## CCS Concepts

•Human-centered computing → Human computer interaction (HCI);

## Author Keywords

haptic interfaces; room-scale haptics; virtual reality; swarm robots

## INTRODUCTION

There is a clear need to provide haptic sensations in virtual environments. Recent advances in display and tracking technologies promise immersive experience in virtual reality, but objects seen in VR such as walls and furniture are only visual: the user cannot touch, feel, sit on, or place objects on them. This limits the sense of full immersion in the virtual world. To overcome these limitations, various haptic interfaces have been explored. In the previous work, most haptic interfaces focus on finger-tip haptic feedback with actuated controllers [6, 11, 12] or on-body haptic sensations with wearable devices [35, 38, 14, 36, 42]. In contrast, encountered-type haptic feedback with a dynamic environment promises to increase the immersion of virtual experiences [8, 10, 51, 53], which are difficult to achieve using an only handheld or wearable haptic devices. Through a dynamic haptic environment, users can touch and interact with the whole virtual scene with their bodies — they can walk, sit on, and lean against objects in the VR environment. Existing approaches for actuated environments, however, are often limited in speed of transformation (e.g., slow transformation with inflatables [46, 51]) and the range of supported interactions (e.g., only walking [24]).

This paper introduces RoomShift, a room-scale dynamic haptic environment for virtual reality. RoomShift provides haptic sensations by reconfiguring physical environments using a small swarm of robot assistants. Inspired by shelf-moving robots [16, 56] that are used in robotic warehouses, we developed a swarm of shape-changing robots that can move a range of existing furniture. Each robot has a mechanical lift that extends from 30 cm to 100 cm to pick up, carry, and place objects such as chairs, tables, and walls. This way, users can touch, sit, place, and lean against objects in the virtual environment. To synchronize the VR scene with the physical environment

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](mailto:permissions@acm.org).

CHI '20, April 25–30, 2020, Honolulu, HI, USA.

© 2020 Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-6708-0/20/04 ...\$15.00.

<http://dx.doi.org/10.1145/3313831.3376523>

in a 10 m x 10 m space, we developed software to track and control the robots with an optical motion capture system. This system continuously maps virtual touchable surfaces in the proximity of users and coordinates the robot swarm to move physical objects to their target location without colliding with each other or the users.

We investigate the use of RoomShift for real estate virtual tours and collaborative architectural design, two increasingly common application areas for VR [22]. To support these scenarios, we propose four types of basic interactions along with the spectrum between *embodied* and *controller-based* interactions: 1) walking and touching, 2) physically moving furniture, 3) teleporting, and 4) virtually moving furniture. We describe and demonstrate how these interactions can be used for architectural design and virtual tour scenarios, such as exploration of the architectural space, remote collaboration and co-designing, navigation in the large space, and virtual scene editing. In a preliminary evaluation with five participants, we test the feasibility of moving furniture robotically to simulate a static physical environment. Based on our insights, we discuss future research directions.

In short, we contribute:

1. A concept of providing room-scale dynamic haptic feedback through furniture-moving swarm robots.
2. Design and implementation of RoomShift: mechanical design of a shape-changing robot with a scissor lift, tracking techniques, software, and hardware implementation to synchronize the physical environment with the virtual scene.
3. Application scenarios for virtual tours and architectural design and appropriate interaction techniques.

## RELATED WORK

### Haptic Interfaces for VR

There are many approaches to providing haptic sensations for VR. Existing approaches can be largely categorized in two ways: using *passive* objects for haptic props or using *active* objects for dynamic haptic feedback.

#### *Haptic Feedback using Passive Objects*

The first approach uses existing passive artifacts as a haptic proxy for virtual objects [23]. For example, Annexing Reality [20] employs physical static objects as props in an immersive environment by matching and adjusting the shape and size of the virtual object. Haptic Retargeting [5] uses visual illusions to simulate multiple virtual objects with a single physical prop. Similarly, by combining passive objects with redirected walking [39], Kohli et al. [32] explored haptics that can go beyond the scale of human hands. Using passive objects as haptic proxies benefits from easy and low-cost haptic feedback, but it is difficult to fully represent the dynamic virtual objects because the physical props are static.

#### *Hand-held and Wearable Haptic Interfaces*

On the other hand, active haptic interfaces leverage actuated devices to provide dynamic haptic feedback. In the literature, the active haptic feedback has been mostly explored as hand-held or wearable for on-body haptic feedback. For example, prior work explores haptic feedback for hands or finger-tip

using a controller with actuated pins [6], graspable exoskeleton [11, 12], and various hand-held controllers that generate force feedback [19, 26, 40, 43]. On-body haptic interfaces have been also investigated to provide haptic sensations beyond the human hands [35]. For example, wearable haptic suits [38, 14], electric muscle simulation [36], and actuated shoes [42] have been proposed to provide haptic feedback to the user's body. Hand-held and wearable approaches have many benefits in portability and mobility, but current hand- or body-based haptic interfaces are still limited to emulate a whole environment, which is the focus of our work.

#### *Encountered-type Haptic Feedback with Robotic Devices*

The other approach of active haptic feedback is *encountered-type haptics*, in which haptic devices dynamically deliver physical props when the user makes contact with a virtual object. Unlike the wearable approach, the encountered-type approach augments the environment, instead of the user's body. Ever since McNeely's concept of Robotic Graphics [37], robots in various form factors have investigated this encountered-type approach to emulate virtual objects and environments. For example, existing work uses shape displays to simulate dynamic surface and shapes for VR (e.g., FEELEX [25], shapeShift [1, 45]), robotic arm to simulate walls and objects (e.g., Snake Charmer [3], VRRobot [53]), and multiple movable robots to simulate dynamic terrain on which the user walks (e.g., CirculaFloor [24]). Recently, drones [2, 21] and small wheeled robots [18] have been also proposed as a way to move proxy objects for encountered-type haptic experiences. Our work is also categorized as the encountered-type haptics, but in contrast to existing approaches, we aim to emulate the whole environment of the virtual space. We achieve this by combining passive objects with robotic actuation. Instead of emulating building elements with the robot itself, we propose actuating furniture with robots, which allows more robust and larger-scale haptic feedback for VR.

