ThrowIO: Actuated TUIs that Facilitate "Throwing and Catching" Spatial Interaction with Overhanging Mobile Wheeled Robots

Ting-Han Lin
University of Chicago
Chicago, Illinois, USA
tinghan@uchicago.edu

Willa Yunqi Yang
University of Chicago
Chicago, Illinois, USA
yunqiy@uchicago.edu

Ken Nakagaki
University of Chicago
Chicago, Illinois, USA
knakagaki@uchicago.edu

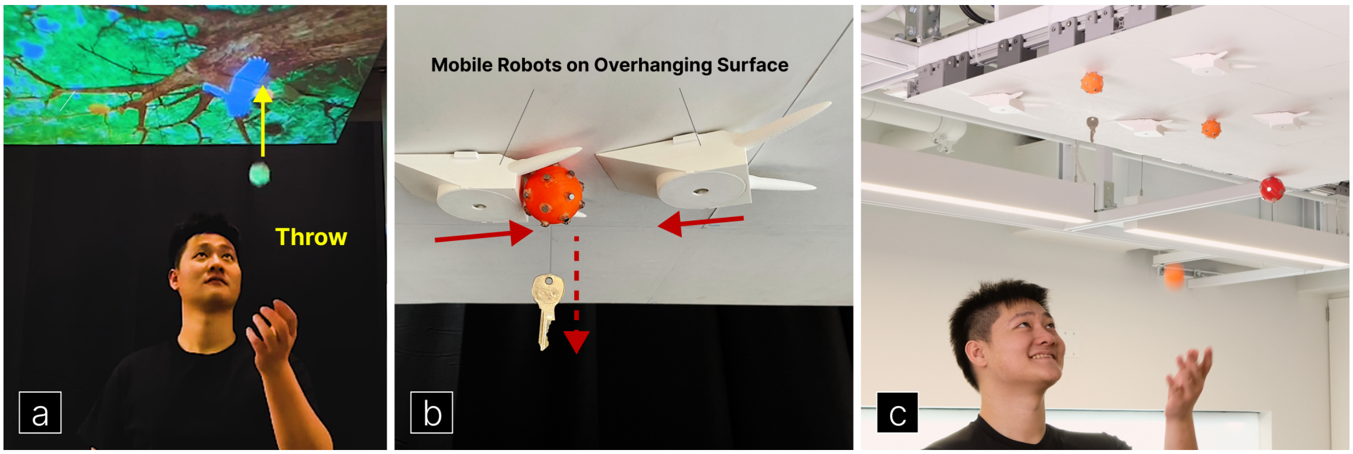

Figure 1: ThrowIO is an actuated tangible user interface that facilitates throwing and catching spatial interactions. (a) A user throws an object to stick to an overhanging surface where (b) two wheeled robots move on the surface to push and drop the thrown object (c) for the users to catch.

ABSTRACT

We introduce ThrowIO, a novel style of actuated tangible user interface that facilitates throwing and catching spatial interaction powered by mobile wheeled robots on overhanging surfaces. In our approach, users throw and stick objects that are embedded with magnets to an overhanging ferromagnetic surface where wheeled robots can move and drop them at desired locations, allowing users to catch them. The thrown objects are tracked with an RGBD camera system to perform closed-loop robotic manipulations. By computationally facilitating throwing and catching interaction, our approach can be applied in many applications including kinesthetic learning, gaming, immersive haptic experience, ceiling storage, and communication. We demonstrate the applications with a proof-of-concept system enabled by wheeled robots, ceiling hardware design, and software control. Overall, ThrowIO opens up novel spatial, dynamic, and tangible interaction for users via overhanging robots, which has great potential to be integrated into our everyday space.

CCS CONCEPTS

• **Human-centered computing** → **Interaction devices**.

KEYWORDS

Actuated Tangible User Interface, Human-Robot Interaction, Human-Computer Interaction, Swarm User Interface, Spatial Interaction

ACM Reference Format:
Ting-Han Lin, Willa Yunqi Yang, and Ken Nakagaki. 2023. ThrowIO: Actuated TUIs that Facilitate "Throwing and Catching" Spatial Interaction with Overhanging Mobile Wheeled Robots. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3544548.3581267

ACM ISBN 978-1-4503-9421-5/23/04…$15.00
https://doi.org/10.1145/3544548.3581267

1 INTRODUCTION

As computers and interactive technologies become more ubiquitous and accessible in our everyday environments, designing spatial user experience – an approach for augmenting everyday space with digital technology – has been of great interest in the Human-Computer Interaction (HCI) community and the commercial User Interface industry [19, 31, 32, 42, 52]. Some of these examples add spatial haptic feedback for Mixed Reality (AR/VR) applications [2, 20, 44, 45, 61]. Others have sought to enable tangible interactions in mid-air by levitating or suspending objects in the space [13, 29, 40, 47]. These types of user interface technologies are developed to provide a sense of touch and tangibility in the 3D space.

While many of these approaches focus on providing a sense of touch by controlling the position of objects in mid-air, they have not looked into interaction and affordance design opportunities for one of the common forms of spatial user interactions with tangibles: *throwing* and *catching*. Throwing and catching are fundamental 3D spatial interactions with physical objects in our daily activities [49]. They are natural and intuitive ways of passing and transferring objects efficiently, for example, when throwing a can into a trash bin from a distance [65]. When learning kinesthetic motor skills, people practice throwing and catching balls in juggling [8]. In sports and entertainment, players in ball sports also intensively perform throwing and catching, such as in basketball, baseball, handball, water polo, and many more [18, 30, 64]. Beyond motor skill learning, throwing and catching balls (or activities related to "catch") is also a typical intimate activity between a parent and a child to develop their relationship and bonding in certain cultures [27]. Therefore, throwing and catching are important actions for motor skills, immersive experiences, spatial interactions, entertainment, etc, which could all be assisted and augmented through digital technologies.

Prior research has partially explored the development of throwing and catching robots in object manipulation contexts [1, 24, 25, 43, 46], but these systems require intrusive installations to the existing physical environment and complicated fast reaction control of the robots in the 3D space, and very few of them involved users in the spatial interactions with robots. Existing VR haptic controllers and devices can help users get a sense of grasp on throwing and catching balls while juggling in virtual reality [3, 9, 15, 26, 35]. Yet, most work on throwing and catching in VR lacks tangible action and the experience of "actually" throwing and catching objects, which are crucial for motor-skill learning and immersive experience.

We present ThrowIO, an actuated tangible user interface for spatial interaction that facilitates throwing and catching actions. Our approach allows users to throw and stick objects to an overhanging surface and catch them once they are dropped by mobile wheeled robots from the surface. ThrowIO tracks the position of a thrown object through a RGBD camera system and directs the robots to perform actions, such as pushing the object to a 2D arbitrary position on the surface and dropping it at an arbitrary timing. With this capability, ThrowIO opens up a wide range of applications for kinesthetic learning, gaming, immersive haptic experience, ceiling storage, and communication. Our approach is unique in that it features throwing and catching *"actual" objects in our living environment*. It is also *relatively easy to install and control* because of the usage of mobile robots and the straightforward installation process of overhanging surfaces. Additionally, it is *able to be integrated into everyday physical spaces* due to its simple installation method.

In our paper, we first introduce the concept and design space of ThrowIO, which overviews our general approach and interaction capability. We then discuss our implementation which is a proof-of-concept prototype of ThrowIO based on off-the-shelf wheeled robots, toio[1]. To demonstrate the novel interactive utility of ThrowIO, we introduce its applications developed with our prototype system. We review the quantitative and qualitative results from a user study (n = 16) on the general usability of our proof-of-concept prototype. We also discuss limitations and future work to share the challenges and opportunities of our approach with the HCI community. This work introduces the community to novel opportunities for enriching spatial interaction by incorporating throwing and catching actions via on-ceiling mobile robots.

Our list of contributions includes:

- introduction of a novel approach in facilitating vertical throwing and catching powered by wheeled robots on an overhanging surface
- general approach and design space of ThrowIO that overviews the design properties and interactivity
- proof-of-concept implementation of ThrowIO, using off-the-shelf toio robots and computer vision tracking
- application demonstration of ThrowIO to be used in a wide range of utilities

2 RELATED WORK

The research contributions of ThrowIO build upon prior research in areas of VR haptics for throwing and catching, robotic juggling, swarm user interfaces, and on-ceiling robots.

2.1 Virtually Emulating Throwing and Catching Interaction (VR Haptics)

Learning complex spatial skills such as throwing and catching balls at a rapid pace can be difficult. Hence, many researchers in the HCI community are interested in investigating how to help people learn and acquire these spatial skills with technology [3, 5, 33, 57]. Some studies in the past utilized Augmented Reality (AR) to give users visual feedback on a thrown ball's trajectory [17, 33]; other past work employed Virtual Reality (VR) with haptic controllers to help users learn spatial skills by emulating throwing and catching in a virtual environment [3, 9, 15, 26, 35]. In those VR environments, users can throw and catch a virtual ball at a lowered gravity with VR controllers. When the virtual ball drops into users' hands, Adolf et al., for example, gave users a vibration from the controller's haptic actuators indicating the ball's dropping force on the palm [3]. However, since users are not throwing and catching any real objects in these VR environments, it can be difficult to learn skills (e.g., juggling) or be fully immersed in the physical interactions. Even though Pan et al. proposed an approach to integrate the physical action of tossing a real ball into VR with visualizations of the ball's trajectory [41], the interaction of throwing and catching is still dependent on visualizations in a virtual space. To address these limitations and utilize throwing and catching in our tangible reality, ThrowIO offers real-world dynamic responses in throwing and catching objects in our living environment. Our approach can also be easily augmented with digital displays to show a ball's trajectory and allow users to use a physical object to interact with the virtual screen via throwing and catching actions.

2.2 Robotic Juggling

Previous work has studied ball-juggling in robotics to understand the motions of moving objects and movements of the human body.

[1] https://www.sony.com/en/SonyInfo/design/stories/toio/

Most of the juggling studies that involve robots have focused on designing autonomous robots that can perform basic juggling movements [7]. With depth camera inputs, Aboaf et al. designed a robot that could juggle one tennis ball by hitting it upwards with a paddle [1]. By using a high acceleration reinforcement learning system, Ploeger et al. enabled a robot to juggle two balls [43]. In those studies, humans rarely take part in the implemented juggling system with the robots.

To examine the relationship between humans and juggling robots, Kober et al. conducted a study where a user would physically move a robot's arm to hit moving objects in order to train the robot to catch and throw them [25]. In another study, Kober et al. built a throwing and catching system where human participants could juggle three balls with an animatronic humanoid robot [24]. Even though the humanoid robot has limited degrees of freedom in catching and throwing the ball, this research has demonstrated that humans and robots can spatially interact with each other by throwing and catching objects.

Our configuration uniquely employs swarm robots on overhanging surfaces, which minimizes the complexity of control and hardware to facilitate throwing and catching interaction. We also believe our hardware setup (the mobile robots) is the smallest among the above robots, and the interaction area is technically scalable as the robots can locomote across a large surface. Also, in contrast to other robotics research, our goal is in developing spatial user interaction oriented to the action of throwing and catching.

2.3 Self-propelled and Swarm User Interfaces

Swarm robots have demonstrated the ability to collectively form shapes or perform tasks [12, 22, 48, 50, 53]. Recently, swarm robots are starting to be seen as interfaces. Swarm User Interfaces (SUIs) have been explored as a different user interface class [16, 28] building on top of the classic Tabletop TUIs [58, 59]. Researchers have started to demonstrate the interaction design opportunities enabled by self-propelled robots' locomotion capabilities and collective operations [28, 38]. The interactivity of such self-propelled interfaces can be further extended by docking a variety of mechanical shells onto the wheeled robots [38]. These add-ons can also be customized to change shape [56], function as fabrication components [14], render haptics for Virtual Reality objects [55], etc. SUI has also been understood together with the environment where it resides such as blending in and being actuated as needed [39].

