iTurk: Turning Passive Haptics into Active Haptics by Making Users Reconfigure Props in Virtual Reality

Lung-Pan Cheng[1], **Li Chang**[12], **Sebastian Marwecki**[1], **Patrick Baudisch**[1]
[1]Hasso Plattner Institute, [2]ETH Zurich
{firstname.lastname}@hpi.de, changl@student.ethz.ch

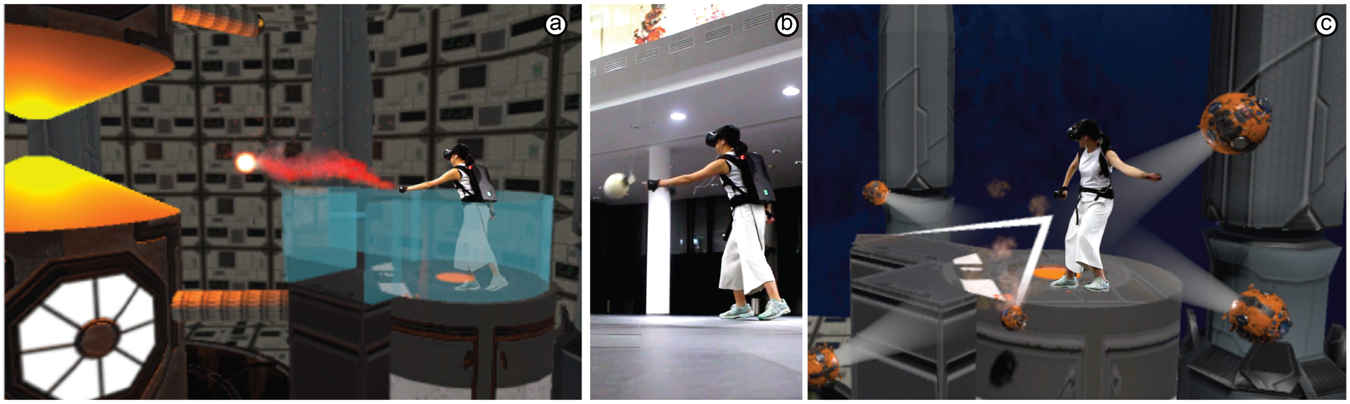

Figure 1: (a) As the user launches a plasma ball into the reactor, she feels the physical impact of hitting the prop. (b) The haptic feedback comes from a physical prop on a pendulum. The user's hit, however, also sets the pendulum in motion. (c) When the user later fends off a group of flying droids, the system renders each one of them using one period of the swinging pendulum. Every one of the user's hits is not only a haptic experience, but also provides the impulse for the next attack. As a result, the experience feels alive, even though the user is the only animate entity in it.

ABSTRACT

We present a system that complements virtual reality experiences with passive props, yet still allows modifying the virtual world at runtime. The main contribution of our system is that it does not require any actuators; instead, our system employs *the user* to reconfigure and actuate otherwise passive props. We demonstrate a foldable prop that users reconfigure to represent a suitcase, a fuse cabinet, a railing, and a seat. A second prop, suspended from a long pendulum, not only stands in for inanimate objects, but also for objects that move and demonstrate proactive behavior, such as a group of flying droids that physically attack the user. Our approach conveys a sense of a living, animate world, when in reality the user is the only animate entity present in the system, complemented with only one or two physical props. In our study, participants rated their experience as more enjoyable and realistic than a corresponding no-haptics condition.

Author Keywords

Human actuation; Haptics; Virtual Reality; Immersion;

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

ACM 978-1-4503-5620-6/18/04…$15.00
https://doi.org/10.1145/3173574.3173663

ACM Classification Keywords

H.5.2 [User Interfaces]: Haptic I/O.

INTRODUCTION

Ever since the conception of the virtual reality headset in 1968 [4], many researchers have argued that the next step in virtual reality has to be to allow users to not only see and hear, but also *feel* virtual worlds [5].

One main approach towards this revolves around the use of physical props, also known as *passive haptics* [21,28]. While simple prop-based systems require users to be mostly stationary [23], more elaborate systems allow users to move around freely in a space filled with physical props. Some systems achieved this effect based on projection [15]; others used head-mounted displays [14].

Unfortunately, the increased level of immersion provided by passive haptics is subject to limitations. First, a room filled with physical props tends to match only one specific virtual room. Redirected walking allows confronting users with the same prop repeatedly [18], yet the immutability of the props makes them hard to reuse. Thus, one room worth of a virtual world tends to require one room worth of physical space [2]. Second, passive props allow rendering only inanimate objects, which limits the perceived "liveliness" of the resulting virtual worlds.

To address the problems, researchers proposed reconfiguring props during use, either using robotics (*robotic graphics* [6], *snake charmer* [8]) or using human actuators (*TurkDeck* [2], *Mutual Turk* [3]). Both approaches work well, but also come at a price: the robotics approach requires reasonably complex equipment, while the human actuation approach requires human workers at every session.

In this paper, we present a new approach to dynamic prop-based virtual reality. It offers some of the functionality of much more complex systems, such as TurkDeck, but gets by requiring neither mechanical nor human actuators. The main idea behind our system is to let *the user* do the work.

iTurk

The key idea behind our system *iTurk* is that it makes users constantly reconfigure and animate otherwise passive props. By integrating these reconfiguration activities into the experience itself, iTurk hides them from the user. This allows iTurk to provide virtual worlds with constantly varying or even animated haptic effects, even though the only animate entity present in the system is the user.

To illustrate the system, we have created an experience that users explore using a head-mounted display (*HTC Vive* [22]) while physically walking through the tracking space (real walking [32]). During the experience, users physically interact with 10 different types of virtual objects, each of which is provided with a matching passive haptic effect. However, all haptic experiences are based solely on the *two* physical props shown in Figure 2, i.e., one foldable board [2] and one pendulum.

Figure 2: Our demo experience is built around only two props: (a) a board that can be reconfigured by the user and (b) a round prop suspended from the ceiling that starts to swing when pushed or hit by the user.