#### *Large-scale Haptic Feedback*

Prior work also explores large-scale haptic feedback by reconfiguring the physical environment around the user. Haptic-Turk [8] aims to achieve this goal through the human actuation method, in which human helpers lift and tilt a player to produce a physical sensation in a video game. Similarly, TurkDeck [10] leverages human volunteers to help to reconstruct the space for realistic haptic feedback. The human actuation approach [7, 8, 9, 10] is intriguing for prototyping or simulating the large-scale haptic feedback, but is limited in availability and scalability as it always requires a group of human helpers to achieve the experience. In contrast, room-scale systems such as TilePoP [51] and LiftTiles [46] investigate the use of inflatable actuators to dynamically change the environment to provide a haptic proxy. Although inflatables are safe and low-cost, they are also slow: the speed of transformation and degrees of freedom are key limitations to providing real-time dynamic haptic sensations. To overcome this limitation, this paper explores a swarm robotic approach to reconfigure the environment. This benefits from faster spatial transformation which is important for the real-time synchronization between virtual and physical environments.

### Human-Robot Interaction with Swarm Robots

Outside of the context in haptics for VR, our work is also inspired by an emerging area of human-robot interaction, particularly with swarm robots [34, 30]. Some existing works explore the use of these small tabletop robots for everyday haptic feedback [15, 31, 47] or constructing haptic proxy objects for VR [57]. We extend these works by exploring the large-scale haptic feedback, inspired by the concept of shape-changing swarm robots [48]. In the context of human-robot interaction, some works integrate wheeled robots with furniture for spatial reconfiguration. For example, Mechanical Ottoman [44] adds a wheeled robot to existing furniture to make it autonomously mobile. Similarly, shape-shifting wall displays move to adapt to the content projected on the wall [49]. We extend prior work on robotic furniture by investigating its use for haptics in VR and proposing the required technologies and interaction techniques unique to VR applications.

### ROOMSHIFT: SYSTEM AND IMPLEMENTATION

#### Overview

RoomShift consists of a small swarm of shape-changing robots; each robot uses a Roomba as a mobile base. On this base is mounted a custom mechanical scissor lift made of two linear actuators and a metal drying rack. As the mechanical lift is compact in its closed state, the robot can move under a table or chair with 30 cm clearance, and extend the scissor lift to pick it up.

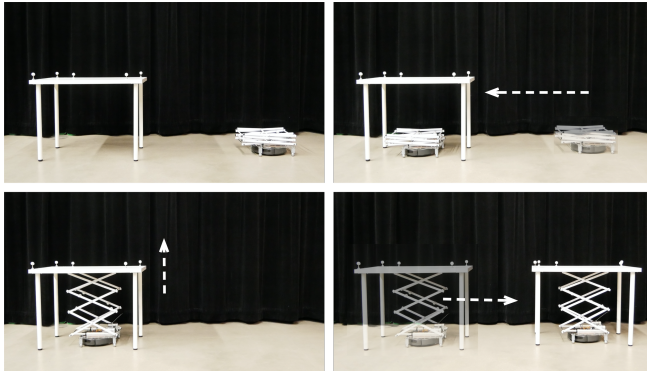


Figure 2. A RoomShift robot drives beneath a desk, lifts it by extending the scissor structure, and moves it.

The goal of RoomShift is to provide whole-body haptic interactions in VR. It does this by dynamically constructing and reconfiguring physical spatial layouts. Our approach is to reconfigure the physical room using a swarm of furniture-moving robots. These robots can relocate existing physical elements of the environment (e.g., chairs, racks, shelves, desks, as well as custom props) as encountered-type haptic interfaces that can support whole-body interactions. The key inspiration comes from the shelf-moving robots [16, 56] that are used in robotic warehouses. By bringing similar capabilities to virtual reality, we can achieve fast, robust, large-scale, and scalable dynamic haptic environments.

The position and orientation of the robots and physical props are tracked with an optical motion capture system through five retro-reflective markers, whose patterns are unique to be

identified as different objects. The motion capture system also tracks the position of the user's VR head-mounted display, so the user can walk around within a 10 m × 10 m area. A VR scene is rendered through the A-Frame API on an Oculus Go headset and the position information of the VR scene is synchronized through a server between the VR headset and the desktop computer that tracks objects and controls the robot swarm. Whenever the user loads a new scene, teleports to a new location, or changes the design of the virtual layout, the system calculates the appropriate position of physical props and drives the robots to reconstruct the physical environment.

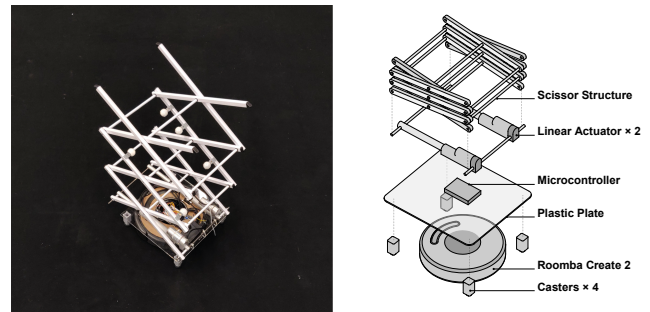


Figure 3. Mechanical design of the robot and the scissor structure.

#### Mechanical Design

RoomShift comprises nine shape-changing swarm robots based on the Roomba Create 2 [13]. For the mechanical lift structure, we repurposed an off-the-shelf expandable laundry rack (Room Essentials Compact Drying Rack) and attached two linear actuators (Homend DC12V 8 inch Stroke Linear Actuator, which extends from 32 cm to 52 cm) at the base of the rack. The linear actuators are fixed to the endpoints of the scissor structure with 8 mm steel rods, so that when the actuator contracts, the mounted scissor structure extends vertically (from 30 cm to 100 cm). The scissor structure moves at a speed of 1.3 cm / sec. To mount the scissor structure, we fixed a 6mm acrylic bottom plate (35 cm x 35 cm) and four omni-directional casters (Dorhea Ball Transfer Bearing Unit) to relieve the Roomba of most of the weight that the robot carries. Each robot moves at 20 cm / sec. Figure 3 illustrates the mechanical design of each RoomShift robot.

