ZoomWalls: Dynamic Walls that Simulate Haptic Infrastructure for Room-scale VR Worlds

Yan Yixian
Tohoku University, Japan
yyyttt120@outlook.jp

Kazuki Takashima
Tohoku University, Japan
takashima@riec.tohoku.ac.jp

Anthony Tang
University of Toronto, Canada
tonytang@utoronto.ca

Takayuki Tanno
Tohoku University, Japan
t-tanno@riec.tohoku.ac.jp

Kazuyuki Fujita
Tohoku University, Japan
k-fujita@riec.tohoku.ac.jp

Yoshifumi Kitamura
Tohoku University, Japan
kitamura@riec.tohoku.ac.jp

ABSTRACT
We focus on the problem of simulating the haptic infrastructure of a virtual environment (i.e. walls, doors). Our approach relies on multiple ZoomWalls—autonomous robotic encounter-type haptic wall-shaped props—that coordinate to provide haptic feedback for room-scale virtual reality. Based on a user's movement through the physical space, ZoomWall props are coordinated through a predict-and-dispatch architecture to provide just-in-time haptic feedback for objects the user is about to touch. To refine our system, we conducted simulation studies of different prediction algorithms, which helped us to refine our algorithmic approach to realize the physical ZoomWall prototype. Finally, we evaluated our system through a user experience study, which showed that participants found that ZoomWalls increased their sense of presence in the VR environment. ZoomWalls represents an instance of *autonomous mobile reusable props*, which we view as an important design direction for haptics in VR.

Author Keywords
Encountered-type haptic devices; Immersion

CCS Concepts
•Human-centered computing → Human computer interaction (HCI); Virtual reality;

INTRODUCTION
Room-scale virtual reality (VR) experiences allow people to freely walk around a physical space (e.g. [10, 55]), but we do not yet have effective ways for providing haptic awareness of the enclosure and objects in the space. In real life, we see and feel the boundaries of a physical space by walking through it, seeing its boundaries, and feeling the walls. Yet, when walking is used for navigating VR spaces [34, 37, 44], people easily and frequently cross over the virtual boundaries (e.g. walls), which breaks the immersive quality of the experience. To provide this haptic feedback, a conventional passive haptics approach uses physical objects in the world to act as tangible stand-ins for corresponding virtual objects. For example, conventional VR arcades reinforce the existence of a virtual wall (seen in VR) with a physical wall—when the user reaches out to touch the virtual wall, the user feels a physical wall—thus, the wall's existence is seen in the visual world and reinforced in the tactile/haptic domain. Such passive props increase immersion and sense of presence in a VR environment [7, 30], even when the props are low-fidelity and do not share all the properties of the corresponding virtual object [21, 35].

Recent efforts have been made to explore how to employ *autonomous mobile reusable props* since a passive haptics approach is generally expensive. For instance, a naïve passive haptics approach would replicate each virtual object with a corresponding prop, meaning that the virtual world is replicated in the real world. A more sophisticated approach re-uses props, focusing on virtual objects that are likely to be interacted with next. TurkDeck takes this approach, where a small group of humans dynamically rearrange props—such as wall elements, ground elements and objects—based on where the user is moving in the virtual space [11]. While this is effective in creating an immersive room-scale VR experience, it requires a massive amount of human labour. Most closely related to our work is RoomShift, which relies on uses a set of autonomous robots carrying pieces of furniture around the room-scale virtual environment [39]. Our work builds on the core idea expressed by RoomShift, and we go beyond this by designing algorithms to coordinate the movement of multiple robots, demonstrating how autonomously reusable props improve room-scale VR experiences.

We propose ZoomWalls, a proof-of-concept prototype of robotic props that move and reposition themselves to provide haptic feedback for represented virtual structures (i.e. walls, doors, and so on). In the ZoomWalls system prototype, the user can feel the haptic infrastructure of the environment, which is represented by ZoomWall props. Each ZoomWall prop is a moving "wall segment," where the movement of the props is coordinated by a system that predicts which parts of the virtual environment are needed haptically next, and dispatches props accordingly. Our system innovates on prior

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

UIST'20, October 20–23, 2020, Minneapolis, Minnesota, USA

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

DOI: https://doi.org/10.1145/3313831.XXXXXXX

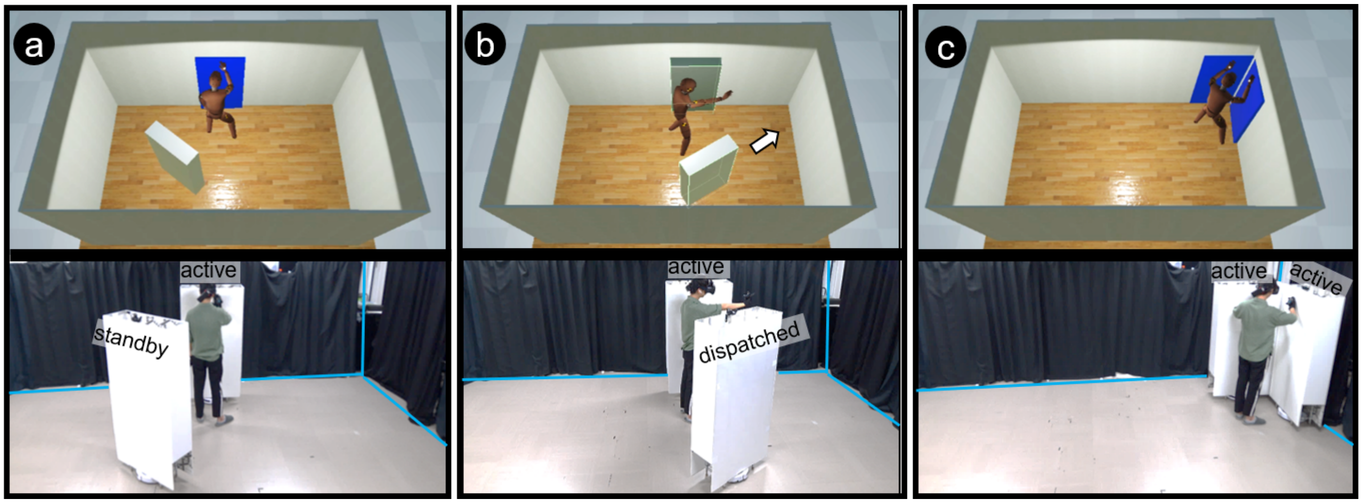

Figure 1. ZoomWalls provide haptic feedback about the virtual infrastructure. In this figure, we show a Debug view of the virtual environment, where we can see where the system understands the physical ZoomWalls to be in the virtual environment; in the system itself, the ZoomWalls themselves do not appear in the VR world. (a) One active ZoomWall (blue) represents the virtual surface, while the other ZoomWall is on standby (grey); (b) As the user turns to approach another virtual surface, the standby ZoomWall is dispatched (yellow) to materialize the surface on the right, while the active ZoomWall follows the user to continue representing the virtual wall along the top; (c) Both ZoomWalls, now in the active state, dynamically organize themselves to provide haptic feedback of the corner.

approaches by providing just-in-time haptic feedback for objects the user is about to touch, ensuring that props operate in concert with one another without colliding.

We see ZoomWalls as an instance of an important forthcoming phase of haptics in VR—the use of autonomous reusable mobile props—where the principal challenge will be to coordinate multiple robots. The ZoomWalls system was developed through an iterative design process, and based on a user experience study, we provide new insights on the development of autonomous reusable prop systems. Figure 1 illustrates the ZoomWalls system: based on a set of prediction algorithms (that source the user's position, orientation, and movement), the system predicts what walls need to be realized next and accordingly dispatch unused ZoomWalls to the destination. These algorithms were refined through several simulation studies, where we identified effective algorithms for coordination, and identified the number of ZoomWall props required for low latency haptic experience. Once we realized the full system prototype, we conducted a user study to examine how the ZoomWalls system prototype would impact the experience of a virtual escape room.

This paper makes three contributions: first, we extend prior work on autonomous reusable mobile props by detailing coordination algorithms that drive the ZoomWall prototype; second, we show through a user study that our proof-of-concept ZoomWalls prototype can be effective in providing an immersive, fun experience; finally, we discuss new challenges and solutions for designing autonomous reusable mobile props.