Figure 1 shows a segment from the demo experience. It focuses on the pendulum prop. (a) As the user launches a plasma ball into the reactor, she feels the physical impact of hitting the prop. (b) The haptic feedback comes from a physical prop on a pendulum. The user's hits, however, also sets the pendulum in motion. (c) When the user later fends off a group of flying droids, the system renders each one of them using one period of the swinging pendulum. Every one of the user's hits is not only a haptic experience, but also provides the impulse for the next attack.

The reuse of the pendulum across scenes illustrates the main idea behind iTurk, which is to make users reconfigure or, in this case, animate otherwise passive props, which makes these props more expressive and allows them to represent multiple virtual objects. To illustrate this point, we now show a larger portion of the same virtual experience.

Walkthrough

Our demo experience takes place in the 1940s at a hypothetical experimental reactor site. The user's mission is to trigger the self-destruction of a reactor and to leave the site before it explodes. In this walkthrough, we typeset every object physically represented by a prop in **bold**. We typeset every step in which the user reconfigures a prop in ***bold italics***.

As shown in Figure 3, (a) the user joins the experience in a dimly lit room. Opening the door requires the user to bridge the electric cable that goes through the ceiling lamp. (b) The user tries, but the **lamp** and its **cable** hang too high to be reached. The user notices a **suitcase** in the corner, ***pushes*** it under the lamp and ***lays*** it flat. (c) Stepping onto the **suitcase** allows the user to reach the **cable** and push its **two severed ends** together. This closes the electric circuit, temporarily turns the lights on, and opens the door—allowing the user to leave the room and enter the hallway. The hallway provides access to the adjacent room.

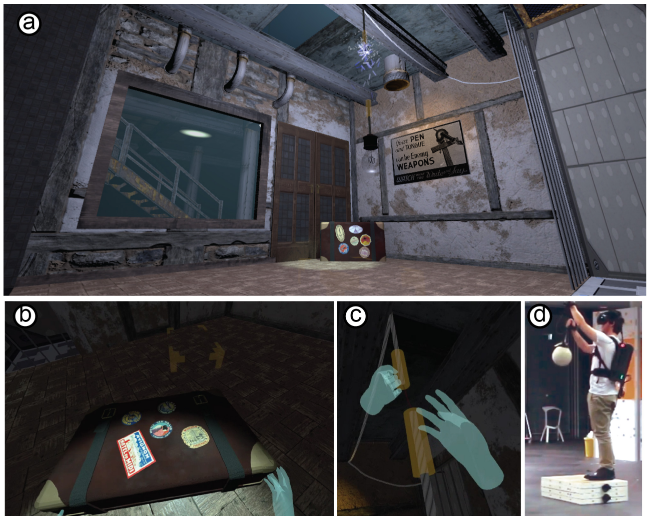

Figure 3: (a) The first room of our example experience requires users to (b) move a suitcase, lay it flat, and (c) step on to short circuit a cable. (d) This is supported by two physical props, i.e., a folded board and a pendulum.

As shown in Figure 4, the adjacent room is physically overlapping with the first room. We achieve this by implementing the impossible spaces [26], making the doors as portals, i.e., the geometry of the second room is represented separately in the VR system and when passing the door the user is unknowingly teleported [7]. This overlap between rooms is crucial as it allows the new room to contain the same physical props as the previous room. However, what used to be a lamp and a suitcase a minute ago are now a spotlight and a fuse box.

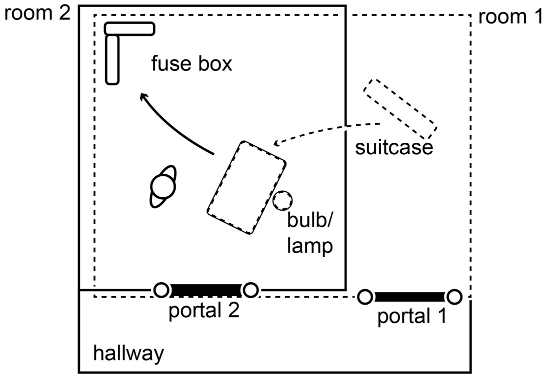

Figure 4: The fuse room is designed to geometrically overlap with the first room. This allows the two physical props to be located in *both* rooms.

As shown in Figure 5, (a) a **spotlight** illuminates a **fuse box** apparently ripped out of its base. (b) The user ***erects*** the **fuse box** and ***places*** it back into its base. Nothing happens. (c) The user ***opens*** the **fuse box**, which reveals an **on/off button.** The user pushes the **on/off button**, which restores power in the building and turns the lights on. (d) These interactions are supported by the same physical prop that served as suitcase in the previous room.

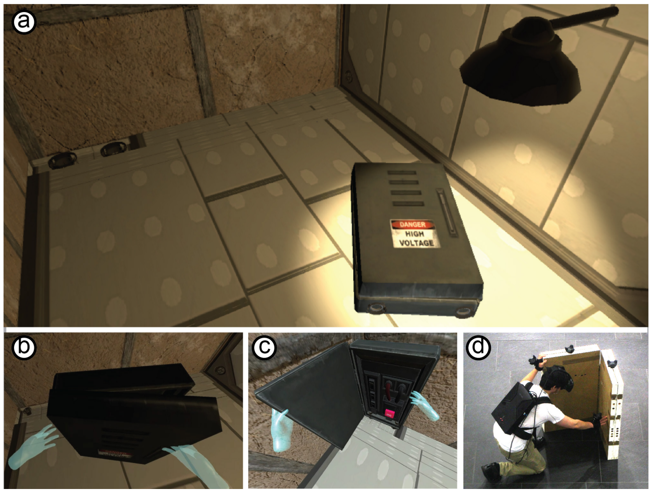

Figure 5: (a) The second room requires users to move and erect a fuse box, (b) open it, and (c) push the "on" button. (d) This is supported by the same physical prop that served as suitcase in the previous room.

Via the hallway, the user reaches a third door that provides the user with access to a huge futuristic reactor room. As shown in Figure 6, the reactor room is overlapping with the first two rooms, which again provides the user with access to the same two physical props. The board turns into a railing and the pendulum, which so far had only served as a passive prop, now allows us to represent a moving object.