Most of the prior endeavors, however, are constrained to the tabletop or upward-facing surfaces, which limit the types of spatial interaction that can occur in a larger space. In this paper, we explore the novel interaction opportunity enabled by an overhanging surface, in throwing and catching specifically, with a wide range of applications.

2.4 Robots on the Ceiling or on Walls

We acknowledge prior research that developed robotic systems that can locomote on walls and ceilings [34]. They have employed different adhesion techniques such as aerodynamic and suction cup attraction [6, 51, 62, 62], elastomeric materials adhesion [36, 60], and ferromagnetic systems [21, 63]. Unlike previous work, our approach simply embeds magnets on the robot's body to create adhesion to the ferromagnetic overhanging surfaces.

More importantly, while these on-wall and on-ceiling robotics studies have focused on engineering locomotion techniques, they have not explored much in manipulating objects. Therefore, facilitating objects on the ceiling to move and drop is a unique direction we introduce in our paper.

3 DESIGN SPACE OF THROWIO

In this section, we introduce the basic architecture of ThrowIO and outline its design space as a generalizable approach for spatial user interaction design in "throw-and-catch" (see Figure 2).

3.1 Basic Architecture and Core Interaction

ThrowIO is an interactive system that facilitates throw-and-catch interactions for users. As illustrated in Figure 2, the generalizable basic architecture consists of (1) an overhanging surface, (2) wheeled robots, and (3) thrown objects.

The core interaction of ThrowIO is in *throwing* and *catching*. Unlike many other VR systems that enable haptics for throwing and catching, our system allows for throwing and catching "actual" objects. When objects are thrown to the overhanging surface, it sticks to the surface through magnetic attraction. Then, the catching interaction is facilitated by the wheeled robots on the overhanging surface to move the object to designated positions and drop them, which allow the objects to be caught by the user.

The design of the overhanging surface allows users to throw and catch in a vertical direction. While horizontal throwing and catching are not within the scope of this paper, our current design still allows for a range of angles and applications, as demonstrated later in the paper.

3.2 Overhanging Surfaces

The overhanging surface in ThrowIO can be set at different heights, which can be used for different interaction design purposes. For example, the heights can not only be adjusted for different throwing interactions but also be incorporated into various types of everyday environments. With the height overhead (*ceiling* height), the design of ThrowIO could be embedded in ceiling surfaces of everyday rooms, which allows for throwing objects high above the head. When the surface is mounted at *chest* height, it could allow medium-height throws and catches, which may be rather suitable for juggling-related applications. Additionally, *under-table* height should be able to facilitate objects in an under-table situation where, for example, in a meeting scenario, objects can be pushed under the table and dropped on a person's lap without other people in the meeting noticing.

3.3 Thrown Objects

Thrown objects would have the capability to stick to the overhanging surface. For example, in our implementation, we attached small magnets to the objects. Thrown objects can be considered as balls with varying sizes from the size of a ping pong ball to the size of a tennis ball. Other conventional objects could be potentially used in ThrowIO by attaching them onto the ball such as pens and keys or by enclosing them with a cloth embedded with magnets to provide

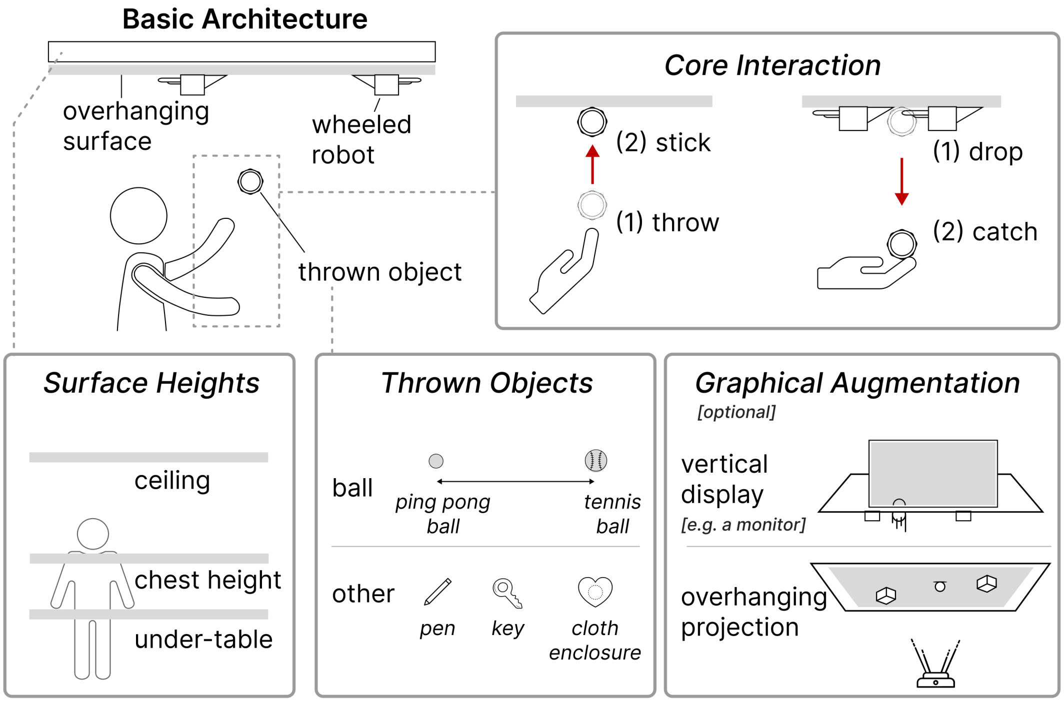

Figure 2: An overview of the basic architecture and design space of ThrowIO.

a variety of shapes based on the specific scenarios. While we have mostly demonstrated ThrowIO with ping pong ball-sized objects, a variety of objects can potentially be explored.

3.4 Graphical Augmentation

Additionally, in our paper, we explore how we can augment the throwing and catching interaction combined with graphical illustration. For example, when combining with a *vertical display* placed above the overhanging surface, the thrown object can be virtually transmitted to the digital space, and vice versa for dropping (similar to the technique explored in [39]). With *overhanging projection*, graphical images can be directly overlaid on the overhanging surface as illustrated in Figure 2.

4 IMPLEMENTATION

We introduce the implementation of our proof-of-concept prototype of ThrowIO. The implementation is based on four main components: two toio robots with custom-designed mechanical shells, overhanging ferromagnetic toio mats, a thrown object, and a Microsoft Kinect v1 camera. The overall implementation is shown in Figure 3. In our implementation, users can toss a magnetic object to an overhanging ferromagnetic toio mat; the overhanging robots will then travel to the thrown object's location and drop it back to the users. Below, we detail the implementation of each component ranging from the hardware design to the software control. We have also made our hardware design and code open-source [2].

4.1 Hardware Design

4.1.1 Mobile Wheeled Robots with Toio and Mechanical Shells. Our wheeled robots consist of toio robots (off-the-shelf mobile two-wheeled robots, developed by Sony Interactive Entertainment [3]) and custom-designed mechanical shells. A mechanical shell is a concept of add-on attachment for Actuated TUIs [37, 38], and in our prototype, we have designed a specific shell for performing *push* and *drop* functions on the overhanging surfaces. Toio robots are equipped with built-in cameras on their bottom side to identify their locations on toio mats (micro-patterned mat, embedded with absolute location information [4]).

The robot design, a combination of a toio robot and a mechanical shell, is shown in Figure 4. The robot, overall, has 4 neodymium magnets embedded (two 6 mm x 1.5 mm disk-shaped ones on the toio body and two 10 mm x 1.5 mm disk-shaped ones on the mechanical shell). These magnets allow the robot to travel on an overhanging ferromagnetic surface. The approach of adding a magnet to increase the actuation capability of the wheeled robot interface is inspired by prior works [23, 38].

As the robots serve to perform *push* and *drop* operations, we have designed the mechanical shells to have two sides, a *prong* side and a *wedge* side, which allow robots to switch between operations (Figure 4a). *Prongs*, with the diagonal fork-shaped design, can *push* thrown objects on the overhanging structure as shown in Figure 5a. The other side of the robot is a *wedge* which has a slightly angled scoop-like shape, and it can be used to scoop and *drop* thrown

[2]https://github.com/AxLab-UofC/ThrowIO

[3]https://www.sony.com/en/SonyInfo/design/stories/toio/

[4]https://toio.io/news/2020/20200423-1.html

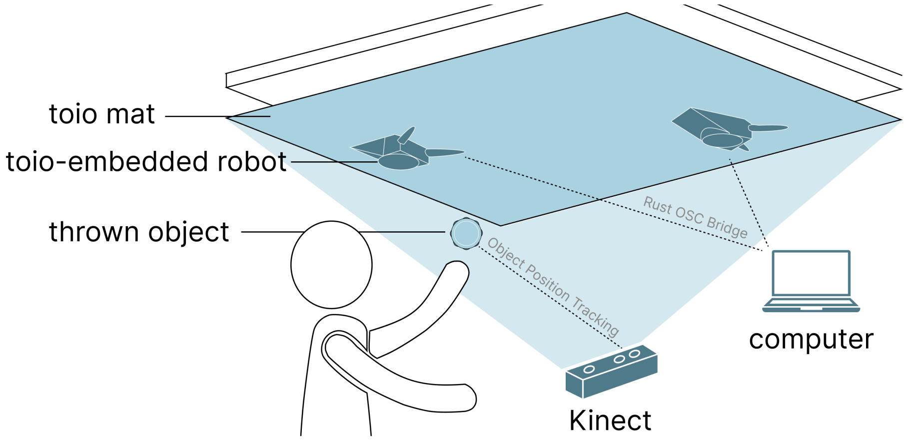

Figure 3: Our implementation of ThrowIO includes an overhanging ferromagnetic toio mat, two toio-embedded robots controlled wirelessly by a central computer, and a thrown object whose location is tracked by a Microsoft Kinect v1 camera placed underneath the toio mat.

objects when combined with another robot pushing (Figure 5b). By combining the *push* and *drop* capabilities, the system of ThrowIO can drop objects at a designated point.

Through our iterative prototypes, we arrive at the current design of the mechanical shell for ThrowIO, which stably performs the required operations (evaluated in sections 4.4.2 & 4.4.3).

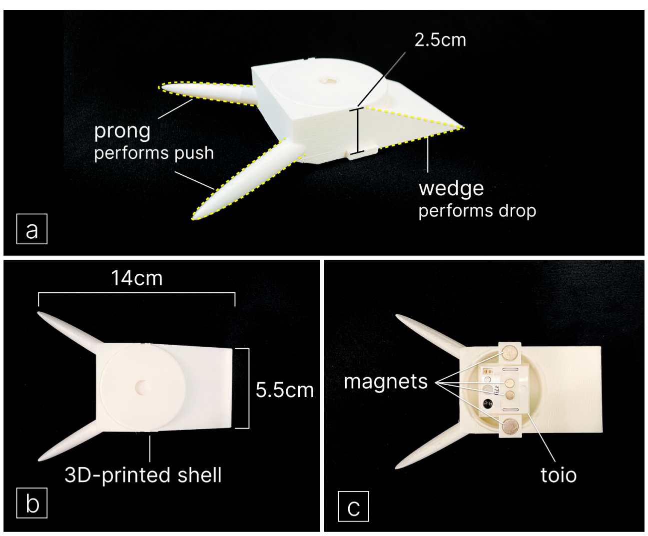

Figure 4: The robot is designed to perform (a) *push* and *drop* operations, enabled by (b) a 3D-printed shell mounted on (c) a toio robot.

4.1.2 Overhanging Ferromagnetic Toio Mats. As for the overhanging surface, we designed a height-adjustable surface for us to easily prototype applications and demonstrations with different heights, as shown in Figure 6.