RELATED WORK

Our interest is in realizing haptic experiences in room-scale virtual reality using ungrounded, encountered-type haptic devices—physical props that the user encounters when exploring or interacting with the space. To set the stage for our work, we describe work in three related areas: reusable props, collaborative robots, and touch/walking prediction in VR.

Reusable Props

Because a conventional passive haptics approach relying on full-sized inert props [7, 30] is impractical for room-scale VR, many researchers have begun exploring the use of reusable props. These explorations have involved perceptual illusions, or by literally re-using multiple props. Mahdi et al. [4] and Cheng et al. [9] develop a perceptual illusion called haptic retargeting which allows a single prop to be reused multiple times for different virtual objects (e.g. to be grasped or touched). TilePop uses a similar approach to dynamically construct body-size props, supporting whole-body interactions by using stacked cube-shaped airbags [43]. Many grounded encounter-type haptic devices have been proposed (e.g. [3, 17, 20, 48]), but they normally do not support users' natural walking experience in room-scale VR.

Some researchers have focused on providing physical feedback using mobile, ungrounded devices for room-scale VR. MoveVR generates force feedback by Roomba-actuated moving props [49]. PhyShare replies on a set of actuated robots for sharing haptic virtual reality between users in distance [16], where these devices can represent several different virtual objects at different times. One of the devices is a floor-based "virtual wall", which inspired our own work on ZoomWalls. RoomShift builds on this approach by representing room-scale haptic environment in VR using furniture-moving robots [39]. Similar to PhyShare, these props can represent different virtual objects at different times. Others have affixed props onto drones, which supply haptic sensation for small floating virtual objects [2, 18, 54]. Beyond just being passive props, the drone can also provide some force-feedback (3N), which can simulate subtle feedback like collision with a bubble and bite of insect [1].

One challenge that needs to be addressed with reusable props is quickly moving them to ensure that they are in the "right place at the right time." TurkDeck [11] relies on human "actuators" who quickly move physical props around the VR user in a just-in-time fashion, allowing the props to be re-used in the room-scale VR experience. Needless to say, the experience relies heavily on human labour: in their demonstration, TurkDeck was facilitated by 10 humans to manipulate a total of 65 props for a 7-min experience.

Our work builds primarily on PhyShare, RoomShift and TurkDeck. Unlike TurkDeck, ZoomWalls employs robots to actuate the reusable props (as opposed to human labour). Like PhyShare and RoomShift, we employ robots as reusable props; in this work, we explicitly explore and detail the algorithms for coordinating multiple actuators in an unified system.

Coordinating Multiple Robots for Human Experiences

Our approach of using multiple coordinating robots to construct an environment builds off considerable prior work. For instance, MovemenTable[41] and AdapTable [26] show how multiple interactive tabletops can physically reconfigure themselves based on users' dynamically varying interaction. Shape-Shifting Wall Display [42] provides several mobile vertical walls that can reconfigure themselves based on task demands and the content. LiftTiles is a room-scale shape-changing interface using modular inflatable block-shapes props to create different room structure and furniture [40].

Within the VR context, Yim et al. [56] suggests that small robots can be used as proxies to provide active haptic feedback by simulating the touch and feel. Zhao et al. [59] illustrate this basic principle using small, self-assembling robots that can combine themselves into arbitrary shapes. We scale up this basic idea for room-scale VR, where we represent walls and large obstacles rather than small handheld objects. CirculaFloor is an unique device providing an infinite ominidirectional walking floor by coordinating and reusing a set of moving tiles [22]. Our work is heavily inspired by this work, where we use autonomously coordinated wall-shape props that responds to the user's movement in VR to provide a coherent haptic environment.

Touch/walking Prediction in VR

To respond appropriately to the human interaction, several VR systems predict how and when a user will interact with it. For instance, Sparse Haptic Proxy uses a highly accurate prediction of users reaching targets by inferring their future actions based on their gaze and hand motions on a desktop VR setup [9]. To create a highly accurate and rapid encounter type haptic system, Yokokohji et al. [57] used a convex polyhedron surface for collision detection with the user's hand, and [58] uses an understanding of kinematics features (e.g., velocity profile) of reaching and grasping behaviours. Their later work [25] relies on constructing reachability maps for a robot that represents surfaces with a much smaller prop. Thus, there is a wide range of prediction approaches, but most are for achieving grounded in-situ haptic devices. CirculaFloor [22] predicts a user's walking direction from their knee position relative to the predefined origin, which is effective to coordinate and reuse the multiple robotic tiles. Unlike the established ML (machine-learning) based human motion classifier in the real world, motion classification in room-scale VR is still an open question. Yet, recent work suggests trajectory-based motion description using DTW (dynamic time warping) classifier [15]. Based on this prior work, we designed two predictors to achieve ZoomWall concept by analyzing on-line user's motion data relative to their surroundings.

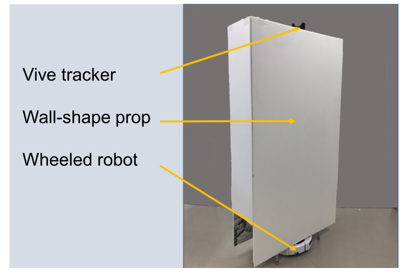

Figure 2. A ZoomWall instance is made of three major components.

ZOOMWALL DESIGN AND IMPLEMENTATION

We propose ZoomWalls—autonomous robotic encounter-type haptic wall-shaped props—that coordinate to provide just-in-time haptic feedback for virtual boundaries (i.e, walls, doors) in room-scale virtual reality. Each ZoomWall is a human-size moving wall segment enabling human-scale encounter-type haptic experiences in a room-scale VR scenario. By combining multiple ZoomWalls, maze-like structures such as corners or culs-de-sac of corridors can be physically represented. The ZoomWall movements can further represent movement of the infrastructure (e.g., sliding windows or doors). Each ZoomWall prop is coordinated by a system that predicts which parts of the virtual environment are needed haptically next, and dispatches props accordingly, ensuring that props operate in concert with one another without colliding with one another. ZoomWalls follow the user as they walk around based on the user's position, orientation and movement. Once the system predicts which virtual surfaces could be touched, it dispatches standby ZoomWalls to locations where it anticipates a need.

Design and Prototype of Individual ZoomWalls

Since each ZoomWall is designed to simulate a part of the infrastructure, we paid particular attention to the size of each ZoomWall. Specifically, we wanted a user to be able to touch the panel with two hands shoulder-width apart, and from ankle height to above the head. We anticipated that users might walk next to a virtual wall, so the ZoomWall would need to be able to follow along. Similarly, some surfaces may not appear at right angles from one another, so the ZoomWalls needed to have the ability to rotate easily.

To realize a first version of the ZoomWall concept, we implemented each using two basic components: foam core boards (140cm $\times$ 80cm $\times$ 2cm), and a wheeled robot (Roomba, a cleaning robot) as an actuator (Figure 2). chosen based on comfortable reaching based on ergonomics studies [29]. The Roomba supports a maximum of 0.5 m/s with 2 DoF (forward/backward and rotation). Each ZoomWall has an upright foam core board on both sides of the device. To make each

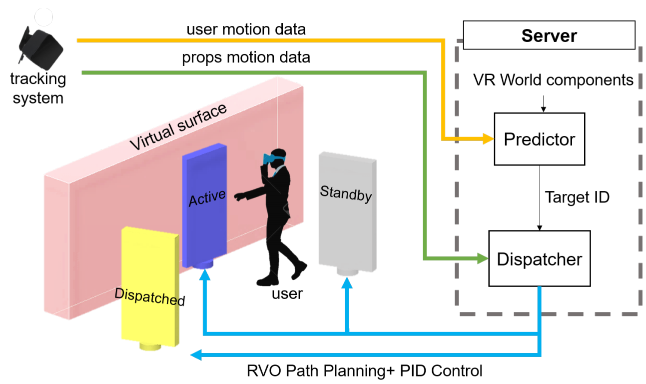

Figure 3. The ZoomWall system is comprised of ZoomWall units, a tracking system, and a server with Predictor and Dispatcher components. Based on motion data from the user, the Predictor determines which virtual wall the user will touch, and Dispatcher determines which ZoomWall should be dispatched to materialize the target surface.