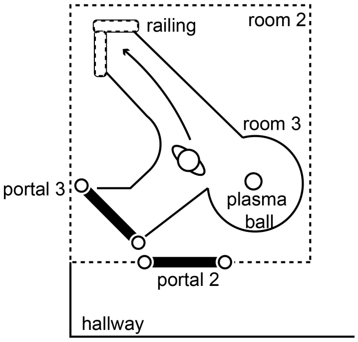

Figure 6: The reactor room also overlaps with the other rooms.

As shown in Figure 7a, the user walks to the end of the walkway and activates the two **pressure panels** on the **railing.** (b) This lifts up the cylindrical cover behind the user revealing the emergency shut down mechanism: a floating **plasma ball**. The user ***hits*** the **plasma ball** towards the reactor. It flies off into the reactor, where it triggers the self-destruct mechanism. (c) The reactor responds by exploding and **reactor shrapnel** is flying all over the place. To evade the **shrapnel** the user runs back to the hallway. (d) The plasma ball and all shrapnel are rendered using the pendulum prop.

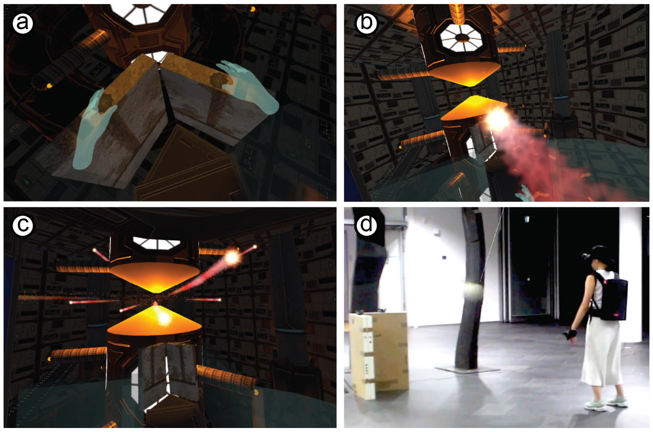

Figure 7: (a) The reactor requires users to press and hold two buttons on the railing, in order to reveal the plasma ball, which (b) users launch by hitting it towards the reactor. (c) The rector explosion causes shrapnel to fly towards the user. (d) The plasma ball and all shrapnel are rendered using the pendulum prop.

Figure 8 shows a few moments of the remainder of the experience. (a) The user goes back to the disintegrating fuse room and (b) ***removes*** the **fuse box** from its base to cut power, while evading more **shrapnel**. After fending of half a dozen **droids** (Figure 1), (c) the user catches the **command module** (which stops the pendulum) and uses it to call the escape pod. (d) The user enters the escape pod, sits down on the **pilot's seat,** which (e) is rendered once more by the foldable board and takes off to safety.

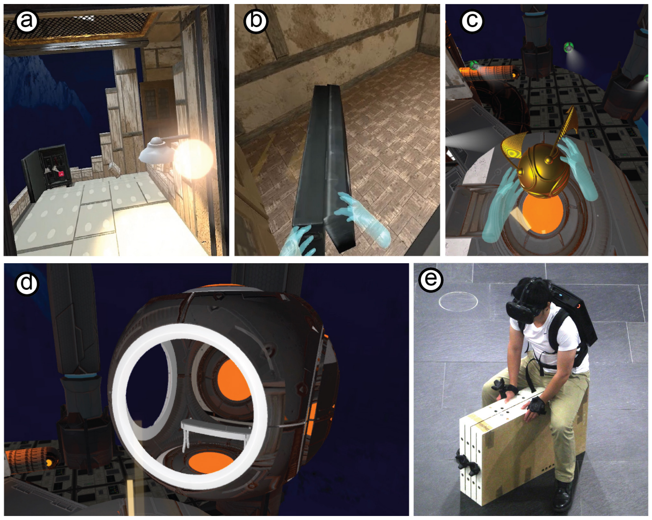

Figure 8: The remainder of the experience.

iTURK'S UNDERLYING PATTERNS

Our demo experience features one example each of iTurk's two main types of props, i.e., *reconfigurable props* and *animated props*. In this section, we discuss the patterns in which these props are deployed.

Reconfigurable props := {reconfigure, use, remap}

iTurk deploys passive props in what we call "reconfigure-use-remap" cycles.

(1) Reconfigure: users encounter an object in the virtual world the state of which prevents users from progressing the story arc. The fuse box, for example, is not working because it has been removed from its socket or users cannot access the buttons inside the fuse box because the fuse box is closed. When users realize that the prop is not in the required state, they reconfigure it. It is in this reconfiguration step where iTurk is "exploiting" the user for manual labor. Without such user-based reconfiguration, this step would have to be performed by a mechanical or human actuator—which is exactly the effort iTurk is saving.

In our demo experience, we used a foldable board as prop in order to illustrate the intentional analogy to *TurkDeck* [2] (Figure 9). Other experiences may use props with more expressiveness, such as large-scale Lego bricks or props with less expressiveness, including monolithic props that can only be moved or flipped.

(2) Use: users can now perform the action that drives the story arc forward. In our experience, the user stands, sits, leans onto the prop, presses buttons on it, etc. (Figure 9).

(3) Remap: Once used, iTurk maps the physical props to a fresh virtual object. In our demo experience, iTurk achieved this by steering users to another "room," yet one that overlaps with the previous one, so that the physical props are again present and can be used to represent a fresh set of virtual objects. For large physical spaces, this can be achieved using redirected walking [26]. The reason we instead used *portals* [7] disguised as doors [9], was primarily to allow our experience to run in a very compact tracking volume (4x4m).

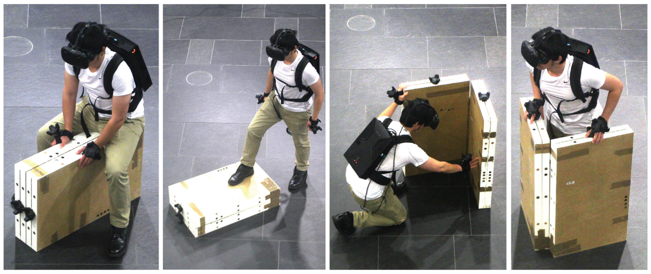

Figure 9: Some of the uses of the foldable board in our demo experience.