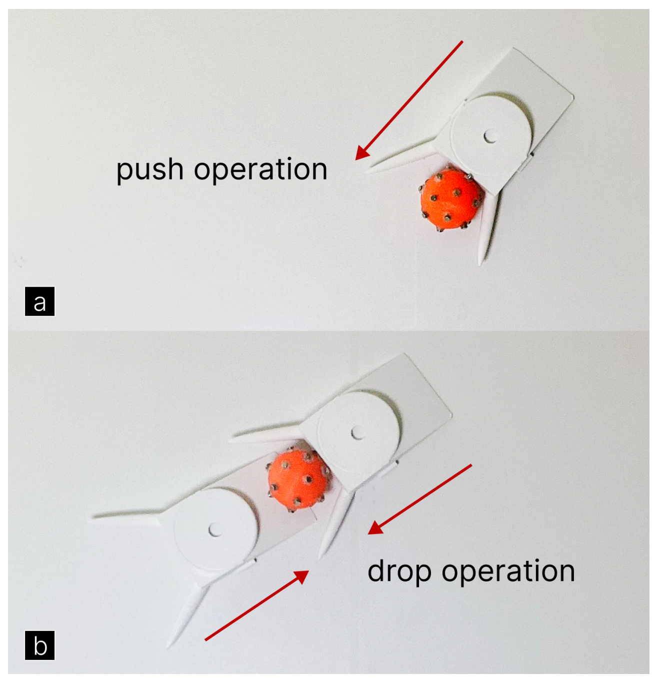

Figure 5: An illustration of (a) the *push* operation and (b) the *drop* operation.

This setup is made using a height-adjustable standing desk where the bottom side of the desk is attached to 4 (2 x 2: 84 cm x 60 cm dimensions) A3-sized toio mats [5]. These toio mats are double-taped to a ferromagnetic metal plate (0.5 mm thickness). The metal mat is fixed underneath the table with 3D-printed mounts embedded with strong magnets to allow easy attachment or removal of the surface

[5]https://toio.io/news/2020/20200423-1.html

under the table. The height of the surface can be adjusted freely from 70 cm to 115 cm. Depending on the application, we also put the entire height-adjustable standing desk on top of another table to prototype the application with an overhead height (see Figure 6a).

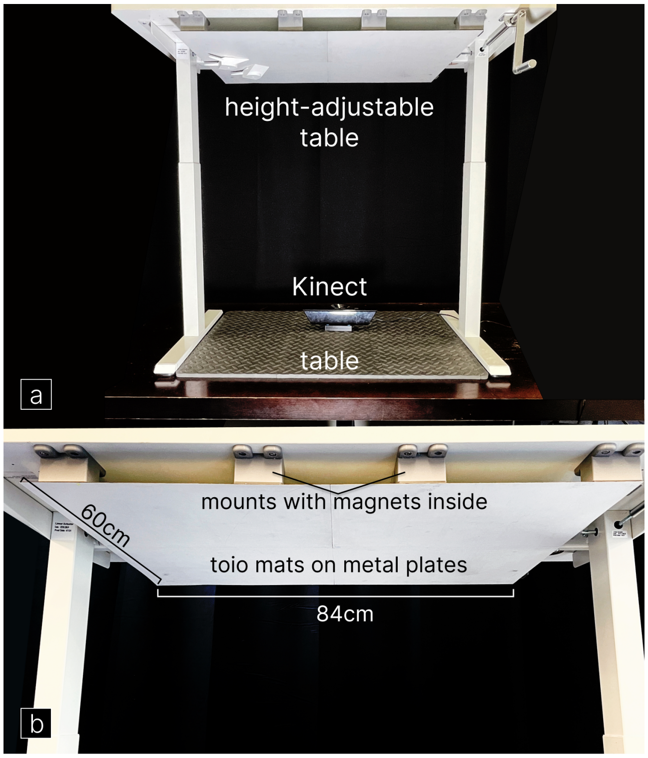

Figure 6: (a) In our setup, we put the height-adjustable table on top of a normal table where we place the Kinect camera. (b) The proposed height-adjustable overhanging surface is built by placing toio mats on a ferromagnetic metal plate that is overhung by custom magnetic 3D-printed structures.

4.1.3 Thrown Object Design. After a number of iterations on the thrown object design, we selected a design of a 3D-printed hollow ball with a 4 cm diameter (see Figure 7a) to demonstrate most of our applications. This design is easy to be stuck, pushed, and dropped, compared to other thrown object prototypes with different shapes. This hollow ball has a wall thickness of 2.5 mm and weighs 15 g. Nineteen 4 mm x 4 mm disk-shaped neodymium magnets are attached to circular indents that are equally spaced on the ball's surface. In order to allow ThrowIO to support objects other than balls, we used strings to attach objects (e.g., key as in Figure 7c left) to our designed ball. The magnets on the ball allow it to hold up to 108 grams. In some of the applications, we also utilize our other prototype of thrown objects such as a larger-sized heart object with 11 cm width and 5 cm thickness (see Figure 7b).

4.2 Software Control

The software control in the ThrowIO prototype includes (1) thrown object location tracking, enabled by a Kinect camera, and (2) robot

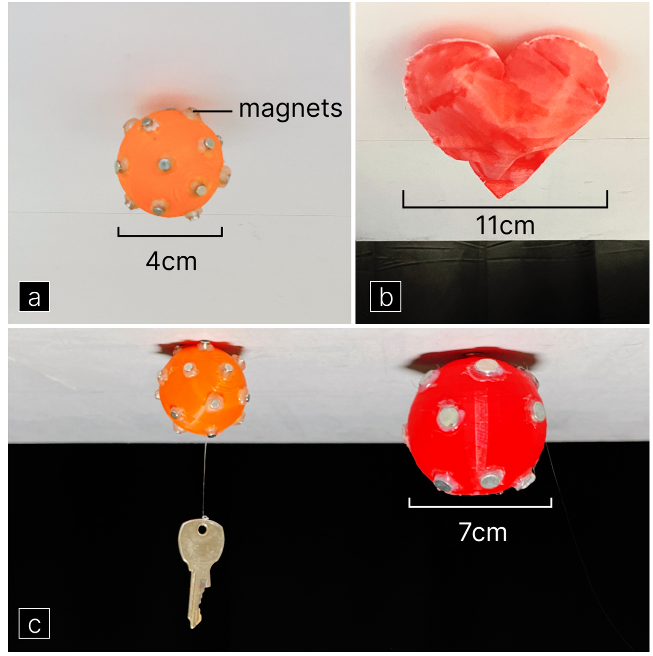

Figure 7: (a) A 3D-printed hollow ball covered with magnets on its surface. (b) Objects such as a heart shape can also be embedded with magnets to be stuck to the surface. (c) Other items such as a key can be attached to the ball, which can also be scaled up to a larger size.

movement control. Both are programmed in the Processing integrated development environment. Our software (including tracking) is performed at a frame rate of 30 fps.

4.2.1 Tracking Thrown Object. A Microsoft Kinect v1 camera, placed on the floor facing upward towards the overhanging toio mat (see Figure 6a), is used in the ThrowIO prototype to track the location of the thrown object, as well as the direction and speed of the thrown object's trajectory. We rely on the RGB camera on Kinect to track the thrown object, which is a ball in our prototype. For calibration, we designed a manual calibration process to identify the toio mat's location within the camera view and the color of the ball, using a cursor input. The calibration results will be saved locally and can be updated as needed (e.g., when the Kinect camera is moved). Based on the calibration information, the tracking software is able to identify the coordinate point (x, y) of the ball's position relative to the toio's coordinate (embedded in the mat). In this way, the robots can be driven to the position of the thrown object and perform *push* and *drop* operations.

We also rely on the depth camera on Kinect to track how the ball is being thrown toward the overhanging surface. As soon as the ball enters the RGB camera (identified with the color tracking), we track the depth, x, and y pixel coordinates of the thrown ball until it sticks and stops moving. With the travel history of the thrown ball (array of x, y, and z coordinate points), we can identify its 3D direction and velocity. These pieces of information are useful for visualizing a thrown object's trajectory beyond the overhanging

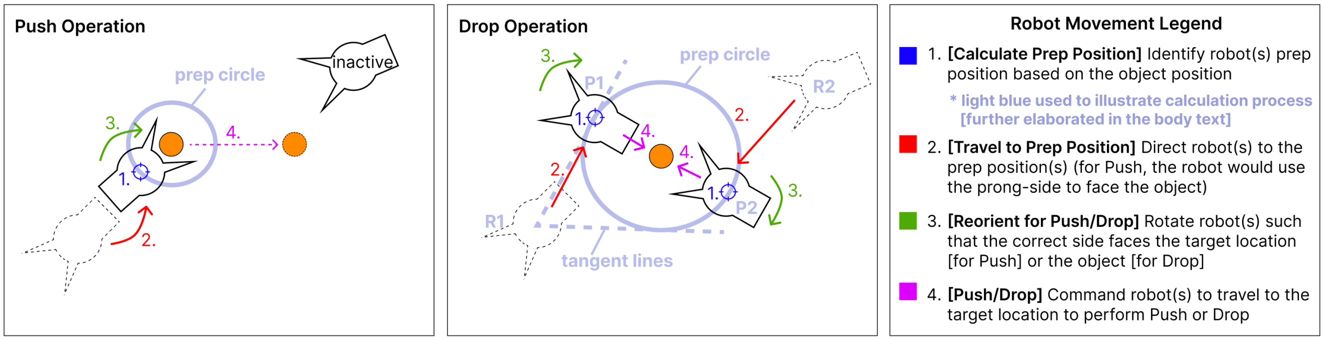

Figure 8: The robots perform *push* operation and *drop* operation. Both operations require robots to follow four main guidelines: calculate prep position, travel to prep position, re-orient for push/drop, and perform push/drop.

surface in certain applications of our implemented system (e.g., as in Figure 12).

4.2.2 Movement Control for Push and Drop. The toio robots in our implemented system are connected through a Rust OSC bridge operated on the computer [6]. Once the toio robots are connected, they are controlled through Processing code to perform basic actions such as traveling and reorienting [7].

To perform the *push* and *drop* operations (Figure 5), we developed a robot movement control algorithm that works based on the tracking information from the Kinect camera to manipulate the thrown objects. Both operations follow a general control flow order of four steps, which is visually illustrated in Figure 8. As shown in the figure, while both operations share general control flow, *push* employs only one robot, while *drop* employs two robots.

Step 1. Calculate Prep Position: The ThrowIO system knows the locations of both the thrown object and robots by default, due to the Kinect camera and toio mats. In the first step, the system calculates a target location for the robot to travel to based on the object's location, in preparation for the upcoming operation.

For the *push* operation, the system finds a location that is on the opposite side of where the thrown object will be pushed to. This location is a point on a sketched prep circle (5 cm radius) centered around the object, and the closest robot to this point would be assigned as the pushing robot (see the light blue sketches in Figure 8 Push Operation).

For the *drop* operation, the system will find two locations for the robots that line up with the object being the center. The two locations for dropping preparation are calculated based on an algorithm that allows both robots to travel to those locations without hitting the ball in later steps (see the light blue sketches in Figure 8 Drop Operation). The algorithm starts by sketching a prep circle (14 cm radius) at the center of the object. From the robot that is closer to the object (denoted as R1), we draw two lines that are tangential to the circle. We then find the tangent point (denoted as P1) that is further away from the other robot (denoted as R2). Finally, we find the antipodal of P1 on the circle (denoted as P2), and set R1's prep position to P1 and R2's prep position to P2.