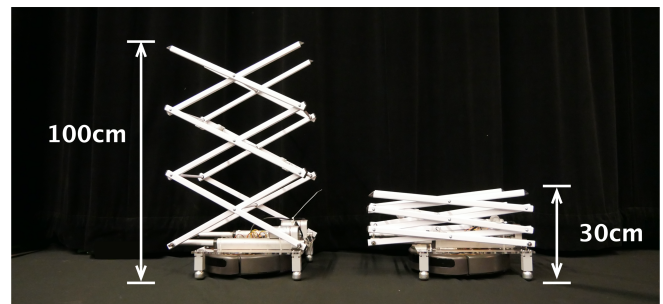


Figure 4. Each robot can extend from 30 cm to 100 cm to lift objects.

We considered and tested several actuation mechanisms such as a pneumatically-actuated inflatable structure [17, 46, 51], a deployable structure using coilable masts [28, 29], and a mechanical reel-based actuation [50]. Pneumatic actuation

is problematic for our mobile setup as it requires a tube connected to a pump or pressure tank to supply air. The deployable structure and mechanical reel-based actuation affords a much higher extension ratio, but is limited in its robustness and load-bearing capability. The mechanical scissor structure is appropriate for our purpose because it is inexpensive (compact drying rack: \$ 15, linear actuators: \$ 35 x 2) and lightweight (2kg). Existing warehouse robots such as Kiva [16] have a limited expandable capability as they are designed for one specific shelf, whereas our mechanical scissor lift can move various objects by leveraging its highly expandable structure (4 times expansion ratio). The current actuator height (30 - 100 cm) was chosen to cover a wide range of standard chairs and tables, which measure 30 - 76cm and 48 - 96 cm respectively [55]. The maximum height of the scissor structure itself can be also extended by adding more elements like combining two scissor structures to double the maximum height, with the trade-off with the less stable structure.

### Object Actuation

One advantage of our approach is that the robot need not support the weight of the user. Once the robot places the furniture, it serves as a static object. Thus, when a user sits on or puts weight on it, all of the weight goes to the furniture, instead of the robot, which significantly reduces the possibility of a mechanical breakdown.



Figure 5. Different types of furniture moved by the system.

Although the maximum load for the Roomba is 9 kg, the corner-mounted casters distribute and carry heavier loads. Thus, our robots can lift and carry heavier objects than an unmodified Roomba. The maximum weight the robot can lift and carry is 22 kg. When we put a heavier object than 23 kg, we observed the scissor structure started to break. The strength of the scissor structure suffices to lift lightweight chairs and tables, such as the IKEA honeycomb furniture used in our prototypes. The weight of the furniture we have tested (depicted in Figure 5) ranges from 3.5 to 11.2 kg. For heavier objects, multiple robots can also coordinate to lift a piece together if there is sufficient space under the furniture. Also, with a more robust scissor structure, we can carry heavier objects, as we observed the Roomba base itself (with the corner-mounted casters) can carry up to 30 kg load.

This approach also increases flexibility because different types of furniture can be actuated with the height-adjustable scissor lift. For example, Figure 5 illustrates various static props that the RoomShift robot can actuate. These objects include

furniture such as a desk, a long table, different chairs, and a side table. Note that due to the robot's minimum collapsed size, objects must have at least 30 cm clearance below them, and enough horizontal space to fit the robot. A designer can also create custom props for specific applications, for instance, the styrofoam wall mounted to a side table seen in Figure 5.

### Tracking System

To accurately control the RoomShift robots, we require precise motion tracking that can cover the play area in which a user walks. We use an optical tracking system with 20 IR cameras (Qualisys Miquis 5) that can track objects in a 10 m × 10 m space. Figure 6 depicts the space and mounted cameras on the ceiling (left) and tracking software (right). The system tracks six degrees of freedom (DOF) position of the objects with retro-reflective spherical markers at 60 FPS frame rate.

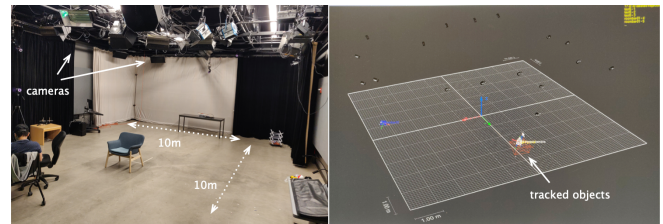


Figure 6. Photo of tracked space and screenshot of tracking software.

To track each robot, we attached five 30 mm spherical retro-reflective markers to the bars of the scissor structure (Figure 7). We attached markers to a pair of parallel bars, so that the markers' relative positions remain constant regardless of the height of the scissor lift. We can also estimate the height of the marker by measuring the orientation of the marker pattern (the pink plane surface depicted in Figure 7).

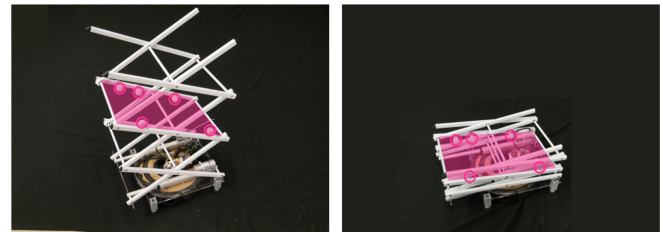


Figure 7. Retro-reflective markers mounted to parallel lift bars, highlighted in pink.