Animated props := {reconfigure/use, remap}

iTurk's animated props can generally be employed the same way passive props are, as we demonstrated in the example of users bridging the electric contact in the first room (Figure 10a). In addition, however, animated props can also be operated in a way that merges the *reconfigure* step and the *use* step into a single interaction, resulting in *reconfigure-by-use-remap* cycles.

(1) Reconfigure-by-use: as before, users encounter an object in the virtual world and interact with it in order to progress the story arc. For animated props, reconfiguration means can be any of the interactions shown in Figure 10, i.e., (b) animate the prop, (c) keep the prop animated, and (d) stop the prop.

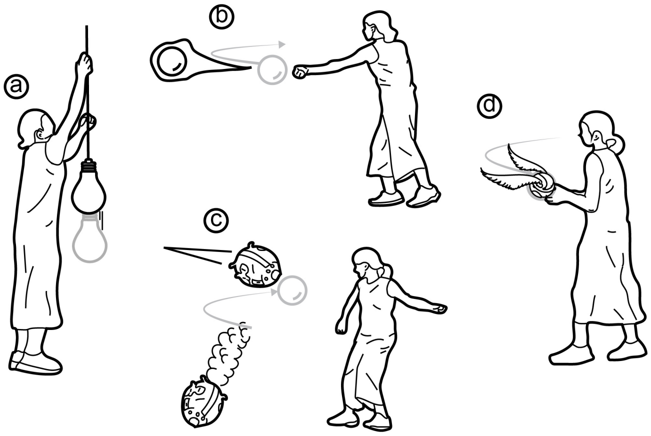

Figure 10: Our demo experience contains examples of (a) inanimate use, (b) animate prop, (c) keep prop animated, and (d) stop the prop.

(2) Remap: As before, iTurk maps the physical props to a fresh virtual object when users move on to a different room. With animated props, however, it is not the configuration that persists across rooms, but the impulse. This allows the system to implement interactions with arbitrary combinations of before and after states. As suggested by the gray arrows in Figure 10, this allows experience designers to concatenate any two interactions where the *before* state of the prop in the new room matches the *after* state of the prop in the previous room.

Also note how entering a room with objects already in motion (Figure 10c and d) are probably experience designers' best tools for conveying a virtual world that is alive.

The qualities of the pendulum

The pendulum is a well-designed mechanism in that it allows a good amount of virtual objects to be mapped to it. In particular, it allows simulating a surprisingly wide range of object trajectories (for details, see section "implementation"). Since users can touch the prop only while it is in close proximity, experience designers can make up the rest of the trajectory. In Figure 1a, we used this to make the plasma ball find its way into the reactor. In Figure 1c, we used this to make the sentinel droids come to the user from wherever they were hovering and to let destroyed droids drop into the void.

Another key quality of the pendulum is that it returns to the user in a way that is hard for humans to predict, as (1) small variations in hitting angle lead to large variations in prop trajectory, (2) the prop returns after a hard-to-predict time delay, and (3) the prop returns from a hard-to-predict angle. It is this unpredictability that allows experience designers to "sell" each instance of the pendulum arriving as a different object. We exploited this in Figure 1 to simulate an attack by multiple droids. By letting multiple droids hover for a while before we map them to the pendulum, we reinforce the illusion that *each* droid individually is an animate object.

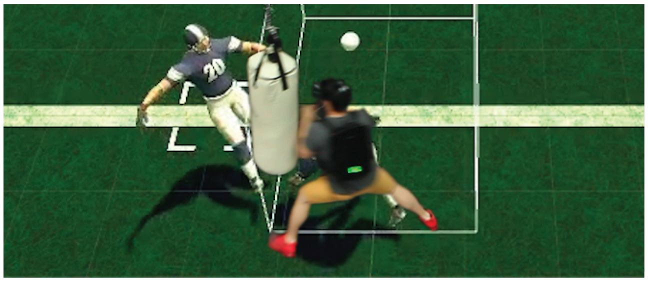

Figure 11: A football experience by replacing the end effector of the pendulum with a punching bag,

The pendulum offers yet additional versatility. Experience designers may, for example, vary its length. A shorter pendulum results in faster interaction; a pendulum the trajectory of which is blocked half way up the tether swings with variable frequency, making it even less predictable. Or we may replace the passive prop at its end. Replacing the prop with a 40kg punching bag, for example, allowed us to implement a simple American football tackling simulator (Figure 11)

CONTRIBUTION, BENEFIT, & LIMITATIONS

The main contribution behind iTurk is the idea of *using users* to reconfigure and animate otherwise passive props in virtual reality. User-reconfigurable props allow us to create arbitrary sequences of rooms, each of which reuses the same physical space and the physical props. Animated props, such as the pendulum, allow rendering animate objects, which brings liveliness to the resulting virtual worlds. This allows iTurk to realize some of the benefits that have traditionally only been achieved with passive haptics either actuated by mechanical actuators (*robotic graphics* [6]) or human actuators (*TurkDeck* [2]).

The main limitation of iTurk is that designing experiences requires additional care, as each scene needs to create the physical pre-conditions for the following scene. It does allow telling encompassing stories nonetheless, as we illustrated with the demo experience presented above. iTurk shares most of the regular virtual reality limitations: potential hazards of operating props (none of which occurred during our study), tracking lost, etc. The experience designers should take these into consideration.

RELATED WORK

The work that is presented in this paper is based on haptics for real-walking virtual reality, passive haptics, and human actuation.

Active Haptics for real-walking virtual reality

While there is a large corpus that demonstrates how to produce tactile effects suitable for virtual reality (e.g., the wind effect *AIREAL* [12], or *normal touch* [11]), producing force feedback is more challenging.

Seminal work by McNeely introduced the idea of using a robotic arm to reposition a single prop so as to simulate a surface wherever the user tries to touch [6].