[6]https://github.com/MacTuitui/toio-osc
[7]https://github.com/MacTuitui/toio_processing

Step 2. Travel to Prep Position: Next, the system directs the robots to travel to the designated locations calculated from the previous step. In *push* operation, the pushing robot will travel to the designated location while having its prong-side facing the object. In *drop* operation, the robot will travel to the prep position assigned to it based on the earlier algorithm. If the thrown object is too close to a robot (i.e., distance is smaller than 14 cm), the robot would travel outwards on the line formed by the thrown object and the robot. Once, the robot travels outwards enough (i.e., the robot's distance is larger than 14 cm), it will then travel to the prep position assigned by the system in Step 1.

Step 3. Reorient for Push/Drop: In the third step, the system rotates the robots to use the correct side of the shell, prong-side or wedge-side, to face the thrown object or the pushed location. In *push* operation, the robot will rotate with the thrown object in its prong-side facing the pushed location. In *drop* operation, one robot will face the object with the prong-side, and the other robot will face it with the wedge-side.

Step 4. Push/Drop: Finally, in the fourth step, the system commands the robots to complete the *push* or *drop* operation. In *push* operation, the robot will push the object until the object arrives at its destination. In *drop* operation, both robots will converge onto the thrown object to drop it, with the prong-side fixating and wedge-side scooping.

4.2.3 Overall Robot Movement Control. Overall, by combining the two primitive operations of **push** and **drop**, the system can move an object that is stuck to the overhanging surface and drop it at a designated point. The robots can also perform the operations at a designated timing by adding wait time flexibly. We have also designed these commands into abstracted functions to allow closed-loop control from Kinect and other custom application software.

As for the general challenge of the control, due to the robot's maximum travel speed, our approach has an inevitable limitation for the speed that it can drop the object after the object is stuck on the overhanging surface. In our implemented system, the maximum travel speed of the robot is 24 cm/s, and the maximum time it takes to drop the object is approximately 5 seconds on our overhanging surface area (when a robot needs to travel diagonally across the mat). This can be further optimized by employing faster robots or applying more robots on the surface to travel efficiently and quickly.

4.3 Graphical Augmentation

As peripheral implementations to graphically augment the ThrowIO system, a monitor and projector are added to develop some of our applications. For the monitor, a 43-inch monitor (94 cm x 53 cm dimensions) is placed vertically above the overhanging surface (see Figure 12). A projector is placed in a diagonal direction below the overhanging surface to perform projection mapping onto the overhanging surface (see Figure 15). The graphical augmentation is created with the Processing application.

4.4 Technical Validation

We ran three evaluations to assess the mechanisms in sticking, pushing, and dropping performed in our implementation of ThrowIO. For all the studies, the 4 cm diameter ball was used as the thrown object (see Figure 7a).

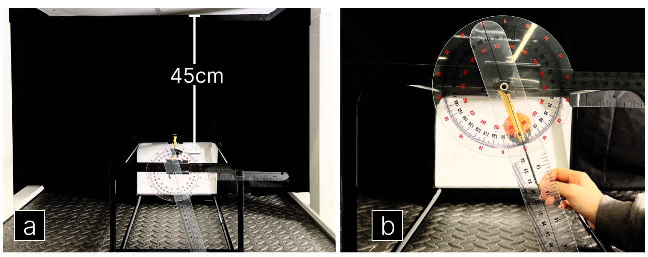

Figure 9: (a) In our evaluation setup where a slingshot mechanism was implemented, (b) we pulled the slingshot down at different angles and distances to vary the throwing speed.

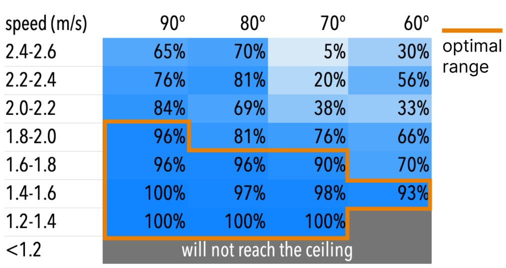

Figure 10: The heatmap shows the sticking performance of our ball prototype where each cell represents the success rate of a minimum of 20 trials. The optimal speed and angle ranges are outlined in orange color.

4.4.1 Throwing and Sticking. For our core throwing interaction, we assess how well our custom-designed ball can stick to the overhanging surface at different speeds and angles. We show our setup and process in Figure 9 and results in Figure 10. We implemented a slingshot mechanism (Figure 9a) and varied the throwing speeds and direction by pulling down the rubber band at different distances and angles (Figure 9b). We filmed slow-motion videos to capture the time duration between launch and hit and calculated vertical speeds. The ceiling height was kept constant (45 cm above the slingshot) for all experiments. As the thrown ball travels quickly, we computed the average vertical speed by dividing the vertical distance by the time duration. We then calculated the average total speed based on the vertical speed and angle using trigonometry. We visualized the success rate for each angle and speed combination in the heatmap in Figure 10 where each cell represents the success rate for a minimum of 20 trials. We outlined the area of the heatmap that shows the optimal range for the sticking performance of our ball prototype. In general, the success rate is higher when the throw is closer to the vertical direction (90°). Within each angle, throwing the ball too fast would introduce big impact that can cause the ball to bounce back from the overhanging surface, resulting in a failure in sticking and reduced success rates. It can be learned from our result that as long as the ball reaches the ceiling, slower throws stick better than faster throws. Close-to-vertical throws (90° and 80°) are more tolerant to fast throws than throwing at a smaller angle (70° and 60°). Future work should continue to iterate on designing thrown objects that stick more robustly for higher speeds and shallower angles to allow for the diverse throwing habits of users.

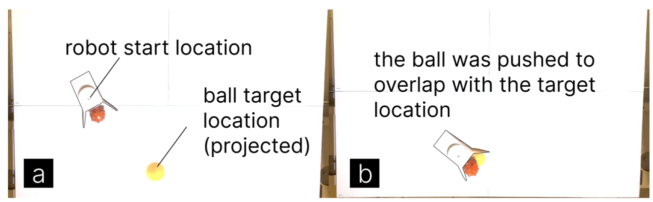

Figure 11: (a) Both the robot's starting location and the ball's target location, which is projected, are randomized. (b) In this trial, the robot successfully pushed the ball to land within a radius range to the target area.

4.4.2 Pushing Operation by Robots. As some scenarios require the object to be dropped at a different location from the sticking point, we evaluated the performance of the *push* operation where a robot pushed the ball from a random point A (where the robot starts) to a random point B (where the ball should land) on the overhanging surface, shown in Figure 11.

Within the detection range on the overhanging surface, the robot started at a random point A, which can be interpreted as the prep position set for the pushing robot in the *push* operation mentioned in section 4.2.2. Since the pushing robot would always navigate to the opposite side of where the ball will be pushed to, the robot was also set to point its prong-side with a random orientation toward a random point B, which is the target point where the ball would be pushed to. Among the 40 trials, the result showed a *100 %* success rate of pushing the object to a certain amount (meaning it never failed to push). Among them, with *92.5 %* of the trials, the object reached within a diameter range (4 cm) to the destination, and *62.5 %* reached within a radius range (2 cm) to the destination. Those that were out of the diameter range were also close to getting to the destination within 8 cm. In the pushing evaluation, the largest distance between point A and point B was 64 cm and the smallest distance was 9 cm, and there was no noticeable trend that showed distance affected the pushing accuracy. If the robot misses pushing the object to the destination, the system will still be able to precisely detect the ball and drop it to the users later. Our current robot gradually decreases its speed as it approaches the pushed location. For potential usages that might require the precision of the dropping location, the relationship between the object size and the angle between prongs can be further investigated. Building on our current shell design, future work can also make the shell smaller by retaining the elements of prongs and a wedge and varying the length of the prongs and the tilt of the wedge.

4.4.3 Dropping Operation by Robots. Finally, we assessed the success rate of the *drop* operation enabled by the mechanical shell design on our robots. In order to test the mechanical design of the toio shell, we placed the ball at a designated spot on the overhanging surface. Then the two robots would travel from their randomized initial positions to drop the object. Our result showed a 100 % of success rate for dropping with 40 times of trials. These results showed the robustness of our mechanical shell designs for performing the operations.

5 APPLICATION

To demonstrate the capability of ThrowIO, we present its applications with our prototypical implementation.

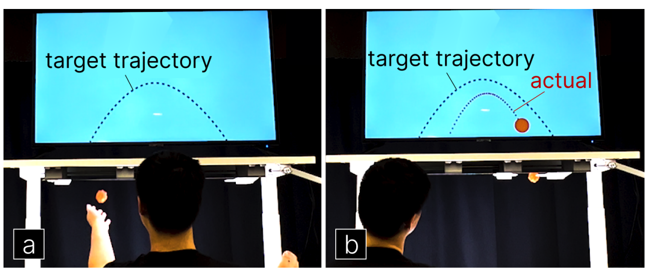

Figure 12: Kinesthetic Learning Application – (a) A user attempts to throw an object following the target trajectory on the monitor screen, but (b) the actual trajectory is a little off.

5.1 Kinesthetic Learning

In the kinesthetic learning application, we demonstrate how ThrowIO could help users learn the ideal way to throw a ball. As shown in Figure 12a, an ideal throwing trajectory that users have to follow will be displayed on the vertical screen. A user will aim at this trajectory by iteratively throwing the objects, shown in Figure 12b. As for the advantage of using the digital display, the time can be slowed down so that users can better understand whether their thrown objects are following the right path to improve their throwing action. Then, users can also practice catching when the objects drop which could be useful for juggling training or rehabilitation. While this application demonstrates the first instance of how a throwing ball can transition between the physical and the virtual worlds to enable novel kinesthetic experiences (building on top of [39]), some technical limitations should be addressed in the future to improve the experience. For example, the system should reflect the 3D depth of the ball on the vertical monitor and replicate a range of angles and speeds for dropping the physical ball.

5.2 Gaming

With the vertical screen, we demonstrate two gaming applications of ThrowIO: Basketball and UFO.

In the basketball game, users can play a basketball-hooping game where players attempt to throw balls in the hoop on the screen (see Figure 13). If users throw the ball at a suitable speed and direction, they will be able to shoot the ball into the hoop, and the next ball will be dropped from the virtual ball feeder on the right-hand side of the screen, accompanied by the actual ball dropping at the same location and at a proper time. This game demonstrates a unique expression to convey the existence of multiple balls in the basket, while, in reality, the experience is handled by only a single physical ball, facilitated by the robot pushing and dropping.

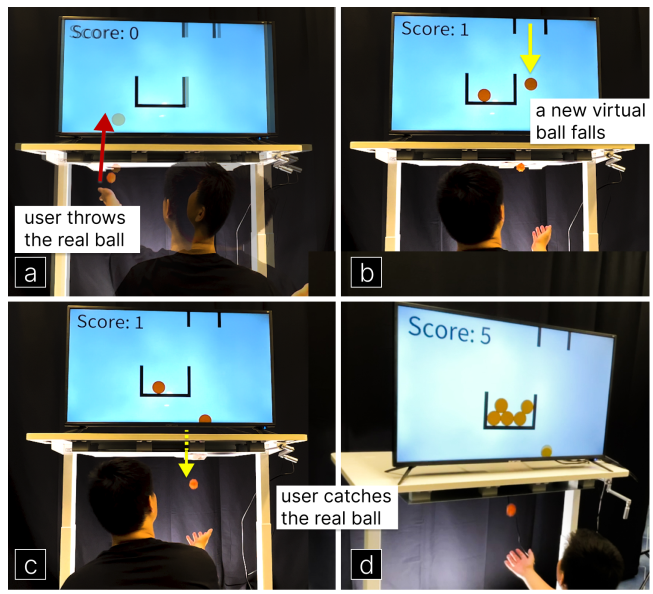

Figure 13: Basketball Game Application – A user attempts to throw a ball into a basket in a basketball game that blends real-world motion with an extended display.