ZoomWall stable, a 10 kg weight was placed on the Roomba, and thus each ZoomWall can withstand approximately 23 N force at arm height—sufficient for general touch interaction, though not quite enough for leaning against. Each ZoomWall is tracked using an HTC Vive tracker, which supplies location and orientation information.

Design of ZoomWall System

Figure 3 illustrates an overview of ZoomWall system comprising an HTC Vive Tracking system, a worn HTC Vive HMD, several ZoomWall instances, and a custom Windows-based server application. The server application maintains the virtual Unity environment, makes predictions about the user's movement (based on the tracking system data), and coordinates and dispatches ZoomWalls to locations where the user is about to interact with virtual objects. The server receives motion data from the tracking system, and sends commands the ZoomWalls using serial communication via Bluetooth. To simulate physical contact between user's hand and virtual/physical walls, users use Vive hand controllers or gloves with an embedded Vive tracker.

Predictor and Dispatcher Servers

As shown in Figure 3 the server is comprised of two components: the `Predictor`, which identifies surfaces of virtual objects that are likely to be touched based on the user motion data, and the `Dispatcher`, which selects the best ZoomWall candidate to represent a virtual surface. The `Predictor` operates primarily on the user's tracked position, orientation, and movement speed. Once the `Predictor` identifies a virtual surface that has not been physically represented, it notifies the `Dispatcher` of the target. In the next section, we discuss our explorations into the design of the Predictor's algorithm, which considered both naïve and machine learning approaches.

The `Dispatcher` selects the best ZoomWall candidate to represent a virtual wall surface based primarily on an estimate of which standby ZoomWall can get to the target location most rapidly. This estimate accounts for distance, and characteristics of our implementation—specifically, which ZoomWall would need to rotate the least, and which ZoomWall (with various characteristics) would be appropriate to represent the target surface. Once a ZoomWall is selected by the `Dispatcher`, the ZoomWall is given the ID of the target virtual surface, and the position of the user's projection onto the surface. The goal position of the dispatched ZoomWall is set to this point, and the goal orientation matches the virtual surface's orientation.

ZoomWall States

The `Dispatcher` manages multiple ZoomWalls in the same space, where each ZoomWall is in one of three states:

- `Active` - The ZoomWall represents a virtual surface, and only moves if the user walks along a path parallel to its face (blue ZoomWall in Figure 3 materializes the pink virtual surface)the `Dispatcher` sends a different target surface ID to this ZoomWall, this would be changed to the `Dispatched` state. In case that no target ID is assigned to this ZoomWall, it will be set to `Standby` state.
- `Standby` - The ZoomWall does not represent anything in the virtual environment but "follows" the user so that it can move into position when needed (white ZoomWall in Figure 3), and
- `Dispatched` - The ZoomWall is selected to represent a virtual surface, but it has not yet arrived its destination location (yellow ZoomWall in Figure 3). Immediately after the ZoomWall arrives at the destination, it will be set to the `Active` state to materialize the surface.

ZoomWalls in the `Standby` state closer to the user based on two principles: (i) `Standby` ZoomWalls stay just out of user's reach—this means ZoomWalls need to move very little distance when they are needed, and matches the intuition that a haptic wall will only be necessary when the user is about to touch a wall; (ii) `Standby` ZoomWalls avoid other ZoomWalls that are already in the `Active` state.

In our implementation, we use four locations around the user as possible ZoomWall `Standby` locations. These points are 0.9m away from the user, and evenly distributed around them. When a ZoomWall is set to `Standby`, the system selects one of the four standby points as the target destination for the ZoomWall. This selection works based on two rules: if all the other ZoomWalls are also in `Standby` state, the closest and unoccupied point will be chosen; if other ZoomWalls are `Active`, the unoccupied standby point furthest from the `Active` ZoomWalls is chosen. The latter rule reduces the cases of conflict between ZoomWalls.

Path Planing and Control

Once the server gives the target location to the `Dispatched` ZoomWall, its path planning and control are executed. We employ a Reciprocal Velocity Obstacles (RVO) algorithm [45] like [39] so that the ZoomWalls avoid collisions with the user and each other. This algorithm avoids obstacles according to velocity of every possible obstacle through a series of implicit predictions. We model the user as a moving obstacle with a known velocity, so the ZoomWalls can move around one another and the user. Similar to [25], this path planning process is always running. When the `Predictor` notices something has changed in the system (e.g. the user has changed speed), the system can re-plan the paths for the ZoomWalls instantly.

We used a PID control method for ZoomWall control after setting the movement path. This PID control algorithm was designed for two-wheel robot. We tuned the tolerance parameters of error correction during the feedback loop for both translation and rotation movements to avoid unnecessary rotation and vibration around the target. Due to the tolerance parameters and potential tracking error in the Vive tracking system [5], the average positioning and rotation errors were 5.5 cm (SD = 0.19) and 1.0 deg (SD = 0.68).

Because our design involves adding obstacles to the physical environment that the user cannot see, safety is an extremely important requirement. During our studies, the operator uses a dead-man switch, an emergency stop mechanism. This stops every ZoomWall in the physical environment, changes the user's view to a "Debug" mode, where all of the reference props are visible to the user. This switch is also automatically activated when the system loses tracking of any of the ZoomWalls or the user.

DESIGNING THE PREDICTION ALGORITHM

While the goal of the prediction algorithm used by the `Predictor` is clear (i.e. determine which virtual surface is likely to be touched), we found that as an unconstrained problem, it was largely intractable: user movement and behaviours depend heavily on the structure of the virtual environment, as well as the application. Some game simulations may require the user to turn at any moment (e.g. to spot an enemy), or insist that a certain wall be touched (e.g. to activate a switch); furthermore, a user may behave erratically or unexpectedly. As a first stage proof-of-concept approach to support the ZoomWall system's design, we chose a simple, four-wall room (4m $\times$ 3m) where the user only walks around inside.

We designed two prediction algorithms: first, a bounding box algorithm, and second, we augmented the bounding box with a machine learning algorithm based on user movement prediction. We expect that both algorithms would be reasonably effective depending on the situation and number of ZoomWalls in use: the bounding box is fairly straightforward and might be applicable in a wide variety of situations with some slight tweaking; the second one is slightly more robust to user movement, but pre-training is required.

Bounding Box Algorithm

This prediction is based on a straightforward collision detection approach: any virtual surface that enters this bounding box is flagged as a surface that is likely to be touched. As illustrated in Figure 4, this surface then would be marked as a target surface, and passed onto the `Dispatcher` to materialize. In this algorithm, we model the user's anticipated touch area based on a bounding box around the user in the horizontal plane. The bounding box is bilaterally symmetrical, and is slightly biased to the forward direction of the user to account for forward walking movement. For our system, this bounding box is a virtual 2m $\times$ 1.4m rectangle that surrounds the user. Its width is based on an average human's arm span.

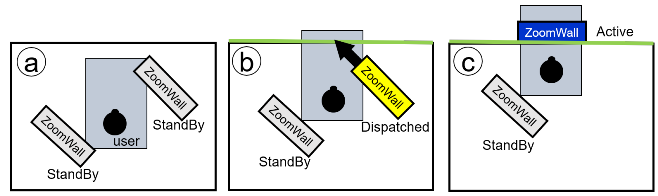

Figure 4. A top-down view of the Bounding Box `Predictor` at work. (a) The user is far from all virtual surfaces, so the ZoomWalls are both on `Standby`. (b) When the user's Bounding Box intersects with the virtual surface, the surface is predicted to be touched, and the top ZoomWall is set to `Dispatched`, and moves toward the destination surface. (c) The ZoomWall has arrived to materialize the surface, and is set to the `Active` state.

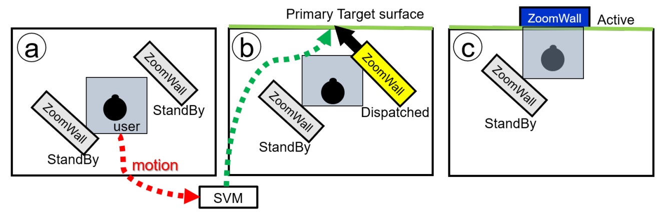

Figure 5. A top-down view of the Bounding Box + Motion ML `Predictor`. (a) The user is far from all virtual surfaces, so the ZoomWalls are both on `Standby` around the user; (b) Based on user's walking movements, the Motion ML predicts a priority surface to be touched, and the `Dispatcher` sends one of the ZoomWalls; (c) The ZoomWall has materialized the surface, and set to the `Active` state.