All physical props have retro-reflective markers attached, so that the system can capture and track their positions and orientations, and plan the paths for the robots to pick them up and avoid collisions. They also enable the system to track the robots while moving objects: when markers attached to the robot are hidden beneath an object it is carrying, the system can still reliably track the robot using the object markers as a proxy for the hidden robot. Our motion capture system uses 20 cameras, so even when large objects (e.g., walls) occlude furniture trackers from the cameras in one direction, opposing cameras maintain tracking. We tested with multiple walls, and when the furniture is enclosed from two sides, the system starts to lose tracking. However, in testing, we did not encounter this scenario frequently.

**Control System**

To control the robots’ movements, we use a simple path planning algorithm. The input is 1) the current positions of the robots, 2) the positions of obstacles (e.g., furniture, other robots, and users), and 3) the target locations. The algorithm outputs the goal of each robot at the next time step. The system continuously updates the path and drives them to their target locations. The main server continuously tracks the robot positions, calculates their wheel speeds, and sends commands at 30 Hz over WiFi.

Here, we describe the control algorithm in more detail. The first step of the algorithm is to assign the target position for each robot. The system assigns the optimal target for each robot by solving an assignment problem. We first constructs a distance matrix  $D = (d_{i,j})$  where  $d_{i,j}$  represents the distance between the robot  $i$  and target  $j$ . The system computes the optimal combination by applying the Hungarian algorithm to the distance matrix. Given the assigned target, the system computes the path based on simple Reciprocal Velocity Obstacles (RVO) algorithm [52]. This algorithm also handles collision avoidance with users, other robots, and obstacles.

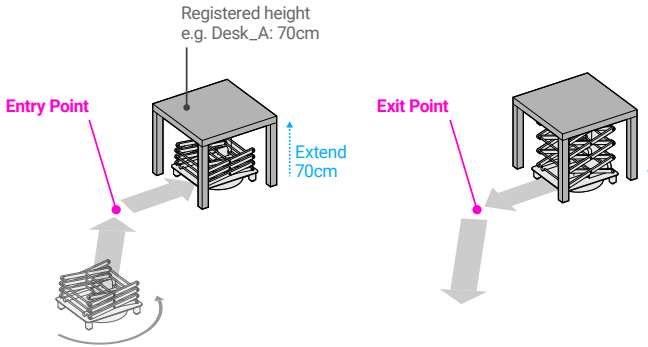


Figure 8. The system first navigates the robot to a user-defined entry point to avoid the collision with the legs of furniture.

With the RVO algorithm, we get the vector the robot should move at each time step. Based on this, the control system determines the speed of the left and right wheels (ranged from -255 to 255) with the PID control. The error function of the PID control is  $e(t) = da_t$ , where  $da_t$  is the difference between the current orientation and the target vector. The system moves the robot by minimizing this  $e(t)$  with the standard PID control function which is  $u(t) = K_p e(t) + K_i \int_0^t e(t') dt' + K_d de(t)/dt$ , where  $K_x$  is the gain parameter of each factor. We determine the speed of left and right wheels as  $A(dd_t) * (1 + u(t))$  and  $A(dd_t) * (1 - u(t))$  respectively.  $A(dd_t)$  is the linear function of the difference in distance at time  $t$ , with a certain minimum and maximum threshold, so that when the distance is smaller than a certain threshold, the robot stops because  $A(dd_t)$  becomes zero.

To pick up and place these, the robot follows a predefined sequence, approaching the object from an angle where it will not collide with the object’s legs. To avoid the collision with the legs of furniture, each object has a user-defined entry and exit point (Figure 8). We also register the height of target furniture before the system starts (e.g., 70 cm for Table\_A, 40

cm for Chair\_B), so that it can extend the scissor lift to certain target height. We could also put a simple sensor on top of the scissor structure to make it a closed-loop system.

**Electronics Implementation**

Figure 9 illustrates the schematic of RoomShift’s circuit. Each robot is controlled with an ESP8266 microcontroller chip (Wemos D1 mini), powered by the Roomba through a voltage regulator.

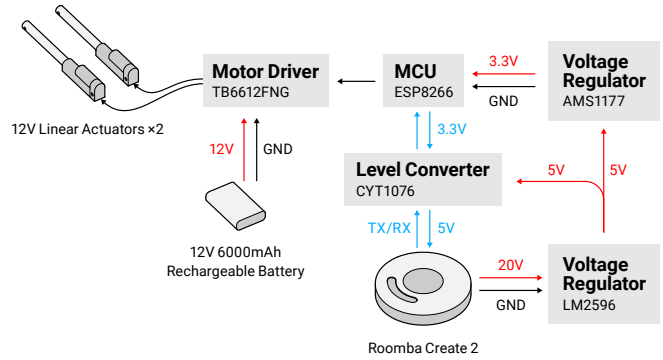


Figure 9. Hardware schematic of the robot.

The power source of the microcontroller is the Roomba’s internal battery which supplies 14-20V. A voltage regulator (LM2596) first steps this down to 5V. The 5V power is supplied to another voltage regulator (AMS1177), which supplies 3.3V to the ESP8266 microcontroller and logic level converter (CYT1076). The logic level converter converts the voltage for serial communication between the microcontroller (3.3V) and Roomba (5V). The microcontroller receives commands over WiFi and controls the left and right wheels of the Roomba using a PWM signal. The microcontroller also operates two linear actuators using a dual motor driver (TB6612FNG). The Roomba’s internal battery is insufficient to supply the current for the linear actuators (600-800mA for average, 1.5-2A for peak current), so we use an external portable rechargeable battery (12V 6000mAh) to power the actuators.

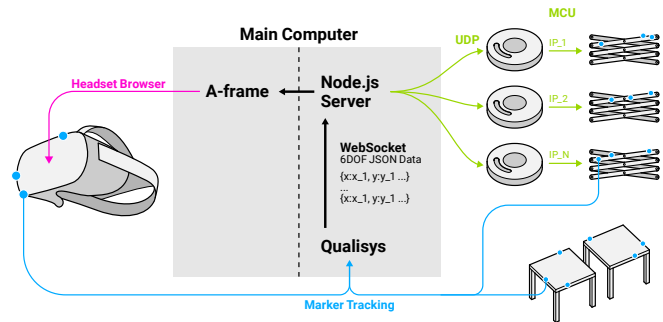


Figure 10. The communication software.