The UFO game demonstrates another "throw at target" gaming experience, but with moving targets and virtually exploding balls. As depicted in Figure 14, users throw a ball to hit a flying UFO on the screen. Once the ball sticks to the overhanging surface, the ball transitions into the virtual screen as a cannon to hit the UFO. The virtual cannon is designed to self-explode if it does not hit the UFO. After the cannon hits the UFO or explodes by itself, a new cannon will be fed to the user from a random position, so users have to catch it by seeing the cannon's trajectory on the screen.

While ThrowIO has technical challenges in not sticking well to the surface depending on the throwing speed and angle (as studied in section 4.4.1), this application demonstrates how the content design could mitigate this limitation by encouraging users to throw the ball again in an appropriate way. For example, after a user throws the ball too fast, resulting in it not sticking to the surface, the system detects the ball speed and displays a message on the screen saying, *"Nice Try! Try throwing SLOWER!"*

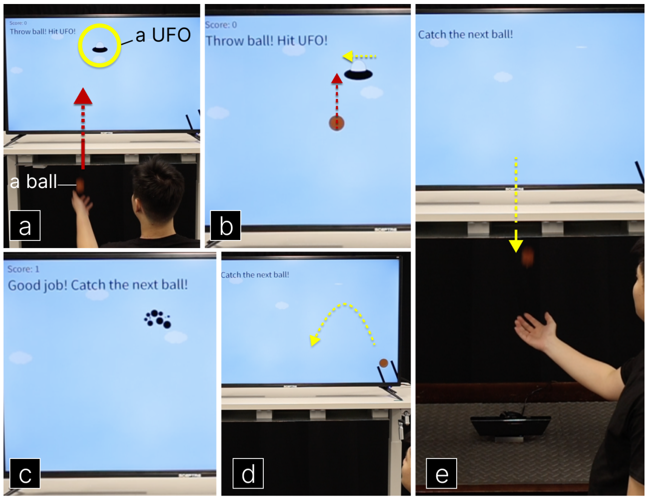

Figure 14: UFO Shooting Game Application – A user attempts to shoot a cannon to hit the UFO in an extended display.

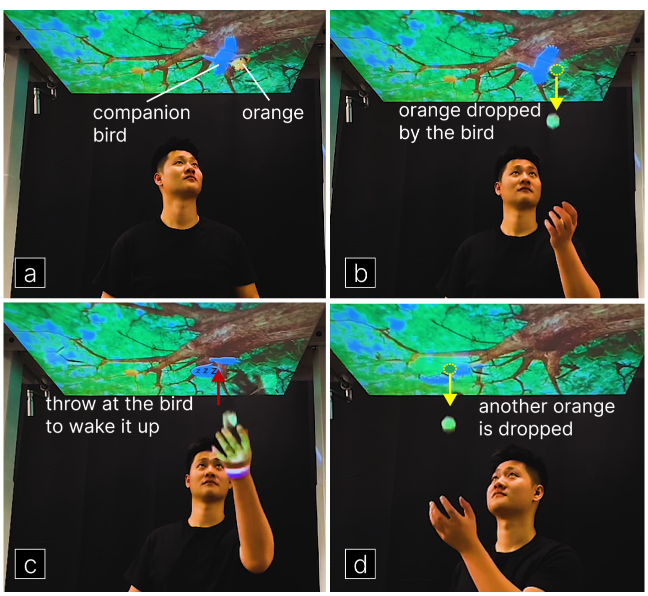

Figure 15: Immersive Haptic Experience Application for fruit-picking experience – (a) The user has a companion bird for orange-picking. (b) The projected orange disappears after the bird's pecking while an actual ball is dropped out of the projected screen by robots. (c) When the bird falls asleep, the user throws the ball back at the bird to wake it up. (d) Now the bird is dropping another orange, reusing the thrown ball which is pushed by robots to the next orange's position.

5.3 Immersive Haptic Experience

ThrowIO can be applied to offer immersive haptic experiences through the use of the overhanging projection. Users could interact with items in the projected digital environment such as picking fruit, catching a baseball, dodging raindrops, or tossing a pizza mediated by ThrowIO's thrown objects. As shown in Figure 15, we illustrate this application to provide an orange-picking experience. In this experience, a user is invited to an orange tree orchard, as the projector screen depicts a view under the orange tree. With this setup, we developed a narrative that a projected lazy white bird is the user's orange-picking companion, shown in Figure 15a. After the bird flies to an orange and pecks for a bit, shown in Figure 15b, the orange will be dropped out of the screen for the user to catch. If the bird falls asleep, as in Figure 15c, the user has to wake it up by throwing a ball at it so that the bird would continue dropping oranges. As the bird flies to the next orange, the same thrown ball that was used to wake up the bird is pushed by robots to the location of the next orange that the bird is dropping (Figure 15d). When the orange is dropped in this haptic experience, an actual ball at the orange's location is dropped by the wheeled robots, showing that the orange has fallen from the screen to the real world. In this way, users could feel a strong immersion into a digitally created experience due to the dropping of physical objects in ThrowIO.

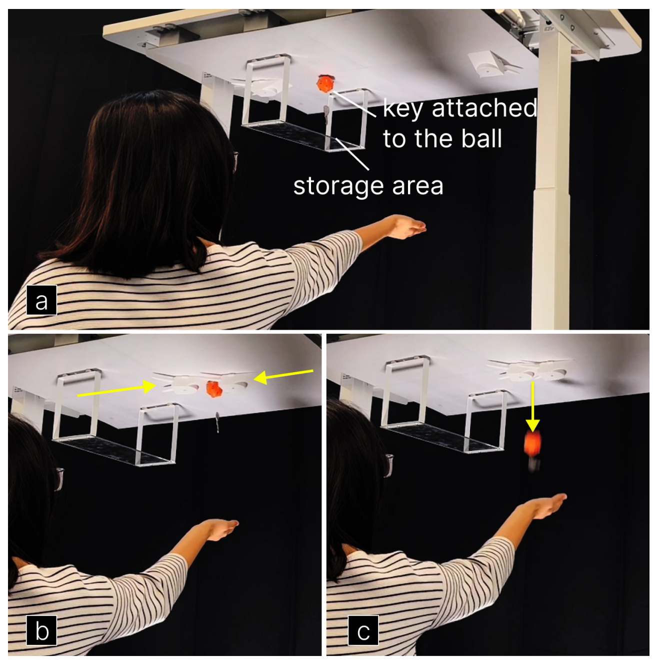

Figure 16: Ceiling Storage Application – (a) Users can attach items, such as a key, to the thrown object and store them to the ceiling by throwing. The wheeled robots will (b) push out the items to a location that is right above the users' hand and (c) drop them to the users.

5.4 Ceiling Storage

ThrowIO can also be applied in organizing room environments, for example, by storing and organizing objects on the ceiling. Users can throw items to be organized on an overhanging surface and store them with the help of robots. Figure 16 shows how a user can store and retrieve a key. Users can throw a key, attached to a magnetic ball, to the ceiling for storage. A robot will then push the sticking objects to a ceiling shelf to store the key. When the user needs to retrieve the stored item, they can just put their hand out under the overhanging surface. The system will track the hand position and detect where the objects are stored. Then, one robot will push the objects out from the ceiling shelf to the user's hand position. The other robot would come from the opposite direction and converge onto the pushing robot to drop the object even if the camera is occluded by the user's hand. We also design the system while keeping user fatigue mitigation in mind. Users can just briefly put their hand in the camera view to activate the robots' action in retrieving an object and don't need to constantly hold their hand while waiting for the robot to travel to the drop location. With this application, we propose that future everyday spaces can use catching and throwing interactions to store and retrieve items on an overhanging surface supported by mobile robots.

5.5 Communication

Lastly, we demonstrate applications in both remote and in-person communications that facilitate human-to-human communication via spatially actuated objects.

For two remote users, they can each have the ThrowIO system and interact with each other by throwing and catching the ball while having their image projected on the overhanging surface or the monitor, shown in Figure 17. For example, a long-distance couple can throw and catch a heart-shaped object with each other while lying on their beds and having their faces projected on the overhanging ceiling. The thrown physical objects become a medium for the users to feel the other's presence, which is stronger than only seeing them on a digital screen. While employing Actuated TUIs for remote communication is a classic topic in HCI [11], ThrowIO contributes to it by incorporating throwing and catching spatial interaction.

For in-person communication, subtle interaction, defined as *"providing input to, or receiving output from systems without being observed"* [4], can be facilitated by ThrowIO. In certain social contexts, it may be disruptive and disrespectful to interrupt a group activity such as talking to another classmate while listening to a teacher's lecture [10]. Our proof-of-concept prototype demonstrates a user scenario where two people can subtly communicate under a table via tangible objects, while not distracting others or breaking the flow of an ongoing conversation. In Figure 18, we show that two users pass a pen under the table in a setting where a presenter is focused on giving a presentation. Through this application, we show how ThrowIO facilitates subtle interaction by utilizing under-table surfaces to push and drop objects onto another person's lap. While our demonstration used a rather small table, limited by the height-changing table we have employed for our paper, this application should be much more effective and useful with a larger table on which it is more difficult for people to hand objects. Our system can technically scale up to 12 (3 x 4: 126 cm x 120 cm dimensions) A3-sized toio mats, and the Kinect camera can also be equipped with a wide angle lens to increase the camera view for detection.

Figure 17: Remote Communication Application – A remote user can interact with another user by throwing and catching a heart object through our system while having their image projected on the overhanging surface.

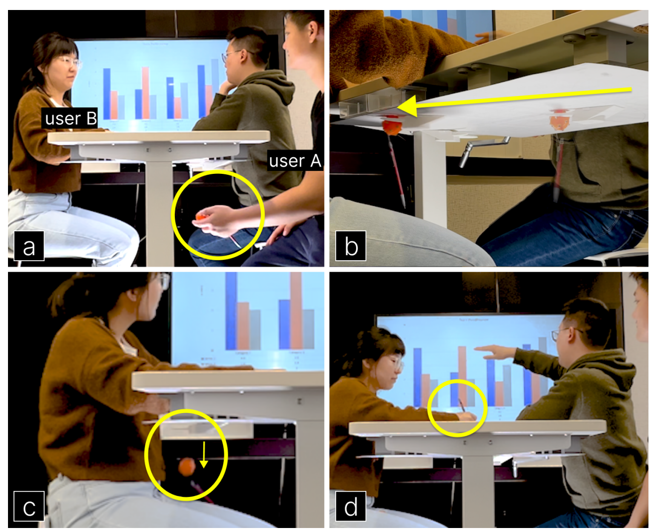

Figure 18: In-Person Communication Application – (a) A person is passing a pen under the table while hoping not to distract the presenter. (b) The wheeled robots push and (c) drop the pen to another person (d) so she could use the pen.

6 USER STUDY

To understand the general usability of ThrowIO, we conducted a user study on our proof-of-concept prototype. The primary goals of our study are to understand the capability and limitations of our current system, and observe how people interact and perceive our system in different applications of ThrowIO. To achieve these goals, we asked participants to complete three tasks that are each part of the three applications of ThrowIO: Gaming, Immersive Haptic Experience, and Ceiling Storage. After participants interacted with one task, they were followed up by a short interview about their experience. Once they finished all three tasks, they were interviewed based on their overall experience. During the study, the order of the tasks was randomized for each participant, and the participant's interaction was recorded. The study protocol was approved by the Institutional Review Board at the University of Chicago (IRB22-1743).

6.1 Participants

We recruited a total of 16 participants who are affiliated with our institution via direct recruitment, flyers, and internal social media. 13 of them identify themselves as male and 3 of them as female. Participants' age ranges from 18 to 22 (M = 22.00, SD = 2.73).

6.2 Tasks

We asked participants to complete three tasks, namely, *UFO*, *Orange*, and *Storage*. Each task is related to one application of ThrowIO, introduced in section 5.