Lopes et al. proposed the use of electrical muscle stimulation as a mean to simulate the impact (*impacto* [10]) and the resistance of objects in real-walking virtual reality (*EMS Walls* [9]).

Passive Haptics

Previous work shows that props, also known as passive haptics [28] can enhance the sense of presence. In a study by Hoffman [13], participants in virtual environment could guess an object's properties, such as the weight of a teapot more accurately if it had been given a physical representation. In a study by Insko et al. [14], participants immersed in a virtual environment crossed a virtual pit by balancing a ledge. Behavioral presence, heart rate, and skin conductivity were affected more, if the ledge was created using a physical wooden plank.

Several passive haptics systems used physical props in real walking environments. Low et al., for example, use Styrofoam walls onto which they project augmented reality experiences [15]. Similarly, mixed reality for military operations in urban terrain [16] uses passive haptics to add a haptic sense to otherwise virtual objects, terrain, and walls.

FlatWorld integrates large props into a physical world; *between* experiences these props can be rearranged to match the next virtual world [17]. Kohli et al. use redirected walking to allow users to encounter a stationary prop at different virtual locations [18]. In Substitutional Reality [19], researchers conducted a study on how the mismatch between physical and virtual props can break believability. Sparse Haptic Proxy [20] used a hemisphere prop combining with haptic retargeting [25] to give touch feedback whenever users touch the virtual object.

Reconfigurable props were a new class of passive haptics which has more flexible use. Aguerreche et al. [27] Manually reconfigurable props. Self-actuated reconfigurable props such as Haptobend [30] and Shifty [29] used machine actuators to change the geometry and the weight distribution.

Human actuation

Since the size and weight of mechanical machinery tends to be proportional to what they actuate, the use of mechanical haptic equipment tends to be constrained to arcades and lab environments. In order to bring haptics to a wider audience, researchers proposed creating haptic effects using *human actuators* [1].

TurkDeck applied the concepts of human actuation to real walking [2]. The system allows a single user to explore a virtual reality experience that is brought to life by ten human actuators that continuously rearrange physical props and apply forces to the user. Mutual Turk [3] demonstrates how two users can serve as human actuators to each other.

One way to think of iTurk is as a variation of human actuation in which users actuate themselves. To enable this, iTurk employs a range of techniques that hide the true origin of the reconfiguration and actuation from the user.

IMPLEMENTATION

To help readers replicate our results, we now present the implementation details.

Setup of virtual reality system

We developed iTurk software system in Unity 3D, running on a Zotac VR go backpack PC.

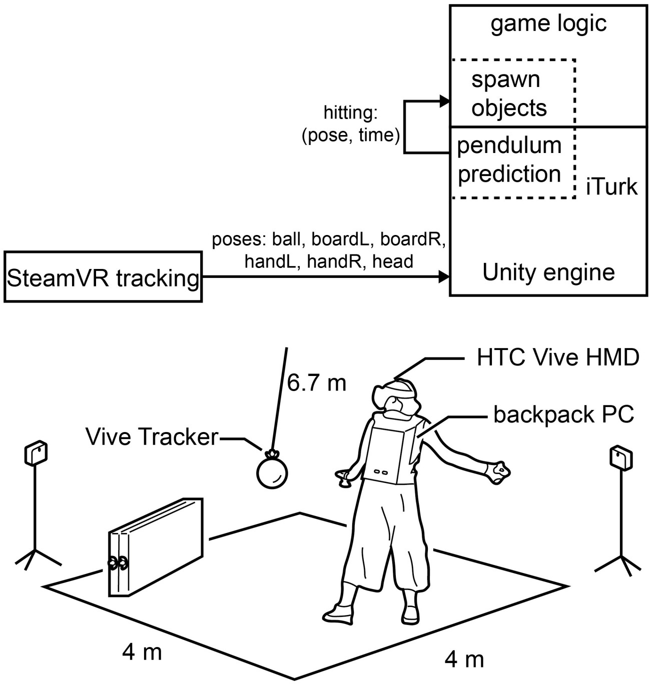

Figure 12: System diagram

Reconfigurable props and tracking

We use an HTC Vive system [22] for tracking as well as 5 Vive trackers for the props and the user's hands (Figure 13a).

The Pendulum

Figure 13b and c show the set-up of the pendulum. The prop consists of a volleyball and a tracker tied to a 6.7 m long braided fishing line (KastKing SuperPower braid fishing line 0.8 mm), resulting in a 5.12 second period. We ceiling-mounted the pendulum so as to swing 1m above the ground. The use of the very light string prevents the prop from vibrating and bobbing up-and-down, as is the case with heavier tethers, such as chains.

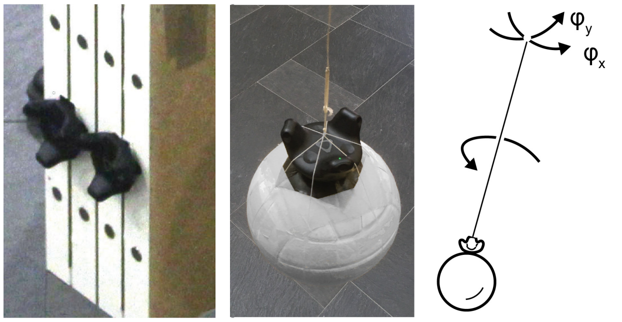

Figure 13: (a) *Two* Vive trackers allow the system to track the reconfigurable board in all its individual states. (b) The pendulum. (c) the state of the spherical pendulum is described by rotation angles on the pivot.

0. Launch Our prop implements a *spherical* pendulum, i.e., a pendulum that not only moves back and forth, but can also orbit, for overall number of two degrees of freedom. This is a desired feature, because it makes the path hard to predict for users, thereby allowing the experience designers to exercise a lot of control, including the ability to remap the single prop to multiple virtual objects in the same session. As illustrated by Figure 14, the system uses two user interactions to get a stationary pendulum to orbit.