**Software Implementation**

Figure 10 illustrates the architecture of the RoomShift software. The main computer runs a Node.js server and the Qualisys tracking software. The 6DOF tracking data that the Qualisys tracking system captures is streamed to the Node.js server through the WebSocket protocol.

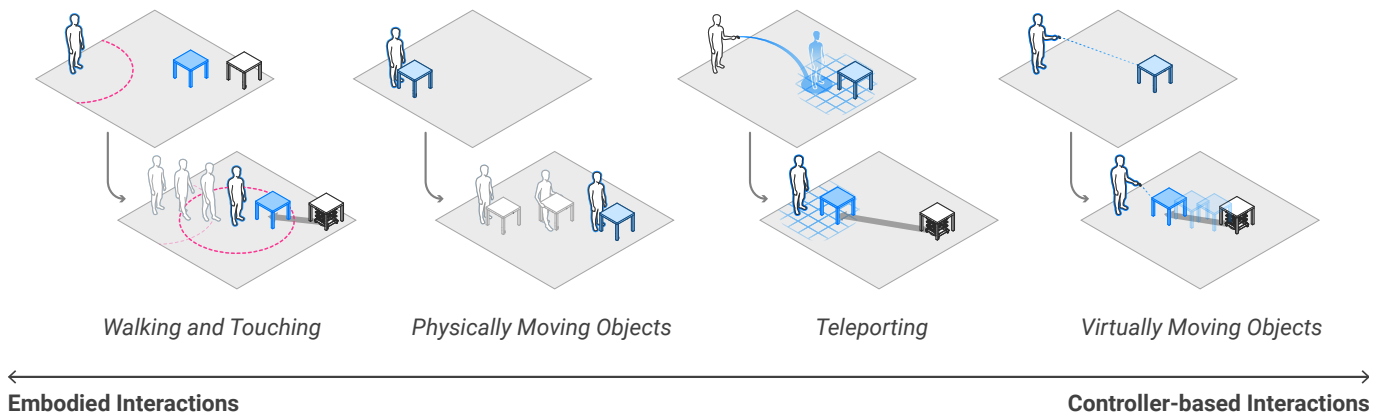


Figure 11. The interaction design space of RoomShift.

Based on the tracking data, a web browser client renders the VR scene with A-Frame. The user experiences the VR scene using an Oculus Go head mounted display and its built-in VR browser. We synchronize the desktop computer and the Oculus Go browser with real-time communication through WebSocket. When the virtual scene changes, the system moves the robots to dynamically reconfigure the physical scene. First, the system computes the types of props and each target position based on the relative position from the user. Once the target position of the physical props is determined, the Node.js server sends commands to each robot over WiFi.

### Preliminary System Evaluation

We conducted a preliminary evaluation to understand how the system would function and gauge user experiences in VR enabled by RoomShift. We posit that the system can reconfigure a simple virtual scene to the point where participants' haptic experiences match those of a static physical environment, and that limitations like robot speed, and physical movement would not diminish the sense of immersion.

We recruited five participants (male: 4, female: 1; between 23–34 years old) from our local institution. All participants had prior experience with VR. In a within-subjects counterbalanced design, participants interacted with physical chairs in a VR scene in two conditions: 1) with physical chairs moved by robots and 2) with static physical chairs. We rendered a virtual  $7\text{ m} \times 7\text{ m}$  room with 4 chairs in VR, matched by a physical area of the same size. We kept a 1.5 m margin around the play area for additional space that robots can move in or stay. Participants wore a head-mounted display (Oculus Go) and noise canceling headphones playing white noise. They were asked to locate a highlighted red chair in the VR scene and sit on it. When sitting, another chair in the room would turn red and the participant was then asked to walk over and sit on the new highlighted chair. This task was repeated eight times in two different conditions. In the first condition, the physical play area contained 4 static physical chairs, which matched the positions of the virtual chairs. In the second condition, RoomShift robots moved two chairs to simulate four chairs. While the user was sitting on a highlighted red chair, a robot moved another chair to the next target location. After the experiment, we asked participants which condition was more

realistic (or if they felt the same), and to provide qualitative feedback on their experiences.

### Results and Feedback

All participants answered that they perceived the realism of the two conditions as the same. This indicates that RoomShift can simulate an environment as intended. While three participants noted they noticed the noise of the moving robots through the headphones, they stated that it did not distract from the realism of the scene. The program did not render a representation of the user's body in VR, and four participants reported that this diminished the perceived realism and their confidence in touching the chairs and sitting down rapidly in both conditions.

Participants provided overall positive feedback in their qualitative responses to RoomShift. Participants were particularly enthusiastic about possibilities for Virtual Tourism and Architectural Design: *"Familiarizing yourself with a place you haven't had a chance to be yet. Maybe someone that is wary of new places, like an agoraphobe, could test out somewhere new as a sort of practice before going there and getting outside of their comfort zone. (P4)"* Users also saw potential to use RoomShift for architectural design: *"Having a model of a proposed space with the ability to interact with furniture (P1)."* Based on the feedback, we explore more in-depth interactions, specifically focusing on these scenarios.

## INTERACTION WITH ROOMSHIFT

### Target Application Domain

In this paper, we specifically focus on architectural application scenarios, such as rendering physical room interiors for virtual real estate tours and collaborative architectural design, two increasingly common application areas for VR [22]. Virtual real estate tours reduce the time and cost compared to on-site viewings, but currently lack the bodily experience of being able to touch surfaces and sit down. In architectural design, VR aids the communication between architects and clients, where proposed designs can be experienced, discussed and modified before building them. We are motivated by how RoomShift can enable people with various physical abilities to experience, test and co-design these environments with their bodies. Most of the elements in these applications can be covered with a finite set of furniture and props (e.g., chairs,

desks, and walls). We discuss some of the basic interactions to support these applications.