UFO is a task related to the UFO shooting game application mentioned in section 5.2. In this task, participants were told that they had five attempts to throw the object onto the overhanging surface which will then transfer into the screen as a virtual cannon. The task was to hit the UFO shown on the monitor with the cannon (see Figure 19 a). Regardless of whether the participant hit the UFO with the cannon, a new cannon would appear and drop to a random position onto the bottom of the screen, followed by the dropping of the actual object at the same time and location. Participants were asked to catch the object when it was dropped.

Orange is a task that is part of the immersive haptic experience application. In this task, participants were asked to experience an orange-picking immersive story mentioned in section 5.3. They would be throwing an object to wake their bird companion up and catch the orange dropped by the bird twice (see Figure 19 c). In order to wake their bird companion up, participants were asked to throw and stick the object to where the bird was projected on the overhanging surface. Then, the bird would wake up and fly to drop one orange from the tree on the projected screen. As the orange dropped and disappeared from the screen, the thrown object would also be dropped at the same time and location back to the participants who were also asked to catch it. To increase the immersion of the story, we asked participants to throw a white ball such that the projection of an orange could be layered on top of the ball. Since the ball's color may be hard to track in this application, the experimenter would use the mouse to click on the camera screen to indicate where the ball was in the system.

Storage is the remaining task associated with the ceiling storage application in section 5.4. In this task, participants were asked to throw an object attached with a key to the overhanging surface for storage and then retrieve the items by putting their hand out under the overhanging surface twice (see Figure 19 b). During the interaction, participants were asked throw and stick items to the right side of the overhanging ceiling so that the robot can store and push them to the storage area on the left side. To retrieve the items, participants were asked to put their hands out to the area under the right side of the overhanging ceiling which is where the robots would drop the items back to the participants.

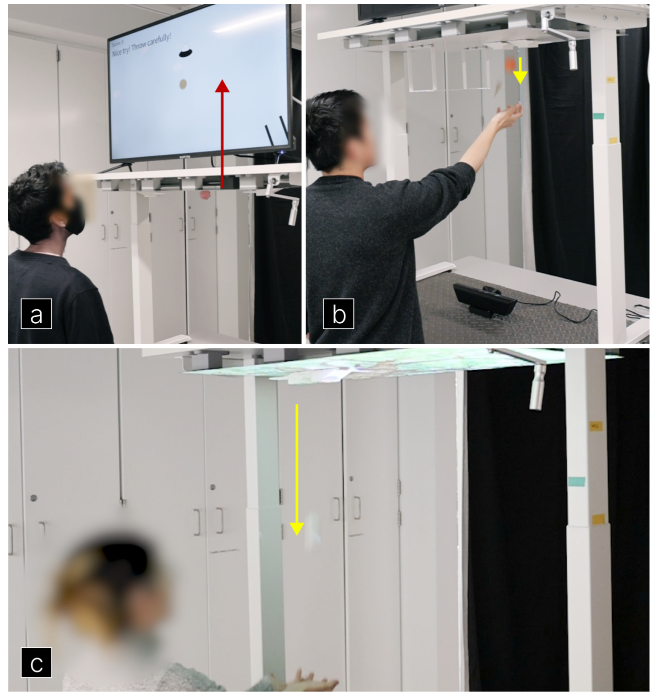

Figure 19: Tasks in the User Study – (a) *UFO*: a participant successfully throws and sticks the object to the overhanging ceiling and the object transitions into a virtual cannon to hit the flying UFO. (b) *Storage*: a participant catches the ball attached with a key. (c) *Orange*: a participant tries to catch the orange dropped by the bird.

Throughout the user study, participants were not able to control the difficulty of the interaction in each task. The only confounding factor in the user study was in the *UFO* task where the height of the overhanging ceiling was adjusted to a height near the participants' eyes, so they could better see the ball transition from the overhanging surface to the vertical screen and vice versa. The height of the overhanging surface for the other two tasks and the practice session was fixed at 188 cm from the floor, which was above all participants' heads.

6.3 Study Protocol

The user study was conducted in a room at our institution. Once the participants were introduced to the overall study flow, they were prompted to sign a consent form to agree to participate in the study and agree to be video-recorded. The participants were first

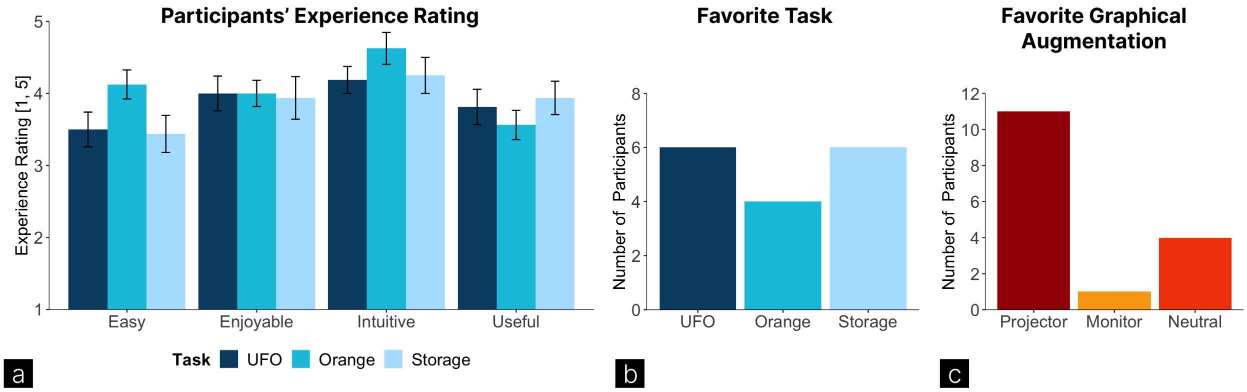

Figure 20: (a) Participant's Experience Rating: Participants rated each task on how easy, enjoyable, intuitive, and useful the system was on a scale from 1 to 5. Error bars depict one standard error from the mean. (b) Participant's Favorite Task: 6 participants liked *UFO* the most, 4 liked *Orange*, and 6 liked *Storage*. (c) Participant's Favorite Graphical Augmentation: 11 participants preferred the projector's screen, 1 preferred the monitor's vertical display, and 4 are neutral to both.

introduced to our system and practiced throwing an object to the overhanging surface and catching the object dropped by the robots 10 times so they could familiarize themselves with the throwing and catching interaction with the system before each task. After practicing, they then interacted with one of the tasks (i.e., *UFO*, *Orange*, *Storage*) in random order. As soon as the participants completed their interaction with one task, they were briefly interviewed on their experience pertaining to how easy, enjoyable, intuitive, and useful the system was on a scale from 1 to 5 and why they gave such a rating. These interview questions were verbally asked by the experimenter to the participants, and their responses were recorded by the camera. When the participants completed all three tasks, they were interviewed with questions regarding their overall experience with the system such as their favorite task, suggested improvements, thoughts of surface height and different graphical augmentations (e.g., monitor/projection). Once the participants were done with the final interview, the experimenter would pull up a laptop showing a demographic survey and ask the participants to fill it up, concluding the user study. The user study took 30 minutes to complete, and participants were compensated with $10 Amazon Gift Card.

Measure	Throw Success Rate (%)	Catch Success Rate (%)
Practice	$M = 82.41$, $SD = 15.41$	$M = 90.77$, $SD = 15.71$
UFO	$M = 88.02$, $SD = 13.58$	$M = 90.83$, $SD = 19.15$
Orange	$M = 75.74$, $SD = 27.04$	$M = 96.88$, $SD = 12.50$
Storage	$M = 71.74$, $SD = 24.69$	$M = 84.38$, $SD = 30.10$
Overall	$M = 79.48$, $SD = 21.43$	$M = 90.71$, $SD = 20.47$

Table 1: Participants' performance in throw success rate and catch success rate in the practice session, tasks, and the overall performance from the user study. The throw success rate is defined by the number of throws that successfully stick to the overhanging surface divided by the number of total throws, and the catch success rate is defined by the number of catches in which participants successfully caught the dropped object divided by the number of total catches.

6.4 Results and Discussion

We collected quantitative responses from the interview in terms of how easy, enjoyable, intuitive, and useful the system was in each task, presented in Figure 20a. Participants' favorite task and preferred method of graphical augmentation were illustrated in Figure 20b and Figure 20c, respectively. Participants' performance in throwing and catching with our system was also summarized in Table 1. Finally, participants' responses during the interview were all recorded to analyze how they perceived and interacted with our system.

6.4.1 Overall Experience. For the overall experience, participants provided much positive feedback, such as *"it is pretty interesting and straightforward," "it is a really cool and innovative system,"* and *"the throwing and catching mechanics in the system were fun to interact"* when they experienced the tasks with our ThrowIO system. One participant mentioned *"it is definitely something that I do not have much experience with, and it is interesting to see how robots and these user interfaces can enhance the experience."* Another participant mentioned *"I thought it was a pretty cool concept translating like a real-life object into a virtual thing."* These responses suggested users can positively accept ThrowIO, and its features could potentially provide more novel spatial interaction and augment even more experiences.

Favorite Task: In terms of the favorite task (see Figure 20b), 6 participants (37.5 %) liked the *UFO* task, as they mentioned it was *"the most fun and interactive"* and it was *"interesting to see physical objects translate into the virtual world."* The *UFO* task was rated 4.00 for how enjoyable the system was, which is as high as the enjoyable rating for the *Orange* task and slightly higher than 3.94 for the *Storage* task (see Figure 20a). 4 participants (25 %) liked the *Orange* task, as they noted it was *"easy, intuitive, and has many elements of fun."* The *Orange* task was scored 4.13 for how easy the system was, which is the highest compared to 3.50 in the *UFO* task and 3.43 in the *Storage* task. Similarly, the *Orange* task was also rated the highest for intuitiveness with a score of 4.63 compared to 4.19 in the *UFO* task and 4.25 in the *Storage* task. 6 participants (37.5 %) liked *Storage* task because they mentioned it was the most *"practical,"* especially at *"utilizing unused space."* The *Storage* task was also rated 3.94 in the usefulness of the system, which is the highest compared to 3.81 in the *UFO* task and 3.56 in *Orange* task. The fact that participants did not skew to like the same task implies that ThrowIO is successful at demonstrating many different applications with various experiences.

Graphical Augmentation: Participants also mentioned that both graphical augmentations were *"helpful"* for them to interact with the system, as a participant commented *"both displays provide a user interface to give context, and they are pretty fluid at reflecting my actions."* Another participant mentioned that *"after the cannon falls out of the frame, then I know the ball is about to drop."* Between the two graphical augmentations, 11 participants (68.75 %) preferred the projector's screen because they mentioned they were *"directly throwing at a graphical target, so it is more immersive and realistic."* Only 1 participant (6.25 %) liked the vertical monitor display because the *"monitor is easy to see the ball trajectory after you throw directly under it, which makes a lot of sense."* 4 participants were neutral to both graphical illustrations (see Figure 20c). From participant's feedback, graphical augmentation is viewed positively to enhance the throwing and catching experience in the system.

Room For Improvements: Participants also pointed out some critiques about the system, which mainly centered around the design of the ball and occasional glitches from the robots. Some participants mentioned that the ball does not stick well to the overhanging surface, as one participant brought up *"the ball can't always stick."* This limitation may have lowered participants' throwing success rate with our system. In Table 1, we summarized the participant's averaged throwing and catching success rates in the practice session, tasks, and overall interaction. We found that the participant's overall averaged throwing success rate is 79.48 % (SD = 21.43 %), and averaged catching success rate is 90.71 % (SD = 20.47 %). We also found that participants had the highest throwing success rate in the *UFO* task (M = 88.02 %, SD = 13.58 %) probably because the height was lower, so it was easier to throw compared to the other tasks. Participants in the *Storage* task had the lowest throwing success rate (M = 71.74 %, SD = 24.69 %) likely due to the attached key onto the thrown object affecting the throwing experience. Other participants commented that they noticed glitches in the robot's movement. One participant commented *"robots are sometimes lagging,"* which was likely due to the user behavior of accidentally occluding the Kinect camera when it tries to track the object position, revealing a disadvantage of the current tracking system being unable to smoothly handle hand occlusion. These shortages in our system should be addressed in future work of ThrowIO.