While this naïve approach is conceptually simple and works for a compact space, there may be significant latency in certain circumstances since the system does not prioritize which surfaces ought to be prioritized. Consider a scenario involving two ZoomWalls that are `Active`, but then the user turns and quickly moves toward the back wall. In this case, moving one of the ZoomWalls to materialize a surface on the opposite side of the room would be challenging. While this problem can be mitigated with more ZoomWalls, we are interested in slightly more intelligent coordination that can take advantage of more of the tracking information —e.g. the user's heading or movement speed.

Bounding Box + Motion ML Algorithm

We designed a second algorithm that accounts for this intuition—the user's head orientation and their movement speed may also help predict surfaces that should be touched next. We retain the underlying bounding box algorithm (albeit with a smaller bounding box: 1.4m *times* 1.4m). This algorithm adds the notion of "priority" to which surfaces need to materialized—a surface that is to be *touched* receives the highest priority, while surfaces that are in the bounding box have a normal priority. Figure 5 shows that at each step, a support vector machine (SVM) predicts which surface the user intends to touch based on the user's movements during the last second (this is set as highest priority). The `Dispatcher` sets the anticipated touch target as the highest priority surface, and sends the closest ZoomWall (regardless of its state) to materialize it. Unlike the bounding box predictor, this rapidly positions a ZoomWall to what may be far surfaces if we know the user is likely to touch it. This improves the performance if there are fewer ZoomWall units.

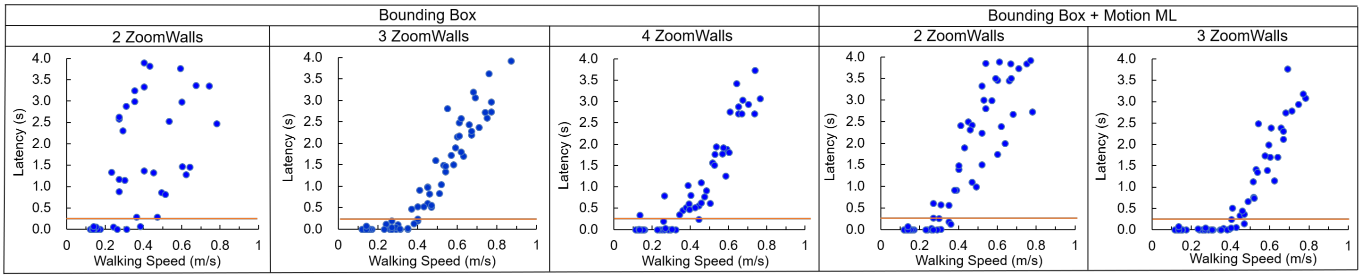

Figure 6. Simulation result: simulated average walking speed vs. latency for different ZoomWall configurations

Our motion ML solves a simple classification problem, where the system predicts the ID of a target wall from the four walls based on seven input variables: position (Pos_x, Pos_z), head orientation (Dir_x, Dir_z, velocity (Vel_x, Vel_z), and angular (turning) velocity ($Gyro_y$). The data is captured at 20Hz, so a total of 140 features are fed into the model at each step [53, 19, 28].

Training. We recruited 3 participants to generate training data for the model. Participants wore a motion tracker, and walked around in the our four-walled 3m×4m room virtual room. As they walked, they selected a target wall to touch, indicating this using a corresponding button on a hand-held controller (this would function as ground truth). They then walked until they touched the wall. Each participant completed three trials of 10 minutes, totaling 90 minutes of training data. We conducted a preliminary evaluation with our motion ML. We used 10-fold cross validation, and the average accuracy of this model was 0.89, which was sufficient for our needs.

Evaluation of Predictor Algorithms using Simulations

To understand the performance of the two predictor algorithms, we conducted a simulation study. The main goal of this study was to understand how user walking speeds would affect the performance of different ZoomWall configurations (i.e. with a different algorithm or number of ZoomWalls). This evaluation would also serve to help us refine and tune parameters of the ZoomWalls implementation before realizing the entire system. The simulator comprised a simulated ZoomWall system in Unity where each ZoomWall had real-life movement ability (two-wheeled robot, and 0.5 m/s movement speed), and the `Predictor` and `Dispatcher` components as described above. We recruited five members of our lab to record user walking data, and then programmatically varied the average walking speed of their data (so it would maintain natural speed variation) before feeding it into the simulator. We then measured the overall performance of the system in terms of latency—how long it would take before a ZoomWall would materialize the virtual surface.

Walking Data. We recruited five members of our lab to record the user's walking data. We asked each to wear a HMD and walk at comfortable speed around the virtual room with four different path patterns, where they would touch the virtual surfaces in order: clockwise, counter-clockwise, back and forth along the length, and back and forth along the width. These patterns represented different challenging use cases for our system; for instance, clockwise and counter-clockwise patterns represent the situation where the next virtual surface is adjacent to the current one. We asked participants to complete each of these paths five times. We scaled the walking data's speed at a number of scales: 0.2, 0.4, 0.6, 0.8, and 1.0 (original) to obtain different average walking speeds (in m/s), and used these as input within the ZoomWall system simulator.

Simulation. We varied two variables: the number of ZoomWalls available to the algorithm (2, 3, or 4), and the algorithm being used (bounding box, and bounding box + motion ML). Within the simulation, we recorded latency—how long did it take for a ZoomWall to locate itself at the correct target location after the user "touched" the surface. The ideal latency should be 0s, where a dispatched ZoomWall arrives at the target location before the virtual surface is touched.

Results. Our simulated walking speed data varied from 0.12 m/s to 0.87 m/s. We observed latencies of 0s to 3.95s in our simulation. Figure 6 illustrates latency for these trials relative to the simulated average walking speed for each configuration of ZoomWall. We also show a 0.2s latency cutoff threshold, which represents the largest latency that would be considered as a good haptic experience [24]. For the ZoomWall configurations we simulated, this would mean that a user's walking speed would need to be lower than 0.4 m/s to ensure a good haptic experience.

This result shows that the bounding box + motion ML with 3 walls was the best, supporting the fastest walking speed (around 0.45m/s). We also learn that other conditions such as bounding-box + motion ML with 2 walls and bounding-box with 3 walls would also be usable if the walking speed is limited to 0.3-0.35 m/s. Furthermore, the bounding box with 2 walls was not satisfactory due to its high variance. The bounding box with 4 walls was not necessary as its performance was roughly equivalent to the three wall version due to increased collisions between ZoomWalls.

Through an iterative analysis, normal VR user walking speeds (0.6m–1m/s in our study) can be supported if ZoomWalls had 1m/s speed capability. While these findings are preliminary, they showed us that the ZoomWall system design with coordinated robots was feasible. In particular, it showed that given the limitations of the current actuators in the ZoomWall units, we needed to carefully design the experience to ensure that the user never walked faster than 0.4m/s with our current setup with two or three ZoomWalls in a 4m × 3m space.

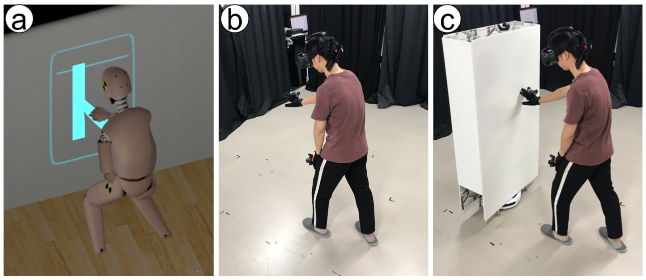

Figure 7. User experience study task. (a) In the virtual world, the user traces a rune pattern on the wall; (b) In the baseline condition, the user does this in mid-air; (c) In the ZoomWall conditions, a ZoomWall functions as a touch surface for the user's tracing movements.

USER EXPERIENCE STUDY

To understand the user experience of the ZoomWall system, we conducted a study where participants would play a VR game with and without the ZoomWall system. We were interested in two aspects of the experience: first, what would the actual latency of the prototype system be in practice, and second, how would participants experience the system. We tested three different conditions: a baseline (without ZoomWalls), the Bounding Box algorithm, and the Bounding Box + Motion ML algorithm. We wanted to understand whether the latency would be acceptable, whether it would detract from the experience, and whether participants would find that ZoomWalls aided their feeling of presence in the environment.