### Interaction Design Space

To support these scenarios, we propose four types of basic interactions RoomShift can support, with the spectrum between embodied interactions and controller-based interactions, as illustrated in Figure 11. Embodied interactions refer to interaction with virtual scenes through physical movements and manipulation. The user can implicitly interact with the system by walking around or explicitly interact with the virtual scene by physically moving furniture. On the other hand, the user can also interact with the virtual scene with controller-based gestural interactions. An example is when the user relocates a distant piece of furniture or remove the wall in the room. The user can also virtually teleport their location to navigate through space. We describe each interaction technique in the context of architectural design and virtual real estate tours.

#### Experiencing Architectural Spaces: Walking and Touching

The most basic interaction is to render an architectural space that the user can walk around in and touch. To render the haptic proxies for a large space would require a large number of physical props and robots. On the other hand, the user's immediate physical reach is usually smaller than the entire virtual scene (e.g., 1.5 m radius). Therefore, the system only places haptic props within the user's immediate proximity. As the user walks around the space, the robots move the props to maintain the illusion of a larger number of objects. In this way, a small number of robots with a finite set of physical props can suffice to provide haptics for the scene as the system does not need to physically render the entire environment (Figure 1).

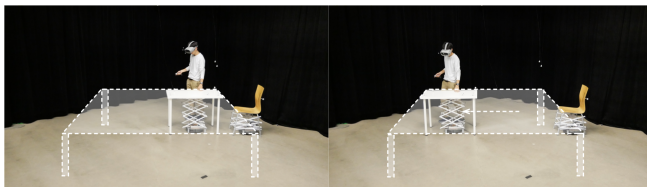


Figure 12. Simulating a larger table by moving a smaller surface.

In addition, the system can mimic larger objects with a single moving robot. For example, when the user is interacting with a large table, either new physical table segments can be added or a single robot can continually move the current table according to the user's position to simulate touching a larger one. This way, a limited number of robots and furniture can simulate large objects (Figure 12). We also employ this technique for rendering larger wall segments, where the robot moves, carrying the proxy, as the user walks along the wall, similar to a technique proposed in PhyShare [18].

#### Architectural Co-Design: Physically Moving Furniture

VR can support teams of architects, designers and their clients to experience and discuss architectural and interior designs. For example, Dollhouse VR [22] proposes such a possibility for the collaborative design of the home and office spaces, where a user experiences space in VR, while a designer views the layout on a desktop computer remotely and changes the design during the discussion. RoomShift system improves the

immersion of this collaborative design process by enabling whole-body interactions with furniture. Suppose a situation where a designer and a client are remotely co-designing a new office space at two separated RoomShift systems. When the designer reconfigures the furniture or space, the robots in the remote location can move the physical objects in real-time to render the designer's change. In this way, two remote physical environments can synchronize with a single virtual space. This aids co-design where the client can touch, feel, walk around, and modify the design in VR.

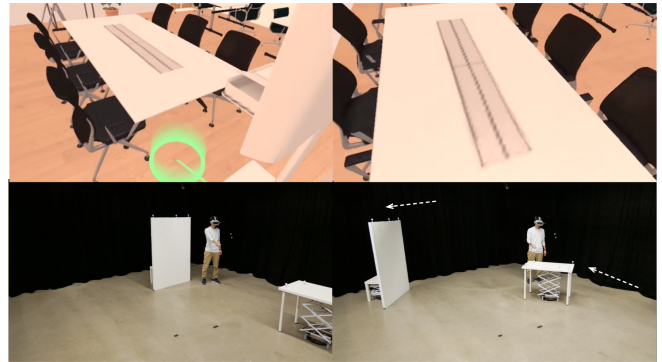


Figure 13. When teleporting, the robots move furniture to match the new scene position.

#### Navigating Large Spaces: Teleporting in VR

The physical play area of a VR setup is often much smaller than the virtual scene. Teleportation is a common navigation technique that enables the user to point with a controller to a distant location in the scene and instantly move there [33]. RoomShift supports teleportation by reconfiguring the room layout to match the new view location (Figure 13). When the user teleports to a new location in the VR scene, the system calculates the positions of the virtual objects relative to the new location and moves the furniture and robots in and out of the play area to enable a fast scene reconfiguration and to avoid collisions with the user and each other.

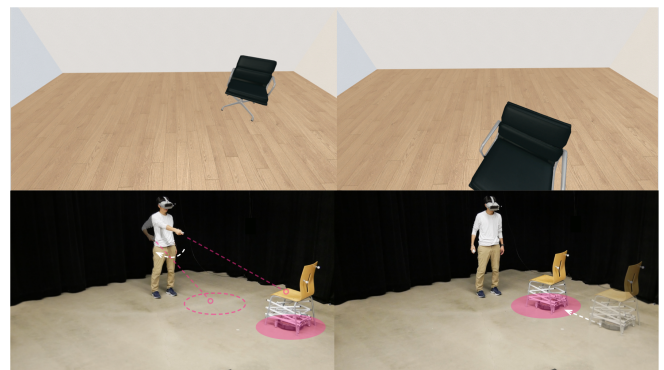


Figure 14. Pointing and moving with a gesture.

#### Virtual Scene Editing: Virtually Moving Furniture

RoomShift system also supports scene editing within VR. The virtual scene layout editing is similar to standard VR interactions and includes functionality like adding, removing, moving, resizing, rotating virtual building elements and furniture with a VR controller or a GUI. For example, the user can

point the controller at a virtual object and move it to a target location. The robot then updates the virtual object position (Figure 14).

## DISCUSSION

The reason for using 9 robots is based on our estimate of the maximum number of furniture and props that fit within close proximity (3-4 m range) of a single user in a standard VR scene - similar to how TurkDeck uses ten human actuators [10]. By utilizing a small swarm of robots, we could also explore unique approaches to task coordination. A major limitation of past robotic graphics systems, where a single robot simulates multiple objects, is that the robot often moves too slowly to match fast graphics transformations. Swarm robots can coordinate and hand off tasks between each other to address this speed limitation. Two robots can, for instance, move two separate chairs into place to simulate a single chair rapidly moving across the room when teleporting. As the robots are not permanently attached to the scene objects, their role division can change depending on the virtual content. A single robot may at one point represent multiple objects, then switch roles to carry a copy of an object. Similar to other swarm robot systems, robots may also hand off tasks to recharge their batteries, providing a longer VR experience.