Finally, participants also suggested some improvements to the current system to convey more information about the system. One participant mentioned *"it will be nice if there are cues that show me whether the ball is detected or not, like a sound or light from the robots."* Another participant suggested that *"the robots can take in the ball and spit it out when you need to drop it, making the ball look like two different objects."* These suggestions can be considered for future work as well.

6.4.2 Task Specific Experience.

UFO: After experiencing the UFO game, participants commented that the system was *"easy"* and *"interesting."* One participant mentioned *"it is entertaining as a multi-model game combining physical stuff and digital screen,"* and another mentioned *"it can be further applied as games in arcades."* A participant said *"with this idea, you can bridge the physical world and the virtual world with a seamless kind of transition."* Participants also stated the interaction with the system was *"intuitive"* and *"straightforward"* as a one mentioned *"it is very self-explanatory and the mapping for the physical ball to the digital display ball is very accurate."* However, some participants commented that the mapping from a ceiling surface to a 2D screen was *"unconventional"* compared to directly interacting with things in the virtual display in the *Orange* task. This suggests that future work on the gaming application should consider mitigating the effect of context-switching.

Orange: From the *Orange* task, participants commented the system was *"immersive"* at prompting users to interact with an ongoing story. One participant commented *"the tangible aspect of throwing something to interact with the screen is stronger than just watching a video."* Participants also considered that the interaction with the system was *"smooth"* and *"easy to follow."* A participant noted *"it is pretty simple and straightforward as you just throw at the bird and it will get motivated."* Another participant brought up that the immersive haptic experience application of ThrowIO *"can be used for interactive museum displays in the future."*

Storage: In the *Storage* task, participants mentioned that the system was *"useful"* as it was a *"convenient and efficient design"* that opens new areas to store items. One participant mentioned *"it opens up additional ways to store things that we conventionally wouldn't be able to take advantage of."* Participants also commented the system was *"easy"* to use. One participant commented *"it is pretty straightforward and there wasn't much of a learning curve."* We interpret the participant meant the storage task was straightforward to use even though the thrown objects don't stick all the time. Another participant mentioned *"all the actions make sense, for example, raising your hand to retrieve the items is skeuomorphic."* Participants also envisioned how the storage application of ThrowIO can be used in the future, as one mentioned *"I would also use [ThrowIO] to store other household items such as a rag next to my fridge."*

6.5 Summary

The user study results have generally shown that the throwing and catching application scenarios enabled by ThrowIO were positively accepted by the users for their usability, enjoyment, intuitiveness, as well as usefulness. Future work should continue building on top of the positive features (e.g., allowing physical objects to transition in and out of the virtual spaces) and incorporate the improvement suggestions from the participants (e.g., showing visual cues to indicate the state of the system). We will be detailing the informed future work in section 7.

7 LIMITATIONS AND FUTURE WORK

The exploration and study in our paper reveal a range of limitations and future research opportunities.

7.1 Scalability for Overhanging Surfaces and the Number of Robots

Our current prototype utilizes a height-adjustable desk with a modification to its underneath surface. We envision the future of ThrowIO to be deployed into everyday surfaces such as a room’s ceiling, dining tables, shelf surfaces, etc. Even beyond overhanging surfaces, we also see interesting opportunities in deploying the system on vertical surfaces such as walls and windows. Together, the overhanging and vertical ferromagnetic surfaces allow users to expand the usage of underused surfaces to facilitate throwing and catching spatial interactions. Towards such deployability, the current system’s scalability challenge has to be addressed, which is now limited by the size of toio mat for tracking. Scalable localization techniques, such as computer vision with external sensors, could be further developed to address this limitation.

While this paper explored developing ThrowIO with two robots as a proof-of-concept system, incorporating more robots is another scalability challenge in the future. By increasing the number of robots, the system should handle multiple objects simultaneously to move and drop, and resolve the latency issue to drop multiple objects in a sequence. To control multiple robots, we need to incorporate advanced control algorithms for path planning that we could address by applying prior research in multi-agent path-finding [54].

7.2 Hardware Updates for Improving the Robustness and Versatility

Various aspects of the hardware could be addressed in the future to improve the experience in ThrowIO. Firstly, the design of the thrown objects could be further explored to improve the sticking performance. Currently, we use rigid balls, as our technical validation reveals in Figure 10 that the ball doesn’t always stick to the ferromagnetic surface under certain speeds and angles. One potential future improvement is to utilize bean-bag balls so that more surface area can stick to the overhanging surface, possibly improving the sticking rate. Our technical evaluation also informs us to use softer and more damping materials to wrap around the thrown object to mitigate the impact and create more contact area when the object hits the overhanging surface. Other uniquely shaped objects or larger objects could be explored for future ThrowIO systems, such as different kinds of juggling props, which will require further designs for the sticking mechanism. Future work can also rely on simulation tools to find the optimal number and strength of magnets to be embedded on the thrown object given its shape and weight.

Another hardware improvement for future work is in the robot design. The current dropping mechanism is using robots to perform free drops using gravity, which cannot variably control the speed and angle of dropping. Future robots could be incorporated with additional actuation mechanisms to vary such object-dropping parameters. These features could better replicate the object’s thrown trajectory with balls dropping in variable speeds and angles for the immersive experience or motor skill training. Additionally, other aspects of the robot hardware could be improved such as by reducing the size of the robot and increasing the speed of motion.

7.3 Object Tracking

Our implementation utilizes an RGB camera to perform color-based object tracking of the thrown object. We see limitations in our tracking technique related to the visibility of the thrown object in less illuminated spaces, in instances where there is less differentiation between the objects and the background, and in the tracking of multiple thrown objects at the same time. Future work can implement a different object-tracking technique, such as Infrared (IR) object detection, for a more robust object detection mechanism. Additionally, as our camera is mounted beneath the interaction surface (as in Figure 6a), our tracking system had to take hand occlusions into account, as we did for our storage application. Future systems could incorporate a multi-camera tracking system [8] to resolve occlusion problems.

7.4 Safety Concerns and Handling of Sticking Failures

As the system physically drops objects from ceilings, safety can be a concern that the future designer around ThrowIO has to consider. For example, objects that can be dangerous to drop (e.g., sharp, fragile objects) could be protected by enclosing them in cushioning material to avoid the object breaking and injuring the users. As for our current developed system, participants never got hurt in our user study, even with the key for the storage application. Also, during the user study, the robots sometimes dropped to a hard surface floor by accident, but they never broke due to the robustness of toio robots.

As there are technical limitations for the objects not sticking to the ceiling after being thrown by users at too fast/slow of a speed (as depicted in Figure 10), future content designers of ThrowIO should design applications to mitigate failures, for example, to encourage users to throw objects again with an appropriate throwing speed.

7.5 Further Understanding Interaction with ThrowIO

Further user studies could inspect how users with different heights, gender, and throwing and catching skill levels utilize ThrowIO in everyday environments and daily use scenarios. In such conditions, we could access other usability aspects in the long term, for example, fatigue and exertion. These user studies can also examine other independent variables such as throwing and catching angles or the weight of the thrown object and investigate how users perceive their interaction with ThrowIO. Furthermore, broader technical validation could be conducted to explore the performance and usability of other types of thrown objects with different shapes and material properties.

8 CONCLUSION

ThrowIO facilitates throwing and catching spatial tangible interaction for users via mobile wheeled robots on overhanging surfaces. The design of ThrowIO allows a breadth of novel applications including kinesthetic learning, gaming, immersive haptic experience, ceiling storage, and communication. To offer a smoother user experience, ThrowIO should address its limitation in future work and

[8] https://optitrack.com/

attempts to scale up its size by enabling more users to throw and catch multiple objects at the same time. We hope this paper opens up a new direction on how users can spatially interact with physical objects via throwing and catching to enrich the future of our spatial user experience in everyday physical space.

ACKNOWLEDGMENTS

We thank David Wu, Chenfeng Gao, and Lilith Yu from Actuated Experience Lab (AxLab) at the University of Chicago for their assistance in building the overhanging surface in our system. We also thank the members of AxLab for their fruitful advice on the project. We also like to thank Anup Sathya, Emily Faracci, and Kevin Wu for proofreading our paper. Specifically, we thank Emily Faracci for taking the photos for our teaser image.

REFERENCES

[1] E.W. Aboaf, S.M. Drucker, and C.G. Atkeson. 1989. Task-level robot learning: juggling a tennis ball more accurately. In *Proceedings, 1989 International Conference on Robotics and Automation*. 1290–1295 vol.3. https://doi.org/10.1109/ROBOT.1989.100158

[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* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300589

[3] Jindrich Adolf, Peter Kán, Benjamin Outram, Hannes Kaufmann, Jaromir Dolezal, and Lenka Lhotska. 2019. Juggling in VR: Advantages of Immersive Virtual Reality in Juggling Learning. *VRST '19: 25th ACM Symposium on Virtual Reality Software and Technology*, 1–5. https://doi.org/10.1145/3359996.3364246

[4] Fraser Anderson, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2015. Supporting subtlety with deceptive devices and illusory interactions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 1489–1498.

[5] Lawrence Gregory Appelbaum, Julia Schroeder, Matthew Cain, and Stephen Mitroff. 2011. Improved Visual Cognition through Stroboscopic Training. *Frontiers in Psychology* 2 (2011). https://doi.org/10.3389/fpsyg.2011.00276

[6] Behnam Bahr, Yingjie Li, and Mahmoud Najafi. 1996. Design and suction cup analysis of a wall climbing robot. *Computers & electrical engineering* 22, 3 (1996), 193–209.

[7] Arthur Lewbel. Beek, Peter J. 1995. The Science of Juggling. *Scientific American* 273, 5 (1995), 92–97. https://doi.org/10.2466/pms.1988.67.2.563 arXiv:http://www.jstor.org/stable/24982089 PMID: 3217207.

[8] PJ Beek and AAM van Santvoord. 1992. Learning the cascade juggle: A dynamical systems analysis. *Journal of Motor Behavior* 24, 1 (1992), 85–94.

[9] Filip Borglund, Michael Young, Joakim Eriksson, and Anders Rasmussen. 2021. Feedback from HTC Vive sensors results in transient performance enhancements on a juggling task in virtual reality. *Sensors* 21, 9 (2021), 2966.

[10] Ahmet Börütecene, Idil Bostan, Ekin Akyürek, Alpay Sabuncuoglu, Ilker Temuzkusu, Çaglar Genç, Tilbe Göksun, and Oguzhan Özcan. 2018. Through the glance mug: A familiar artefact to support opportunistic search in meetings. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*. 674–683.

[11] Scott Brave and Andrew Dahley. 1997. inTouch: a medium for haptic interpersonal communication. In *CHI'97 Extended Abstracts on Human Factors in Computing Systems*. 363–364.

[12] Y Uny Cao, Andrew B Kahng, and Alex S Fukunaga. 1997. Cooperative mobile robotics: Antecedents and directions. In *Robot colonies*. Springer, 7–27.

[13] Severin Engert, Konstantin Klamka, Andreas Peetz, and Raimund Dachselt. 2022. STRAIDE: A Research Platform for Shape-Changing Spatial Displays based on Actuated Strings. In *CHI Conference on Human Factors in Computing Systems*. 1–16.

[14] Samin Farajian, Hiroki Kaimoto, and Ryo Suzuki. 2022. Swarm Fabrication: Reconfigurable 3D Printers and Drawing Plotters Made of Swarm Robots.

[15] Mel E Finkenberg and Bonnie Mohnsen. 2003. Virtual reality applications in physical education. *Journal of Physical Education, Recreation & Dance* 74, 9 (2003), 13–15.