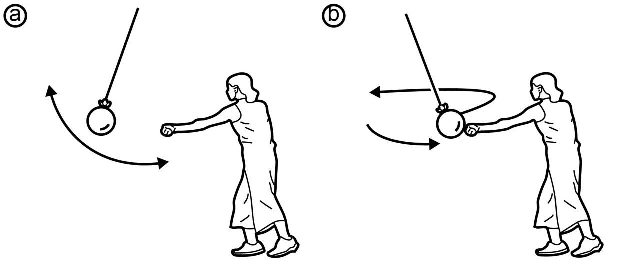

Figure 14: Making the pendulum reach its two degrees of freedom requires two user interactions: (a) The first hit makes the prop swing back and forth and (b) the second hit allows it to orbit.

1. Sensor reading The system obtains the current pendulum angle and the current angular speed from the Vive tracker attached to the prop. This gives us the two rotation angles along the two rotation axes on the pivot ϕ_x and ϕ_y as shown in Figure 13b.

2. Extrapolation The system extrapolates the pendulum's movement using the Euler-Lagrange equations for spherical pendulums.

$$\begin{bmatrix}\ddot{\phi}_x \\ \ddot{\phi}_y\end{bmatrix} = \begin{bmatrix}\frac{1}{l\cos\left(\phi_y(t)\right)}\left(-\mathrm{g}\sin(\phi_x(t)) + 2l\dot{\phi}_x(t)\dot{\phi}_y(t)\sin\left(\phi_y(t)\right)\right) \\ -\frac{1}{l}\left(\mathrm{g}\cos(\phi_x(t)) + l\dot{\phi}_x(t)\cos\left(\phi_y(t)\right)\sin\left(\phi_y(t)\right)\right)\end{bmatrix}$$

It then extrapolates using ordinary differential equations solver from the boost library [24] with the Fehlberg 78 stepper. This provides the system with an extrapolation of the pendulum's trajectory for the next 5 seconds, which accounts for one full period of the pendulum.

3. Hit trajectory The system picks the longest matching curve with low curvature as the trajectory during which the user is supposed to hit the prop. Figure 15 shows the hit trajectory.

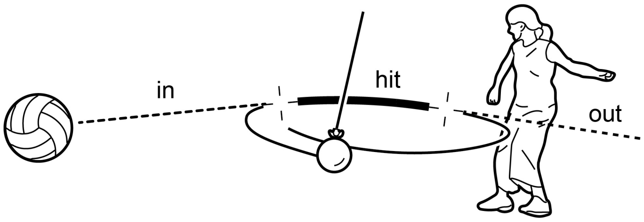

Figure 15: iTurk's pendulum movement behind the scene.

iTurk offers three modes of deciding hitting position: (1) *Closest point* tends to work best for standing in one position. (2) *Slowest point* tends to work best for rhythm games. (3) *Longest overlap,* as shown in Figure 15, tends to work best for ball games, such as tennis or baseball, as it gives users a longer time window to hit the ball.

For the demo experience presented earlier we do want users to succeed in order to drive the story line forward. The experience therefore picks the slowest point in the path as the center of this *hit trajectory* and also displays a preview of the trajectory to the user early. Other experiences may increase the difficulty by showing the trajectory later or not at all.

4. In trajectory The system creates a fake *in trajectory* connecting a visible virtual object to the *hit trajectory*. For that purpose it computes such trajectories for all candidate objects, such as all droids in Figure 1. It then picks the object that connects to the hit trajectory with the path of least maximum curvature. Based on the trajectory the system computes when to start the animation.

5. Out trajectory The system creates a fake *out trajectory*, e.g., by connecting the *hit trajectory* to a visible virtual object or simply by extrapolating the yaw direction. For the droids, for examples, the system just bends the trajectory back up to a free position on the droid "pool". The system uses a one/two-sided clamped spline to implement this.

6. Continuous updating during prop animation At the start time computed in step 4, the system starts animating the virtual object along the in trajectory. During the in trajectory, the system gradually morphs the virtual object's trajectory into the prop's trajectory.

The system's sensing and resulting extrapolation is generally reasonably accurate (~100 fps, < 3 cm errors when tracked). However, depending on how the user hit the prop last round, the prop may bounce, spin, or vibrate, which are harder to model and extrapolate. To guarantee a high-accuracy trajectory nonetheless, the system continuously re-reads the trackers and re-computes the trajectories of the pendulum based on a sliding window of sensor samples (we use the last 40 sensor readings = a time window of 400 milliseconds). When necessary, it continuously morphs any trajectory previews continuously to the updated state.

Our system uses some extra precaution to deal with the limitations of the tracking system. Because of the HTC Vive's limited tracking volume (5m diameter) users occasionally hit the prop so hard that it leaves the tracking volume. In this case the system runs the extrapolation on the sensor readings up to the point where it loses tracking. This generally works well. However, the Vive system also requires about 1-2 seconds (depending on the speed of the object from our testing) to start tracking again when an object re-enters the tracking volume and in rare cases the user hits the prop in that time window. While the system learns about the hit based on the marker's accelerometer, it lacks the positioning data required to compute the props *post hit trajectory*. In this case, the system informs the experience, which covers up the situation, e.g., by generating smoke or by turning the lights in the scene off until the tracking is back.

7. Hitting during the hit trajectory the user may or may not hit (or catch) the prop. The system learns about it instantaneously by detecting the resulting spike in the accelerometers in the tracker attached to the prop. The system now discards all further trajectories, gets another set of sensor readings as described in step 1, and uses those readings to compute a *post hit trajectory*. For the droids, for examples, the system simply takes the droid's actual trajectory and bends it downwards to simulate the droid falling.

At the same time, the system gets a new object on its way using *the same* set of sensor readings. This means that for some time, two virtual objects are moving at the same time, which helps make the illusion work that there are multiple living objects in the scene.

If the user did not hit the prop, the virtual object continues to follow the *out trajectory* and also a new virtual object is mapping to the prop and sent on its way.

USER STUDY

To validate our system, we conducted a user study in which participants experienced our demo world with and without iTurk haptic feedback. We hypothesised that iTurk's reconfigurable props would contribute to participants' experience and that the reconfigure step would not affect participants' sense of realism.