Task Design

Participants played a VR escape room game where the goal was to escape a dark, four-walled room of 4m$\times$3m while holding a small lantern (the VR controller). Participants could unlock the room by tracing five different rune patterns that appeared one at a time on different walls (Figure 7). Tracing each rune required large hand movements on the wall (participants wore a tracked glove), and each rune appeared at random on a different wall after they complete the previous one.To understand performance of ZoomWalls as encounter-type haptic devie, this "simulation-level" information about the target wall was not fed into the `Predictor` subsystem; instead the `Predictor` ran blind to this information (and only used the bounding box or user movement data).

Consistent with others who have designed for room-scale VR (e.g. [11]), we designed a narrative to slow the movement speed of participants. Within the context of the narrative, we told participants that an invisible demon living in the room could spot them if they moved too quickly (like in the movie The Predator). This speed threshold was set at 0.4m/s, and an audio cue would be played if participants moved faster.

Conditions

We studied three variations of the ZoomWall system:

- *Baseline* - No robotic props;
- *Bounding box (BB)* - Three ZoomWalls using the bounding box predictor;
- *BB + ML* - Two ZoomWalls using the bounding box + Motion ML predictor.

While the simulation results suggested that the bounding box + motion ML predictor with three walls is the most powerful configuration, we skipped this as the latency performance would be satisfactory. Instead, we tested lower performance conditions with two or three ZoomWalls, which allows us to consider the actual necessary number of the walls. As shown in Figure 7, the baseline condition did not have haptic feedback, while the Bouding Box and Bounding Box + motion ML conditions provided haptic feedback by ZoomWalls when the user traced the rune pattern on the virtual wall.

Measures

We used objective measures to assess system performance, and subjective measures to assess the user experience. As with the evaluation of the prediction algorithms, we also measured the system latency—how much delay was there between the time participants touched a virtual surface and the time the ZoomWall arrived to materialize the virtual surface. We selected two sub-scales from the Witmer and Singer's Presence Questionnaire [52], which comprised ten 7-point likert items assessing "Involvement" and "Naturalness." We also added new 7-point Likert scale question asking "How fun was your experience?"

Participants

We recruited 12 participants (3 females) from our university. Their age ranged from 22 to 28 years (mean 24.8). Five of these had previously experienced a VR environment.

Procedure

After a brief explanation of the system and study protocol, participants donned equipment for the study, including a head-mounted display, a pair of gloves with embedded Vive trackers, and a noise-canceling headphone. The headphone (Sony MDR-100ABN) was used to mask the noises the ZoomWalls generated. Participants then completed a warm up task to train them in: (i) walking at a slow speed for the system, and (ii) tracing shapes with their hands. In the warm-up virtual environment, a green arrow sign in front of the participant indicated that his/her walking speed was slow enough, and this arrow would turn red if the participant walked too quickly (i.e. >0.4 m/s).

Participants completed three trials with each system configuration condition. The presentation order of the conditions was counter-balanced across participants. After each trial, participants completed the presence questionnaire. Finally, after all trials were complete, we conducted a semi-structured interview asking about participants' experiences, where we probed for possible improvements to the system.

Findings

Latency

We measured an average 0.09 s (SD = 0.34) latency for the bounding box condition, and 0.07s (SD = 0.31) for the BB + ML condition. These values were better than the 0.2s guideline [24], suggesting we only need two ZoomWalls to achieve this kind of simple room scenario (with a movement speed of less than 0.4m/s).

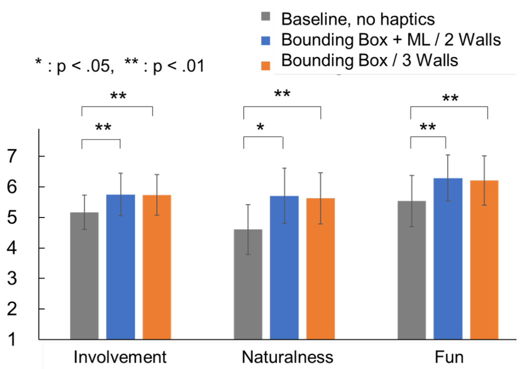

Figure 8. Participants' ratings of sense of presence and fun experience across conditions in the user experience study.

Presence and Fun
Participants rated their sense of presence through our 12-point questionnaire with 7 likert scale for each condition, where six items related to Involvement, four items comprised Naturalness, and two assessed Fun. For each sub-scale, we used a mean score as the rating on the sub-scale.

Here, we focused on comparing the baseline and the two ZoomWall conditions (BB and BB+ML). We confirmed that all three sub-scales had normality and conducted an one-way ANOVA ($n = 12$) to understand the differences between Baseline and BB, and between Baseline and BB + ML. All results are summarized in Figure 8. We found that BB + ML and BB conditions had significantly higher involvement than Baseline condition ($F_{(1,11)} = 32.2, p < .001, \eta^2 = .745$ and $F_{(1,11)} = 59.9, p < .001, \eta^2 = .845$, respectively). We also found that the two ZoomWall conditions, BB + ML and BB, were perceived more natural than the baseline ($F_{(1,11)} = 15.2, p < .005, \eta^2 = .581$, and $F_{(1,11)} = 31.9, p < .001, \eta^2 = .744$, respectively). Finally, for our original question set about fun, the same tendency as other scales were observed: that BB + ML and BB conditions offered more fun experience than the baseline. ($F_{(1,11)} = 19.8, p = .001, \eta^2 = .643$ and $F_{(1,11)} = 12.5, p < .001, \eta^2 = .533$, respectively).

Interview Findings
All participants reported that the ZoomWall improved their experience of the VR environment. They also did not report noticeable latency, which is consistent with the latency scores. Finally, none of the participants felt bothered by the slow walking speed as it was thought to be a part of the experience (i.e. to avoid waking up the demon). One of them even said he looked forward to playing an FPS game demonstration version of the system (not ready at that moment).

Nevertheless, participants indicated several aspects of the ZoomWall user experience that detracted from the experience, or could be improved. 4/12 participants reported that the noise from the ZoomWall was distracting (i.e. they could hear the ZoomWall as it moved). 4/12 participants also reported that the position of the ZoomWall did not always match the apparent virtual surface. The problem seems occur due to sensor drift, either from the PID controller or tracking system, though these might be mitigated with haptic retargeting techniques [4, 9]. Finally, 6/12 participants described feeling uneasy knowing that the ZoomWalls were moving around them throughout the study. This is a deeper issue that relates to users feeling uncomfortable moving around a physical space without being able to see potential obstacles they could collide with. Our participants suggested embodying the ZoomWalls in the simulation using semi-translucent assets/objects.

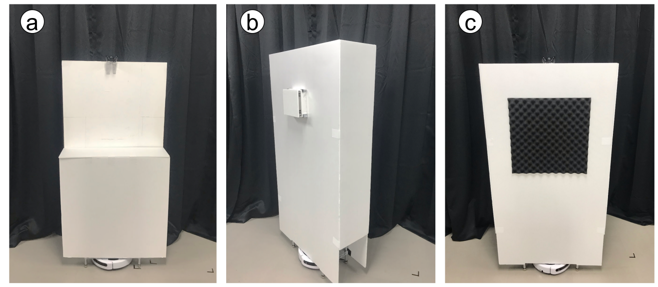

Figure 9. Variations of the touchable area of the ZoomWall: (a) a half-height ZoomWall; (b) a ZoomWall with a button prop; (c) a ZoomWall with a ridged texture for its surface.

FURTHER ZOOMWALL EXPLORATIONS
Since ZoomWalls represent an instance solution of autonomous mobile reusable props, we used the system to explore the design space. In several different designs of different scales, we explored using ZoomWalls to simulate various types of objects/surfaces and movable objects (like [39]), and designed more sophisticated room and space scenarios.

Physical Interactions. We have used the ZoomWalls to simulate movable structures. For instance, we used ZoomWalls to simulate normal doors when we move the rotation axis of the robot to one side of the foam core prop. Here, when the door is pressed, it rotates inwardly or outwardly, and this can be felt by the user. As illustrated in Figure 11b, the "default" ZoomWall can also be used to simulate a sliding door. When the prop is touched or pushed gently from the side, it slides to the side like a real-world sliding door.