[16] Phil Frei, Victor Su, Bakhtiar Mikhak, and Hiroshi Ishii. 2000. Curlybot: designing a new class of computational toys. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 129–136.

[17] Stefan Gradl, Bjoern M. Eskofier, Dominic Eskofier, Christopher Mutschler, and Stephan Otto. 2016. Virtual and Augmented Reality in Sports: An Overview and Acceptance Study. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct* (Heidelberg, Germany) *(UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 885–888. https://doi.org/10.1145/2968219.2968572

[18] Ken Hashizume and Tomoyuki Matsuo. 2004. Temporal and spatial factors reflecting performance improvement during learning three-ball cascade juggling. *Human movement science* 23, 2 (2004), 207–233.

[19] Tobias Höllerer, JoAnn Kuchera-Morin, and Xavier Amatriain. 2007. The allosphere: a large-scale immersive surround-view instrument. In *Proceedings of the 2007 workshop on Emerging displays technologies: images and beyond: the future of displays and interacton*. 3–es.

[20] 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* (Cairo, Egypt) *(MUM 2018)*. Association for Computing Machinery, New York, NY, USA, 7–18. https://doi.org/10.1145/3282894.3282898

[21] Junyu Hu, Xu Han, Yourui Tao, and Shizhe Feng. 2022. A magnetic crawler wall-climbing robot with capacity of high payload on the convex surface. *Robotics and Autonomous Systems* 148 (2022), 103907.

[22] Hiroki Kaimoto, Kyzyl Monteiro, Mehrad Faridan, Jiatong Li, Samin Farajian, Yasuaki Kakehi, Ken Nakagaki, and Ryo Suzuki. 2022. Sketched Reality: Sketching Bi-Directional Interactions Between Virtual and Physical Worlds with AR and Actuated Tangible UI. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 1, 12 pages. https://doi.org/10.1145/3526113.3545626

[23] 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.

[24] Jens Kober, Matthew Glisson, and Michael Mistry. 2012. Playing catch and juggling with a humanoid robot. *IEEE-RAS International Conference on Humanoid Robots*, 875–881. https://doi.org/10.1109/HUMANOIDS.2012.6651623

[25] Jens Kober, Katharina Muelling, and Jan Peters. 2012. Learning throwing and catching skills. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 5167–5168. https://doi.org/10.1109/IROS.2012.6386267

[26] Rotem Lammfromm and Daniel Gopher. 2011. Transfer of Skill from a Virtual Reality Trainer to Real Juggling. *BIO Web of Conferences* 1 (12 2011). https://doi.org/10.1051/bioconf/20110100054

[27] Ralph LaRossa. 2009. 'Until the ball glows in the twilight': Fatherhood, baseball, and the game of playing catch. In *Fathering through sport and leisure*. Routledge, 39–55.

[28] 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*. 97–109.

[29] Jinha Lee, Ernest Post, and Hiroshi Ishii. 2011. *ZeroN: Mid-air tangible interaction enabled by computer controlled magnetic levitation*. Ph. D. Dissertation. https://doi.org/10.1145/2047196.2047239

[30] Ann MacPhail, David Kirk, and Linda Griffin. 2008. Throwing and catching as relational skills in game play: Situated learning in a modified game unit. *Journal of Teaching in Physical Education* 27, 1 (2008), 100–115.

[31] Michael R Marner, Sam Haren, Matthew Gardiner, and Bruce H Thomas. 2012. Exploring interactivity and augmented reality in theater: A case study of Half Real. In *2012 IEEE International Symposium on Mixed and Augmented Reality-Arts, Media, and Humanities (ISMAR-AMH)*. IEEE, 81–86.

[32] Michael R Marner, Ross T Smith, James A Walsh, and Bruce H Thomas. 2014. Spatial user interfaces for large-scale projector-based augmented reality. *IEEE computer graphics and applications* 34, 6 (2014), 74–82.

[33] Benjamin Meyer, Pascal Gruppe, Bastian Cornelsen, Tim Claudius Stratmann, Uwe Gruenefeld, and Susanne Boll. 2018. Juggling 4.0: Learning Complex Motor Skills with Augmented Reality Through the Example of Juggling. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings* (Berlin, Germany) *(UIST '18 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 54–56. https://doi.org/10.1145/3266037.3266099

[34] Tohru Miyake, Hidenori Ishihara, Ryu Shoji, and Shunichi Yoshida. 2006. Development of small-size window cleaning robot by wall climbing mechanism. In *Proceedings of the 23rd International Symposium on Automation and Robotics in Construction*. Citeseer, 215–220.

[35] Pouya Mohammadi, Milad Malekzadeh, Jindrich Kodl, Albert Mukovskiy, Dennis L Wigand, Martin Giese, and Jochen J Steil. 2018. Real-time control of whole-body robot motion and trajectory generation for physiotherapeutic juggling in VR. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 270–277.

[36] Michael P Murphy and Metin Sitti. 2007. Waalbot: An agile small-scale wall-climbing robot utilizing dry elastomer adhesives. *IEEE/ASME transactions on*

Mechatronics 12, 3 (2007), 330–338.

[37] Ken Nakagaki. 2020. Mechanical Shells: Physical Add-ons for Extending and Reconfiguring the Interactivities of Actuated TUIs. In *Adjunct Publication of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 151–156.

[38] Ken Nakagaki, Joanne Leong, Jordan L. Tappa, João 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* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 882–896. https://doi.org/10.1145/3379337.3415831

[39] Ken Nakagaki, Jordan L Tappa, Yi Zheng, Jack Forman, Joanne Leong, Sven Koenig, and Hiroshi Ishii. 2022. (Dis)Appearables: A Concept and Method for Actuated Tangible UIs to Appear and Disappear Based on Stages. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 506, 13 pages. https://doi.org/10.1145/3491102.3501906

[40] Yoichi Ochiai, Takayuki Hoshi, and Jun Rekimoto. 2014. Pixie dust: graphics generated by levitated and animated objects in computational acoustic-potential field. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–13.

[41] Matthew KXJ Pan and Günter Niemeyer. 2017. Catching a real ball in virtual reality. In *2017 IEEE Virtual Reality (VR)*. IEEE, 269–270.

[42] 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* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 809–818. https://doi.org/10.1145/1240624.1240746

[43] Kai Ploeger, Michael Lutter, and Jan Peters. 2020. High acceleration reinforcement learning for real-world juggling with binary rewards. *arXiv preprint arXiv:2010.13483* (2020).

[44] Parinya Punpongsanon, Daisuke Iwai, and Kosuke Sato. 2015. SoftAR: Visually Manipulating Haptic Softness Perception in Spatial Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics* 21, 11 (2015), 1279–1288. https://doi.org/10.1109/TVCG.2015.2459792

[45] GR Reddy and Damien Constantine Rompapas. 2020. Visuotouch: Enabling haptic feedback in augmented reality through visual cues. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*.

[46] Robin Ritz, Mark Müller, Markus Hehn, and Raffaello D'Andrea. 2012. Cooperative quadrocopter ball throwing and catching. *Proceedings of the ... IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4972–4978. https://doi.org/10.1109/IROS.2012.6385963

[47] Calvin Rubens, Sean Braley, Antonio Gomes, Daniel Goc, Xujing Zhang, Juan Pablo Carrascal, and Roel Vertegaal. 2015. Bitdrones: Towards levitating programmable matter using interactive 3d quadcopter displays. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 57–58.

[48] Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. 2014. Programmable self-assembly in a thousand-robot swarm. *Science* 345, 6198 (2014), 795–799.

[49] James R Rudd, Lisa M Barnett, Michael L Butson, Damian Farrow, Jason Berry, and Remco CJ Polman. 2015. Fundamental movement skills are more than run, throw and catch: The role of stability skills. *PloS one* 10, 10 (2015), e0140224.

[50] Daniela Rus, Zack Butler, Keith Kotay, and Marsette Vona. 2002. Self-reconfiguring robots. *Commun. ACM* 45, 3 (2002), 39–45.

[51] Asuki Saito, Kazuki Nagayama, Kazuyuki Ito, Takeo Oomichi, Satoshi Ashizawa, and Fumitoshi Matsuno. 2018. Semi-autonomous multi-legged robot with suckers to climb a wall. *Journal of Robotics and Mechatronics* 30, 1 (2018), 24–32.

[52] Dominik Schmidt, Raf Ramakers, Esben W. Pedersen, Johannes Jasper, Sven Köhler, Aileen Pohl, Hannes Rantzsch, Andreas Rau, Patrick Schmidt, Christoph Sterz, Yanina Yurchenko, and Patrick Baudisch. 2014. Kickables: Tangibles for Feet. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3143–3152. https://doi.org/10.1145/2556288.2557016

[53] Ryota Shibusawa, Mutsuhiro Nakashige, and Katsutoshi Oe. 2022. DualityBoard: An Asymmetric Remote Gaming Platform with Mobile Robots and the Digital Twins. In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction* (Sapporo, Hokkaido, Japan) *(HRI '22)*. IEEE Press, 1035–1039.

[54] Roni Stern, Nathan R Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, TK Satish Kumar, et al. 2019. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Twelfth Annual Symposium on Combinatorial Search*.

[55] 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*. ACM. https://doi.org/10.1145/3472749.3474821

[56] Ryo Suzuki, Clement Zheng, Yasuaki 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*. 493–505.

[57] Dishita G Turakhia, Yini Qi, Lotta-Gili Blumberg, Andrew Wong, and Stefanie Mueller. 2021. Can Physical Tools That Adapt Their Shape Based on a Learner's Performance Help in Motor Skill Training?. In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction* (Salzburg, Austria) *(TEI '21)*. Association for Computing Machinery, New York, NY, USA, Article 17, 12 pages. https://doi.org/10.1145/3430524.3440636

[58] Brygg Ullmer and Hiroshi Ishii. 1997. The metaDESK: models and prototypes for tangible user interfaces. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*. 223–232.

[59] Brygg Ullmer, Hiroshi Ishii, and Robert JK Jacob. 2005. Token+ constraint systems for tangible interaction with digital information. *ACM Transactions on Computer-Human Interaction (TOCHI)* 12, 1 (2005), 81–118.

[60] Ozgur Unver and Metin Sitti. 2009. A miniature ceiling walking robot with flat tacky elastomeric footpads. In *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2276–2281.

[61] Peng Wang, Xiaoliang Bai, Mark Billinghurst, Shusheng Zhang, Dechuan Han, Mengmeng Sun, Zhuo Wang, Hao Lv, and Shu Han. 2020. Haptic Feedback Helps Me? A VR-SAR Remote Collaborative System with Tangible Interaction. *International Journal of Human–Computer Interaction* 36, 13 (2020), 1242–1257. https://doi.org/10.1080/10447318.2020.1732140 arXiv:https://doi.org/10.1080/10447318.2020.1732140

[62] Jizhong Xiao, William Morris, Narashiman Chakravarthy, and Angel Calle. 2006. City climber: a new generation of mobile robot with wall-climbing capability. In *Unmanned Systems Technology VIII*, Vol. 6230. SPIE, 471–480.

[63] Zeyu Yan, Anup Sathya, Pedro Carvalho, Yongquan Hu, Annan Li, and Huaishu Peng. 2021. Towards On-the-wall Tangible Interaction: Using Walls as Interactive, Dynamic, and Responsive User Interface. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–6.

[64] J-F Yang and JP Scholz. 2005. Learning a throwing task is associated with differential changes in the use of motor abundance. *Experimental brain research* 163, 2 (2005), 137–158.

[65] Stephen Yang, Brian Ka-Jun Mok, David Sirkin, Hillary Page Ive, Rohan Maheshwari, Kerstin Fischer, and Wendy Ju. 2015. Experiences developing socially acceptable interactions for a robotic trash barrel. In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 277–284. https://doi.org/10.1109/ROMAN.2015.7333693