Interfaces There were two interface conditions. In the *iTurk* condition, users experienced the demo scene supported by the foldable board and the pendulum presented earlier. In the *no-haptics* control condition they experienced the same virtual experience, albeit without the props.

Procedure After a brief explanation on how to safely interact with the props and the goal of the game, participants put on the Vive HMD, two Vive trackers, and a Zotac VR go backpack PC. They received 1 minute of training during which they move the board around and hit the pendulum prop about 3-5 times. They then performed the experience in both interface conditions in counterbalenced order and filled in a questionnaire.

Figure 16: Participant during study

Participants We recruited 12 participants from our institute and 3 of them had used HMD before.

Result

Figure 17 shows the result. Overall, the participants highly enjoyed the experience (6.25/7-point Likert scale, 1= not at all, 7=very, Mann-Whitney U test U = 23.5, p<0.01). Participants experienced the virtual world as more realistic (5.5/7, U = 16.5, p<0.01) when in the iTurk condition.

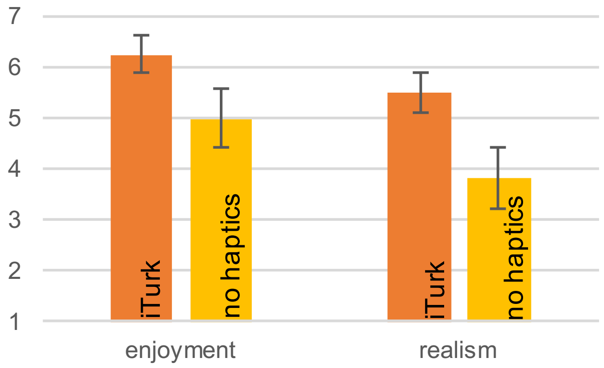

Figure 17: Participants rated their experience in the *iTurk* condition as significantly more enjoyable and realistic than in the *no-haptics* control.

We did receive positive comments from the participants. “The ball is really nice. I hit it only at one point and it started moving around and came back in unexpected directions” said p1. “I wasn’t expecting to feel the wind when the ball flew past me, and it really surprised me” said P4. “When I revisited the box-like object, I suspected it would be the same board. However, I didn’t recall that I put it in that position. It feels like it should have been a couple of meters away.” said P6. P8, P11 and P12 made similar comments.

One participant said that during the experience he did not notice that he was touching the same props. Two participants said that they realized that at the second time they touched the prop. The others suspected that because they were spectating the others but they said they could not tell either when they were in the experience.

We observed that the participants reconfigured the foldable board using different ways. Some of them moved the board to the destination before folding it, some of them flip it before moving it. Although in our system there is no correct order when reconfiguring the board, the process of reconfiguring can certainly be optimized and display to the user.

All the participants said that the system was running smoothly and they did not feel any offset.

From these study result, we conclude that iTurk is working. The general concept of user-based reconfiguration leads to more enjoyable experiences and can improve current home-edition virtual reality system.

CONCLUSION

In this paper, we presented iTurk, a system that complements virtual reality experiences with passive props, yet still allows modifying the virtual world at runtime. The main contribution behind iTurk is the idea of *using users* to reconfigure and animate otherwise passive props in virtual reality. First, user-based reconfiguration of props allows creating arbitrary sequences of rooms, each of which reuses the same physical space and the physical props. Second, animated props, such as the pendulum, allow rendering animate objects, which brings liveliness to the resulting virtual worlds. This allows iTurk to realize some of the benefits that have traditionally only been achieved with passive haptics such as actuated either by mechanical actuators (*Robotic graphics* [6] or human actuators (*TurkDeck* [2]).

As future work, we are planning on exploring self-reconfiguring props based on energy harvesting.

REFERENCES

1. Lung-Pan Cheng, Patrick Lühne, Pedro Lopes, Christoph Sterz, and Patrick Baudisch. 2014. Haptic turk: a motion platform based on people. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14). ACM, New York, NY, USA, 3463-3472. DOI=http://dx.doi.org/10.1145/2556288.2557101
2. Lung-Pan Cheng, Thijs Roumen, Hannes Rantzsch, Sven Köhler, Patrick Schmidt, Robert Kovacs, Johannes Jasper, Jonas Kemper, and Patrick Baudisch. 2015. TurkDeck: Physical Virtual Reality Based on People.

In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (UIST '15). ACM, New York, NY, USA, 417-426. DOI: https://doi.org/ 10.1145/2807442.2807463