We also explored several variations of the touchable area of the ZoomWall. Figure 9 illustrates ZoomWalls with different heights, different on-board textures, as well as additional props affixed to the ZoomWall surface itself. "Low" ZoomWalls could be used to simulate a window sill, or a table, for example. Similarly, ZoomWalls with different textures can simulate different types of surfaces. Finally, an additional prop affixed to the ZoomWall surface could realize hand-sized structures in the simulation (e.g. a doorknob). These physical "effects" may not be necessary in the general case—Edward [21] argue that physical props that only roughly approximate the virtual model increase immersion. The width of each ZoomWall could also vary. Our informal simulation showed that the use of wider walls was more efficient to materialize the virtual surfaces, but can potentially increase collision risk, especially when they rotate. Thus, we believe that the current human-sized prop is reasonable to offer sufficient flexibility. If a wider continuous surface that supports swiping is desired (as shown in Cliff application), placing multiple walls side-by-side would be appropriate.

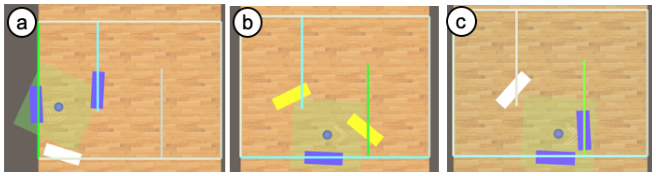

Figure 10. A complex room example for ZoomWall (top view): (a)-(b) show a sequence of the three walls movements around the walking user along the zigzag room structure. A priority virtual surface to be touched, active state ZoomWalls, and dispatched ZoomWalls are in green blue and yellow, respectively.

Room Shapes and Size. We also simulated more complex room shapes such as mazes. Figure 10 illustrates an example of a complex room structure where three ZoomWalls coordinate to support maze exploration. Our motion ML algorithm relies on an extremely simplified virtual scenario (a four-walled room), and training data collected for that scenario. However, for the FPS game shown in Figure 11, we divided the maze into four parts and sequentially applied our SVM predictor for each according to the player's location. We then confirmed that our simple motion ML worked well for each part. It may be possible to generalize this algorithm for general virtual environments by modeling them as a series of connected 1m$\times$1m room units. Example room shapes include a narrow corridor, a room with a central pillar, L-shape, zig-zag shapes, which were materialised by ZoomWalls without significant latency. We illustrate some of these examples in our video figure.

Regarding room size, we found that the ZoomWall's latency performance strongly depends on the difference between the user and actuator's speeds. The second important variable is the bounding box size, which should be varied according to the physical room size. Without considering redirected walking techniques [33], the room size should be limited to the capture volume. However, with a sufficiently large number of ZoomWalls, combined with redirected walking techniques (e.g. [38, 27]), we argue that it should be possible to design larger virtual spaces that can provide haptic infrastructure.

Example Applications. We have primarily simulated game environments. Figure 11 illustrates a simple first person shooter game, where the ZoomWalls simulate a corridor, corners, a door, and a room. This is similar to the approach taken by VR theme parks like "The Void" [46], except that ZoomWalls dynamically reconfigures the haptic infrastructure on demand without the need of human intervention (cf. [11]). In Figure 11, ZoomWalls represents the corridor and door, and works as a shield for the user to avoid the enemies' shots. We also have a cliff simulation where a user walks along with a narrow path with support from a cliff wall. Beyond entertainment, ZoomWalls can provide stronger spatial awareness for tactical training, emergency drill or traditional architectural design. Room-scale VR is also promising for remote office scenarios, where ZoomWalls can represent shared digital whiteboard or remote users avatars with physical constraints. Finally, ZoomWalls could be actual physical boundaries between the tracking and non-tracking areas, which would prevent bystanders from accidentally going into tracking areas, or to create comfortable social distances between HMD users.

DISCUSSION AND FUTURE WORK

Based on our experiences designing and implementing this prototype, we identify several opportunities for designers of autonomous mobile reusable props in future implementations.

Limitations due to slow actuator. Our robot is based on a robot vacuum cleaner with typical two-wheel mechanism which generally limits its degrees of freedom, as well as its speed (angular rotation speed, and forward/backward speed). The limited speeds meant that the ZoomWalls took longer to arrive properly. In our demonstration, we asked users to walk slowly, which is a major limitation of this work. To address this, we propose several solutions, which might be integrated together.

Firstly, a more sophisticated actuator appropriate for use in this context would be an omni-directional robot capable of moving at higher speeds. Based on our informal simulation tests, faster actuators reduce latency. These would be especially appropriate for VR sports in gymnasium-sized spaces where people might be running. On the other hand, faster robots may introduce new safety concerns or anxiety on the part of users.

Secondly, if more ZoomWalls (e.g. 7 walls) can be deployed, faster-walking speed would be supported by applying different standby strategies for differently managed walls, e.g, three walls around user and four walls for faraway potential targets.

Thirdly, it is usable to design the narrative experience to encourage slower walking or different locomotion styles altogether (e.g., jumping, sliding feet etc.). Our participants also agreed our approach that the slow walking is the part of the fun experience. TilePops used diegetic visual effects to provide users with the status of pop preparation, which temporarily impacts the user's experience. Another technique may be to apply a higher translation gain to users walking to let them experience faster walking in the VR than the real [23, 32], which might maintain their immersive VR experience.

Path planning algorithm. We used a rudimentary point-to-point path-finding algorithm (RVO) that accounted for whether the ZoomWall would collide with another ZoomWall (or the user). More sophisticated planning algorithms would be able to predict possible collisions, and plan more ideal routes. In a more general case, where each ZoomWall is differentiated based on other kinds of haptic capabilities, this planning algorithm would need to account for these capabilities in relation to the needs of the virtual simulation (e.g. a half-height ZoomWall is needed for a window; a full-sized ZoomWall would be inappropriate since it does not provide the void of where the window hole is).

Shape-changing surfaces. The surfaces of the ZoomWalls could be replaced by shape-changing surfaces (e.g. [12, 36]), which would provide a more sophisticated physical rendering of the virtual model. One might imagine that the wall width is extendable (e.g., using a roll-screen), which provides wider continuous wall surfaces that can be swept by the user's hand. If the wall surface is deformable (e.g, by pin-array actuators [36], this would provide additional feedback for simple virtual objects (e.g. a button that depresses when touched).

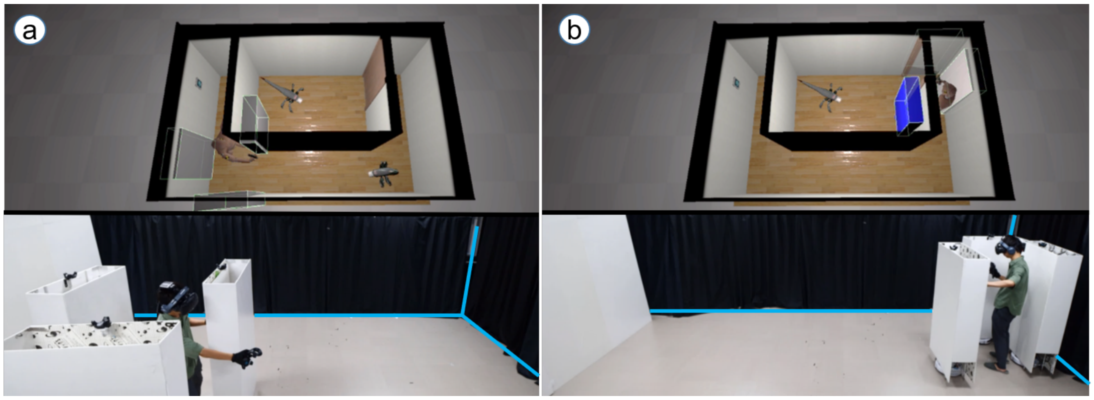

Figure 11. A first person shooter game played with the ZoomWall system. Here, three ZoomWalls work together to provide the haptic infrastructure of the environment. (a) A user takes cover behind a corner of a corridor, where the ZoomWalls represent the corner and the walls of the corridor. (b) At the end of the corridor, the user opens a sliding door (again, represented by a ZoomWall).