Utilizing props beyond furniture to represent a wider range of virtual scenes is also possible. For example, a combination of multiple pieces of furniture can represent different objects, such as a staircase simulated from stools of different heights. Moreover, the robots can move generic building blocks, custom cardboard props, or mannequins, which are useful for games such as Minecraft, or interaction design for human-robot communication. Beyond constructing static scenes, the system can also simulate dynamic objects and environments, such as height-changing desks, robotic beds, or transforming walls. This setup aids HCI researchers and practitioners in designing interactions for such emerging technologies.

## LIMITATIONS AND FUTURE WORK

There are many limitations of the current prototype system and opportunities for future work. First, due to the mechanical scissor lift design, the system can only actuate objects with a minimum 30 cm clearance underneath. We tried different actuation mechanisms such as pushing or dragging the object, but this is only possible with lightweight objects (e.g., the robot was not able to drag a sofa). A more compact mechanical structure could alleviate this limitation. Second, the drivetrain of the robot is not omni-directional, therefore the orientation of the robot matters when picking up furniture, as it will influence in which direction the robot can move it without rotating it.

Currently, our mechanical scissor structure and wheeled-based robot are not robust enough to carry heavy objects and humans. However, by using a more robust scissor structure and stronger actuators, these shape-changing robots could also actuate furniture while the user is sitting or standing on. If this is possible, there are many more interesting application scenarios, such as simulating dynamic floor or terrain (e.g., CirculaFloor [24] or LevelUps [42]) or simulating dynamic objects. In addition, such robots may be able to simulate the

environment itself — instead of bringing an existing chair, the robot changes its height to render a chair prop itself. This can introduce much more flexibility and dynamism for haptic sensations. We look forward to the future work which will investigate and demonstrate a more robust shape-changing structure.

The feedback from our preliminary evaluation is encouraging, but a larger scale study would provide further insights into the appropriate utilization of swarm robots for VR haptics.

In this paper, we focused on furniture rearrangement with wheel-based robots, but there are some intriguing alternative approaches. For example, recent research advances the capability of swarm construction for 3D architectures [27, 54]. Alternatively, we could also leverage different types of robots; for example, a swarm of drones [4] or cable-based robots [41] can provide mid-air haptic sensations. For future work, we are interested in exploring other types of swarm robots to enable more flexible spatial haptic interfaces.

Finally, while this paper entirely focused on haptics for a VR environment, RoomShift also has the potential for broader application space for dynamic office or home environments. For example, these distributed robots can help the automation of home, labs, store, and public space by automatically reconfigure the spatial elements based on the situation (e.g., set up the meeting space, desk, and chair based on the calendar event, and clean up and reconfigure the space after the meeting.) RoomShift's capability of actuating existing objects is particularly interesting for this application space. There are several technical challenges that would need to be addressed for such applications, including tracking methods (e.g., inside-out vs. outside-in tracking), interaction modalities (e.g., voice, gesture), and path planning (e.g., collision avoidance with multiple people). For example, the tracking system currently requires a dedicated setup, which is difficult to deploy outside a laboratory. We are interested in reducing this required hardware setup, for example, using a single depth camera and AR markers to make the system more accessible. Future work will investigate how to deploy these RoomShift robots into the real-world environment and investigate how these robots can be distributed and embedded in our physical environments and adapt to our everyday life.

## CONCLUSION

This paper introduces RoomShift, a room-scale dynamic haptic environment for virtual reality. The goal of RoomShift is to provide room-scale haptic experiences. To achieve this goal, we propose the new approach of using a swarm of shape-changing robots that relocate existing furniture for reconfigurable physical environments. We described the design and implementation of RoomShift robot which leverages a wheel-based robot and expandable mechanical structure. The user evaluation study with five participants confirms the RoomShift's benefits in providing a realistic and enjoyable experience for the VR environment. We demonstrate applications in virtual tours and architectural design and investigated several interaction techniques for these scenarios.



**ACKNOWLEDGEMENTS**

This research was partially supported by the Nakajima Foundation and the Strategic Information and Communications R&D Promotion Programme (SCOPE) of the Ministry of Internal Affairs and Communications of Japan, and with equipment sponsored by the Engineering Excellence Fund at CU Boulder.

**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*. ACM, 150.
- [2] Parastoo Abtahi, Benoit Landry, Jackie (Junrui) 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 (CHI '19)*. ACM, New York, NY, USA, Article 359, 13 pages. DOI:<http://dx.doi.org/10.1145/3290605.3300589>
- [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] Frederico Augugliaro, Sergei Lupashin, Michael Hamer, Cason Male, Markus Hehn, Mark W Mueller, Jan Sebastian Willmann, Fabio Gramazio, Matthias Kohler, and Raffaello D'Andrea. 2014. The flight assembled architecture installation: Cooperative construction with flying machines. *IEEE Control Systems Magazine* 34, 4 (2014), 46–64.
- [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] Lung-Pan Cheng, Li Chang, Sebastian Marwecki, and Patrick Baudisch. 2018. iTurk: Turning Passive Haptics into Active Haptics by Making Users Reconfigure Props in Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 89.
- [8] Lung-Pan Cheng, Patrick Lühne, Pedro Lopes, Christoph Sterz, and Patrick Baudisch. 2014. Haptic turk: a motion platform based on people. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3463–3472.
- [9] Lung-Pan Cheng, Sebastian Marwecki, and Patrick Baudisch. 2017. Mutual human actuation. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 797–805.
- [10] Lung-Pan Cheng, Thijs Roumen, Hannes Rantzsch, Sven Köhler, Patrick Schmidt, Robert Kovacs, Johannes Jasper, Jonas Kemper, and Patrick Baudisch. 2015. Turkdeck: Physical virtual reality based on people. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 417–426.
- [11] Inrak Choi, Heather Culbertson, Mark R Miller, Alex Olwal, and Sean Follmer. 2017. Gravity: A wearable haptic interface for simulating weight and grasping in virtual reality. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 119–130.
- [12] 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.
- [13] Martin Dekan, F Duchon, and others. 2013. iRobot create used in education. *Journal of Mechanics Engineering and Automation* 3, 4 (2013), 197–202.
- [14] Alexandra Delazio, Ken Nakagaki, Roberta L Klatzky, Scott E Hudson, Jill Fain Lehman, and Alanson P Sample. 2018. Force jacket: Pneumatically-actuated jacket for embodied haptic experiences. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 320.
- [15] Darren Guinness, Annika Muehlbradt, Daniel Szafir, and Shaun K Kane. 2018. The Haptic Video Player: Using Mobile Robots to Create Tangible Video Annotations. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*. ACM, 203–211.
- [16] Eric Guizzo. 2008. Three engineers, hundreds of robots, one warehouse. *IEEE spectrum* 45, 7 (2008), 26–34.
- [17] 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.
- [18] Zhenyi He, Fengyuan Zhu, and Ken Perlin. 2017. Physshare: 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*. ACM, 525.