3. Lung-Pan Cheng, Sebastian Marwecki and Patrick Baudisch. 2017. Mutual Human Actuation. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software & Technology* (UIST '17).
4. Ivan E. Sutherland. 1968. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I* (AFIPS '68 (Fall, part I)). ACM, New York, NY, USA, 757-764. DOI: https://doi.org/10.1145/1476589.1476686
5. Ivan E. Sutherland. 1965. The ultimate display. In *Proceedings of the Congress of the International Federation of Information Processing (IFIP)*, 506–508. https://doi.org/10.1109/MC.2005.274
6. W. A. McNeely, "Robotic graphics: a new approach to force feedback for virtual reality," Proceedings of IEEE Virtual Reality Annual International Symposium, Seattle, WA, 1993, pp. 336-341. DOI: 10.1109/VRAIS. 1993.380761
7. Doug A. Bowman, David Koller, and Larry F. Hodges. 1997. Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques. In *Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97)* (VRAIS '97). IEEE Computer Society, Washington, DC, USA, 45-.
8. Bruno Araujo, Ricardo Jota, Varun Perumal, Jia Xian Yao, Karan Singh, and Daniel Wigdor. 2016. Snake Charmer: Physically Enabling Virtual Objects. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '16). ACM, New York, NY, USA, 218-226. DOI: https://doi.org/10.1145/2839462.2839484
9. Pedro Lopes, Sijing You, Lung-Pan Cheng, Sebastian Marwecki, and Patrick Baudisch. 2017. Providing Haptics to Walls & Heavy Objects in Virtual Reality by Means of Electrical Muscle Stimulation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*(CHI '17). ACM, New York, NY, USA, 1471-1482. DOI: https://doi.org/10.1145/3025453.3025600
10. Pedro Lopes, Alexandra Ion, and Patrick Baudisch. 2015. Impacto: Simulating Physical Impact by Combining Tactile Stimulation with Electrical Muscle Stimulation. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (UIST '15). ACM, New York, NY, USA, 11-19. DOI: https://doi.org/10.1145/2807442.2807443
11. Hrvoje Benko, Christian Holz, Mike Sinclair, and Eyal Ofek. 2016. NormalTouch and TextureTouch: High-fidelity 3D Haptic Shape Rendering on Handheld Virtual Reality Controllers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (UIST '16). ACM, New York, NY, USA, 717-728. DOI: https://doi.org/10.1145/2984511.2984526
12. Rajinder Sodhi, Ivan Poupyrev, Matthew Glisson, and Ali Israr. 2013. AIREAL: interactive tactile experiences in free air. ACM Trans. Graph. 32, 4, Article 134 (July 2013), 10 pages. DOI=http://dx.doi.org/10.1145/2461912.2462007
13. H. G. Hoffman, "Physically touching virtual objects using tactile augmentation enhances the realism of virtual environments," Proceedings. IEEE 1998 Virtual Reality Annual International Symposium (Cat. No.98CB36180), Atlanta, GA, USA, 1998, pp. 59-63. DOI: 10.1109/VRAIS.1998.658423
14. Insko, B.E. *Passive haptics significantly enhances virtual environments*. Dissertation at University of North Carolina at Chapel Hill, 2001.
15. Kok-Lim Low, Greg Welch, Anselmo Lastra, and Henry Fuchs. 2001. Life-sized projector-based dioramas. In *Proceedings of the ACM symposium on Virtual reality software and technology* (VRST '01). ACM, New York, NY, USA, 93-101. DOI=http://dx.doi.org/10.1145/ 505008.505026
16. C. E. Hughes, C. B. Stapleton, D. E. Hughes and E. M. Smith. Mixed reality in education, entertainment, and training. In *IEEE Computer Graphics and Applications,* vol. 25, no. 6, pp. 24-30, Nov.-Dec. 2005. DOI: 10.1109/MCG.2005.139
17. J. Pair, U. Neumann, D. Piepol and B. Swartout, "FlatWorld: combining Hollywood set-design techniques with VR," in IEEE Computer Graphics and Applications, vol. 23, no. 1, pp. 12-15, Jan/Feb 2003. DOI: 10.1109/MCG.2003.1159607
18. Luv Kohli, Eric Burns, Dorian Miller, and Henry Fuchs. 2005. Combining passive haptics with redirected walking. In *Proceedings of the 2005 international conference on Augmented tele-existence* (ICAT '05). ACM, New York, NY, USA, 253-254. DOI=http://dx.doi.org/ 10.1145/1152399.1152451
19. Adalberto L. Simeone, Eduardo Velloso, and Hans Gellersen. 2015. Substitutional Reality: Using the Physical Environment to Design Virtual Reality Experiences. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 3307-3316. DOI: https://doi.org/10.1145/2702123.2702389.
20. Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D. Wilson. 2017. Sparse Haptic Proxy: Touch Feedback in Virtual Environments Using a General Passive Prop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17). ACM, New York, NY, USA, 3718-3728. DOI: https://doi.org/10.1145/3025453.3025753

21. Hinckley, K., Pausch, R., Goble, J.C., and Kassell, N.F. Passive Real-world Interface Props for Neurosurgical Visualization. *Proc. CHI '94*, 452–458.

22. HTC Corporation. 2017. HTC Vive. Retrieved September 7, 2017 from https://www.vive.com..

23. Ortega, M. and Coquillart, S. Prop-based haptic interaction with co-location and immersion: an automotive application. *Proc. HAVE'05*.

24. Boris Schling. 2011. *The Boost C++ Libraries*. XML Press.

25. Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. 2016. Haptic Retargeting: Dynamic Repurposing of Passive Haptics for Enhanced Virtual Reality Experiences. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16). ACM, New York, NY, USA, 1968-1979. DOI: https://doi.org/10.1145/2858036.2858226

26. Evan A. Suma, Zachary Lipps, Samantha Finkelstein, David M. Krum, and Mark Bolas. 2012. Impossible Spaces: Maximizing Natural Walking in Virtual Environments with Self-Overlapping Architecture. *IEEE Transactions on Visualization and Computer Graphics* 18, 4 (April 2012), 555-564. DOI=http://dx.doi.org/10.1109/TVCG.2012.47

27. Laurent Aguerreche, Thierry Duval, and Anatole Lécuyer. 2010. Reconfigurable tangible devices for 3D virtual object manipulation by single or multiple users. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology* (VRST '10). ACM, New York, NY, USA, 227-230. DOI=http://dx.doi.org/10.1145/1889863.1889913

28. R. W. Lindeman, J. L. Sibert and J. K. Hahn, "Hand-held windows: towards effective 2D interaction in immersive virtual environments," *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*, Houston, TX, 1999, pp. 205-212. doi: 10.1109/VR.1999.756952

29. Andre Zenner and Antonio Kruger. 2017. Shifty: A Weight-Shifting Dynamic Passive Haptic Proxy to Enhance Object Perception in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 23, 4 (April 2017), 1285-1294. DOI: https://doi.org/10.1109/TVCG.2017.2656978

30. John C. McClelland, Robert J. Teather, and Audrey Girouard. 2017. Haptobend: shape-changing passive haptic feedback in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction* (SUI '17). ACM, New York, NY, USA, 82-90. DOI: https://doi.org/10.1145/3131277.3132179

31. Razzaque, S., Kohn, Z., & Whitton, M. C. (2001, September). Redirected walking. In Proceedings of EUROGRAPHICS (Vol. 9, pp. 105-106).

32. Usoh, M., Arthur, K., Whitton, M. C., Bastos, R., Steed, A., Slater, M., & Brooks Jr, F. P. (1999, July). Walking> walking-in-place> flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 359-364.