Touch prediction algorithms. Our prediction algorithm was designed for a simple room and mazes with perpendicular walls. For more complex arbitrary spaces and rooms, including curved wall or diagonal walls, the SVM would need to be retrained. A bubble cursor-like approach [14] that dynamically rescales the bounding box rather than relying on static-sized bounding box might also be an effective approach.

While the current algorithm was sufficient for our explorations, the prediction failed when the user touched a side wall while facing the frontal wall due to the strict assumption of our motion ML design. This failure could be addressed by Bounding Box with three ZoomWalls, but cannot always be resolved in a two ZoomWalls solution. To address this issue, a more sophisticated motion ML should be established using additional behavioral data (e.g. eye gaze).

Anxiety about invisible moving robots. Even though our ZoomWalls system has a number of fail-safes to prevent them from colliding with users, our participants still felt somewhat uneasy knowing that robots they could not see were moving around them. This is perhaps understandable, since the participants could hear the whirring of the robots, which were noisy enough to be heard over the sound/music from the VR experience. Furthermore, because the ZoomWalls move in ways that are essentially unpredictable to the users, they represent potentially unexpected obstacles on the ground that the user might trip on. This unease was probably somewhat exacerbated by the fact that the ZoomWalls essentially follow the user around as they walk.

In the VR simulation, the actual positions of the ZoomWalls are hidden deliberately, but participants asked for ways to be able to see the positions to ease their uncertainty. Others have explored this idea of showing objects and people from the real world in the VR simulation (e.g. [47, 31, 13, 55]), and this shows promise. As described earlier, our system already supports displaying the positions via a debug/safety mode, though our rationale for hiding their position was because we believed that users would feel less immersed in the simulation. On the other hand, if participants were regularly concerned about bumping into the ZoomWalls, it is hard to say that they felt fully immersed in the VR experience.

One possibility is visualizing the ZoomWalls with semi-transparent avatars or as diegetic objects in the simulation, but only doing so when the user is in danger of bumping into them. This approach is reminiscent of the "seamful design" approach advocated by many ubicomp researchers (e.g. Chalmers [8], Broll [6], and Weiser [51, 50]), where rather than hide the deficiencies or "gaps" in the system, one reveals them to the user so the user can make choices about how to interact with the technology. Taken to the logical conclusion, a seamful design approach would reveal in the VR simulation which walls were materialized (i.e. ready to be touched—a ZoomWall is in place), and which were not yet ready to be touched (i.e. a ZoomWall is dispatched, but not yet in place). These are interesting avenues for further design exploration.

CONCLUSION

ZoomWalls simulates haptic infrastructure for VR worlds using semi-autonomous robots. The ZoomWall units dynamically position and orient themselves using a just-in-time approach based on the user's movements within the virtual space. We explore two predictor algorithms that predict which walls will be touched, where the ZoomWalls are dispatched based on these predictions. Based on simulations, we show that for simple spaces where the units are able to move quickly enough, only two or three ZoomWall units are needed to provide less than 0.2s latency. Based on a user experience evaluation with 12 participants, we found that the ZoomWall system enhances users' immersion into the VR environment. We also learned that participants do not always feel comfortable not being able to see the ZoomWall units, and discuss techniques to address these shortcomings in future applications of this approach. Together, our explorations suggest that, given appropriate hardware, the ZoomWall approach is viable for simulating a wide range of haptic experiences for VR worlds without necessitating a large number of props, or human labour to rapidly assemble/move/reassemble props on the fly—an approach we call *autonomous mobile reusable props*.

ACKNOWLEDGEMENTS

This work was partially supported by JSPS Kakenhi (17K00262, 19KK0258), and the Natural Sciences and Engineering Research Council of Canada (RGPIN-2017-04883).

REFERENCES

[1] Muhammad Abdullah, Minji Kim, Waseem Hassan, Yoshihiro Kuroda, and Seokhee Jeon. 2017. Hapticdrone: an encountered-type kinesthetic haptic interface with controllable force feedback: initial example for 1D haptic feedback. In *Adjunct Proceedings of Symposium on User Interface Software and Technology (UIST)*. 115–117.

[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 Conference on Human Factors in Computing Systems (CHI)*. 1–13.

[3] Bruno Araujo, Ricardo Jota, Varun Perumal, Jia Xian Yao, Karan Singh, and Daniel Wigdor. 2016. Snake charmer: physically enabling virtual objects. In *Proceedings of Conference on Tangible Embedded and Embodied Interaction (TEI)*. 218–226.

[4] 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 Conference on Human Factors in Computing Systems (CHI)*. 1968–1979.

[5] Miguel Borges, Andrew Colquhoun Symington, Brian Coltin, Trey Smith, and Rodrigo Ventura. 2018. HTC VIVE: analysis and accuracy improvement. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2610–2615.

[6] Gregor Broll and Steve Benford. 2005. Seamful design for location-based mobile games. In *Proceedings of International Conference on Entertainment Computing*. 155–166.

[7] Frederick P Brooks. 1999. What's real about virtual reality? *IEEE Computer Graphics and Applications* 19, 6 (1999), 16–27.

[8] Matthew Chalmers. 2003. Seamful design and ubicomp infrastructure. In *Proceedings of Ubicomp Workshop at the crossroads: The interaction of HCI and systems issues in Ubicomp*.

[9] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D Wilson. 2017. Sparse haptic proxy: touch feedback in virtual environments using a general passive prop. In *Proceedings of Conference on Human Factors in Computing Systems (CHI)*. 3718–3728.

[10] Lung-Pan Cheng, Eyal Ofek, Christian Holz, and Andrew D. Wilson. 2019. VRoamer: generating on-the-fly VR experiences while walking inside large, unknown real-World building environments. In *Proceedings of IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 359–366.

[11] 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 Symposium on User Interface Software & Technology (UIST)*. 417–426.

[12] Daniel Fitzgerald and Hiroshi Ishii. 2018. Mediate: a spatial tangible interface for mixed reality. In *Extended Abstract of Conference on Human Factors in Computing Systems (CHI)*. 635:1–6.

[13] Sarthak Ghosh, Lauren Winston, Nishant Panchal, Philippe Kimura-Thollander, Jeff Hotnog, Douglas Cheong, Gabriel Reyes, and Gregory D Abowd. 2018. NotifiVR: exploring interruptions and notifications in virtual reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1447–1456.

[14] Tovi Grossman and Ravin Balakrishnan. 2005. The Bubble Cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of Conference on Human Factors in Computing Systems (CHI)*. 281–290.

[15] Tomasz Hachaj and Marek R. Ogiela. 2019. Head motion classification for single-accelerometer virtual reality hardware. In *Proceedings of IEEE International Conference on Frontiers of Signal Processing (ICFSP)*. 45–49.

[16] Zhenyi He, Fengyuan Zhu, and Ken Perlin. 2017. PhyShare: sharing physical interaction in virtual reality. In *Adjunct Proceedings of Symposium on User Interface Software and Technology (UIST)*. 17–19.

[17] Ryokichi Hirata, Hiroshi Hoshino, Taro Maeda, and Susumu Tachi. 1996. A force and shape display for virtual reality system. *Transactions of the Virtual Reality Society of Japan* 1, 1 (1996), 32–32.

[18] 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 International Conference on Mobile and Ubiquitous Multimedia*. 7–18.

[19] Yuuki Horiuchi, Yasutoshi Makino, and Hiroyuki Shinoda. 2017. Computational foresight: forecasting human body motion in real-time for reducing delays in interactive system. In *Proceedings of International Conference on Interactive Surfaces and Spaces (ISS)*. 312–317.

[20] Hsin-Yu Huang, Chih-Wei Ning, Po-Yao Wang, Jen-Hao Cheng, and Lung-Pan Cheng. 2020. Haptic-Go-Round: a surrounding platform for encounter-type haptics in virtual reality experiences. In *Proceedings of Conference on Human Factors in Computing Systems (CHI)*. 349:1–10.

[21] 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.

[22] Hiroo Iwata, Hiroaki Yano, Hiroyuki Fukushima, and Haruo Noma. 2005. CirculaFloor. *IEEE Computer Graphics and Applications* (2005), 64–67.