- [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] 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.
- [22] Hikaru Ibayashi, Yuta Sugiura, Daisuke Sakamoto, Natsuki Miyata, Mitsunori Tada, Takashi Okuma, Takeshi Kurata, Masaaki Mochimaru, and Takeo Igarashi. 2015. Dollhouse vr: a multi-view, multi-user collaborative design workspace with vr technology. In *SIGGRAPH Asia 2015 Emerging Technologies*. ACM, 8.
- [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] Seungwoo Je, Myung Jin Kim, Woojin Lee, Byungjoo Lee, Xing-Dong Yang, Pedro Lopes, and Andrea Bianchi. 2019. Aero-plane: A Handheld Force-Feedback Device that Renders Weight Motion Illusion on a Virtual 2D Plane. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, 763–775.
- [27] Ben Jenett and Kenneth Cheung. 2017. Bill-e: Robotic platform for locomotion and manipulation of lightweight space structures. In *25th AIAA/AHS Adaptive Structures Conference*. 1876.
- [28] F Jensen and S Pellegrino. 2001. *Arm development review of existing technologies*. Technical Report. Cambridge Univ.(United Kingdom). Dept. of Engineering.
- [29] B Kent Joosten. 2007. Preliminary assessment of artificial gravity impacts to deep-space vehicle design. (2007).
- [30] 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.
- [31] 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*. ACM, 688.
- [32] 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.
- [33] Eike Langbehn, Paul Lubos, and Frank Steinicke. 2018. Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking. In *Proceedings of the Virtual Reality International Conference-Laval Virtual*. ACM, 4.
- [34] 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.
- [35] Robert W Lindeman, Yasuyuki Yanagida, Haruo Noma, and Kenichi Hosaka. 2006. Wearable vibrotactile systems for virtual contact and information display. *Virtual Reality* 9, 2-3 (2006), 203–213.
- [36] Pedro Lopes, Alexandra Ion, and Patrick Baudisch. 2015. Impacto: Simulating physical impact by combining tactile stimulation with electrical muscle stimulation. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 11–19.
- [37] 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.
- [38] Kazuki Nagai, Soma Tanoue, Katsuhito Akahane, and Makoto Sato. 2015. Wearable 6-DoF wrist haptic device SPIDAR-W. In *SIGGRAPH Asia 2015 Haptic Media And Contents Design*. ACM, 19.
- [39] Sharif Razzaque, Zachariah Kohn, and Mary C Whitton. 2005. *Redirected walking*. Citeseer.
- [40] Tomoya Sasaki, Richard Sahala Hartanto, Kao-Hua Liu, Keitarou Tsuchiya, Atsushi Hiyama, and Masahiko Inami. 2018. Leviopole: mid-air haptic interactions using multirotor. In *ACM SIGGRAPH 2018 Emerging Technologies*. ACM, 12.
- [41] Makoto Sato. 2002. Development of string-based force display: SPIDAR. In *8th international conference on virtual systems and multimedia*. Citeseer.
- [42] Dominik Schmidt, Rob Kovacs, Vikram Mehta, Udayan Umaphathi, Sven Köhler, Lung-Pan Cheng, and Patrick Baudisch. 2015. Level-ups: Motorized stilts that simulate stair steps in virtual reality. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2157–2160.

- [43] Jotaro Shigeyama, Takeru Hashimoto, Shigeo Yoshida, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. 2019. Transcalibur: A Weight Shifting Virtual Reality Controller for 2D Shape Rendering based on Computational Perception Model. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 11.
- [44] David Sirkin, Brian Mok, Stephen Yang, and Wendy Ju. 2015. Mechanical ottoman: how robotic furniture offers and withdraws support. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 11–18.
- [45] 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.
- [46] 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.
- [47] 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*. ACM, 190–199.
- [48] 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.
- [49] Kazuki Takashima, Takafumi Oyama, Yusuke Asari, Ehud Sharlin, Saul Greenberg, and Yoshifumi Kitamura. 2016. Study and design of a shape-shifting wall display. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, 796–806.
- [50] Shohei Takei, Makoto Iida, and Takeshi Naemura. 2011. Kinereels: extension actuators for dynamic 3d shape. In *ACM SIGGRAPH 2011 Posters*. ACM, 84.
- [51] 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.
- [52] 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.
- [53] 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.
- [54] Justin Werfel, Kirstin Petersen, and Radhika Nagpal. 2014. Designing collective behavior in a termite-inspired robot construction team. *Science* 343, 6172 (2014), 754–758.
- [55] Highland Woodworking. 2019. Standard Furniture Dimensions. (2019). <http://highlandwoodworking.com/library/furnituredimensions.pdf>
- [56] Peter R Wurman, Raffaello D’Andrea, and Mick Mountz. 2008. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI magazine* 29, 1 (2008), 9–9.
- [57] 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.