[23] Omar Janeh, Eike Langbehn, Frank Steinicke, Gerd Bruder, Alessandro Gulberti, and Monika Poetter-Nerger. 2017. Walking in virtual reality: Effects of manipulated visual self-motion on walking biomechanics. *ACM Transactions on Applied Perception* 14, 2 (2017), 12:1–15.

[24] Caroline Jay and Roger Hubbold. 2005. Delayed visual and haptic feedback in a reciprocal tapping task. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (World Haptics)*. 655–656.

[25] Yaesol Kim, Hyun Jung Kim, and Young J Kim. 2018. Encountered-type haptic display for large VR environment using per-plane reachability maps. *Computer Animation and Virtual Worlds* 29, 3-4 (2018), e1814.

[26] Yoshiki Kudo, Kazuki Takashima, Morten Fjeld, and Yoshifumi Kitamura. 2018. AdapTable: extending reach over large tabletops through flexible multi-display configuration. In *Proceedings of International Conference on Interactive Surfaces and Spaces (ISS)*. 213–225.

[27] Eike Langbehn, Paul Lubos, and Frank Steinicke. 2018. Redirected spaces: going beyond borders. In *Proceedings of IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 767–768.

[28] Xiuhua Liu, Zhihao Zhou, and Qining Wang. 2018. Real-Time onboard human motion recognition based on inertial measurement units. In *Proceedings of IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. 724–728.

[29] Jed Looker and Thomas Garvey. 2015. Reaching for holograms: assessing the ergonomics of the Microsoft Hololens 3D gesture known as the Air Tap. In *International Design Congress*. 504–511.

[30] Néstor Andrés Arteaga Martin, Victor Mittelstädt, Michael Prieur, Rainer Stark, and Thomas Bär. 2013. Passive haptic feedback for manual assembly simulation. *CIRP Conference on Manufacturing Systems* 7 (2013), 509–514.

[31] Mark McGill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. 2015. A dose of reality: overcoming usability challenges in vr head-mounted displays. In *Proceedings of Conference on Human Factors in Computing Systems (CHI)*. 2143–2152.

[32] Betty J Mohler, William B Thompson, Sarah H Creem-Regehr, Herbert L Pick, and William H Warren. 2007. Visual flow influences gait transition speed and preferred walking speed. *Experimental Brain Research* 181, 2 (2007), 221–228.

[33] Sharif Razzaque, Zachariah Kohn, and Mary C. Whitton. 2001. Redirected Walking. In *Eurographics*.

[34] Robert Shewaga, Alvaro Uribe-Quevedo, Bill Kapralos, and Fahad Alam. 2020. A comparison of seated and room-scale virtual reality in a serious game for epidural preparation. *IEEE Transactions on Emerging Topics in Computing* 8, 1 (2020), 218–232.

[35] Adalberto L Simeone, Eduardo Velloso, and Hans Gellersen. 2015. Substitutional reality: using the physical environment to design virtual reality experiences. In *Proceedings of Conference on Human Factors in Computing Systems (CHI)*. 3307–3316.

[36] 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 Conference on Human Factors in Computing Systems (CHI)*. 1–13.

[37] Mel Slater, Martin Usoh, and Anthony Steed. 1995. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Transactions on Computer-Human Interaction (TOCHI)* 2, 3 (1995), 201–219.

[38] Qi Sun, Anjul Patney, Li-Yi Wei, Omer Shapira, Jingwan Lu, Paul Asente, Suwen Zhu, Morgan Mcguire, David Luebke, and Arie Kaufman. 2018. Towards virtual reality infinite walking: dynamic saccadic redirection. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 67.

[39] Ryo Suzuki, Hooman Hedayati, Clement Zheng, James L. Bohn, Daniel Szafir, Ellen Yi-Luen Do, Mark D. Gross, and Daniel Leithinger. 2020a. RoomShift: Room-Scale Dynamic Haptics for VR with Furniture-Moving Swarm Robots. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. 396:1–11.

[40] Ryo Suzuki, Ryosuke Nakayama, Dan Liu, Yasuaki Kakehi, Mark D. Gross, and Daniel Leithinger. 2020b. LiftTiles: constructive building blocks for prototyping room-scale shape-changing interfaces. In *Proceedings of Conference on Tangible, Embedded, and Embodied Interaction (TEI)*. 143–151.

[41] Kazuki Takashima, Yusuke Asari, Hitomi Yokoyama, Ehud Sharlin, and Yoshifumi Kitamura. 2015. MovemenTable: The design of moving interactive tabletops. In *Proceedings of IFIP Conference on Human-Computer Interaction (INTERACT)*. 296–314.

[42] 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 Conference on Designing Interactive Systems (DIS)*. 796–806.

[43] 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 Symposium on User Interface Software and Technology (UIST)*. 639–649.

[44] Martin Usoh, Kevin Arthur, Mary C Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P Brooks Jr. 1999. Walking> walking-in-place> flying, in virtual environments. In *Proceedings of Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 359–364.

[45] Jur Van den Berg, Ming Lin, and Dinesh Manocha. 2008. Reciprocal velocity obstacles for real-time multi-agent navigation. In *Proceedings of IEEE International Conference on Robotics and Automation*. 1928–1935.

[46] The Void. 2019. (2019). `https://www.thevoid.com`.

[47] Julius von Willich, Markus Funk, Florian Müller, Karola Marky, Jan Riemann, and Max Mühlhäuser. 2019. You invaded my tracking space! using augmented virtuality for spotting passersby in room-scale virtual reality. In *Proceedings of Conference on Designing Interactive Systems (DIS)*. 487–496.

[48] Emanuel Vonach, Clemens Gatterer, and Hannes Kaufmann. 2017. VRRobot: robot actuated props in an infinite virtual environment. In *Proceedings of IEEE Virtual Reality (VR)*. 74–83.

[49] Yuntao Wang, Zichao (Tyson) Chen, Hanchuan Li, Zhengyi Cao, Huiyi Luo, Tengxiang Zhang, Ke Ou, John Raiti, Chun Yu, Shwetak Patel, and Yuanchun Shi. 2020. MoveVR: enabling multiform force feedback in virtual reality using household cleaning robot. In *Proceedings of Conference on Human Factors in Computing Systems (CHI)*. 1–12.

[50] Mark Weiser. 1993. Ubiquitous computing. *Computer* 10 (1993), 71–72.

[51] Mark Weiser. 1994. Creating the invisible interface:(invited talk). In *Proceedings of Symposium on User Interface Software and Technology (UIST)*.

[52] Bob G Witmer and Michael J Singer. 1998. Measuring presence in virtual environments: a presence questionnaire. *Presence* 7, 3 (1998), 225–240.

[53] Erwin Wu and Hideki Koike. 2019. FuturePose-mixed reality martial arts training using real-time 3D human pose forecasting with a RGB camera. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV)*. 1384–1392.

[54] Kotaro Yamaguchi, Ginga Kato, Yoshihiro Kuroda, Kiyoshi Kiyokawa, and Haruo Takemura. 2016. A non-grounded and encountered-type haptic display using a drone. In *Proceedings of Symposium on Spatial User Interaction (SUI)*. 43–46.

[55] Keng-Ta Yang, Chiu-Hsuan Wang, and Liwei Chan. 2018. ShareSpace: facilitating shared use of the physical space by both VR head-mounted display and external users. In *Proceedings of Symposium on User Interface Software and Technology (UIST)*. 499–509.

[56] Mark Yim, David G. Duff, and Kimon D. Roufas. 2000. PolyBot: a modular reconfigurable robot. In *Proceedings of IEEE International Conference on Robotics and Automation*. 514–520.

[57] Yasuyoshi Yokokohji, Junji Kinoshita, and Tsuneo Yoshikawa. 2001. Path planning for encountered-type haptic devices that render multiple objects in 3D space. In *Proceedings of IEEE Virtual Reality (VR)*. 271–278.

[58] Yasuyoshi Yokokohji, Nobuhiko Muramon, Yuji Sato, Takaharu Kikura, and Tsuneo Yoshikawa. 2004. Design and path planning of an encountered-type haptic display for multiple fingertip contacts based on the observation of human grasping behavior. In *Proceedings of IEEE International Conference on Robotics and Automation*. 1986–1991.

[59] 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 International Conference on Interactive Surfaces and Spaces (ISS)*. 82–91.