HUMAN ACTUATION

LUNG-PAN CHENG

Dissertation zur Erlangung des Grades eines Doktors der Naturwissenschaften
- Dr. rer. nat -

Human Computer Interaction
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany

June 2018

ABSTRACT

Ever since the conception of the virtual reality headset in 1968, many researchers have argued that the next step in virtual reality is to allow users to not only see and hear, but also feel virtual worlds. One approach is to use mechanical equipment to provide haptic feedback, e.g., robotic arms, exoskeletons and motion platforms. However, the size and the weight of such mechanical equipment tends to be proportional to its target's size and weight, i.e., providing human-scale haptic feedback requires human-scale equipment, often restricting them to arcades and lab environments.

The key idea behind this dissertation is to bypass mechanical equipment by instead leveraging human muscle power. We thus create software systems that orchestrate humans in doing such mechanical labor—this is what we call *human actuation*. A potential benefit of such systems is that humans are more generic, flexible, and versatile than machines. This brings a wide range of haptic feedback to modern virtual reality systems.

We start with a proof-of-concept system—*Haptic Turk*, focusing on delivering motion experiences just like a motion platform. All Haptic Turk setups consist of a user who is supported by one or more *human actuators*. The user enjoys an interactive motion simulation such as a hang glider experience, but the motion is generated by those human actuators who manually lift, tilt, and push the user's limbs or torso. To get the timing and force right, timed motion instructions in a format familiar from rhythm games are generated by the system.

Next, we extend the concept of human actuation from 3-DoF to 6-DoF virtual reality where users have the freedom to walk around. *TurkDeck* tackles this problem by orchestrating a group of human actuators to reconfigure a set of passive props on the fly while the user is progressing in the virtual environment. TurkDeck schedules human actuators by their distances from the user, and instructs them to reconfigure the props to the right place on the right time using laser projection and voice output.

Our studies in Haptic Turk and TurkDeck show that human actuators enjoy the experience but not as much as users. To eliminate the need of dedicated human actuators, *Mutual Turk* makes everyone a user by exchanging mechanical actuation between two or more users. Mutual Turk's main functionality is that it orchestrates the users so as to actuate props at just the right moment and with just the right force to produce the correct feedback in each other's experience.

Finally, we further eliminate the need of another user, making human actuation applicable to single-user experiences. *iTurk* makes the

user constantly reconfigure and animate otherwise passive props. 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. Our demo experience features one example each of iTurk's two main types of props, i.e., reconfigurable props (the foldable board from TurkDeck) and animated props (the pendulum).

We conclude this dissertation by summarizing the findings of our explorations and pointing out future directions. We discuss the development of human actuation compare to traditional machine actuation, the possibility of combining human and machine actuators and interaction models that involve more human actuators.

ZUSAMMENFASSUNG

Seit der Konzeption des Virtual-Reality-Headsets im Jahr 1968 argumentieren Forscher, der nächste Schritt in der virtuellen Realität ist nicht nur zu sehen und zu hören, sondern in virtuelle Welten auch fühlen zu können. Ein Ansatz solch haptisches Feedback zu geben ist die Verwendung mechanischer Ausrüstung, etwa Roboterarme, Exoskelette und Bewegungsplattformen. Jedoch sind die Größe und das Gewicht solcher Ausrüstung proportional zur Größe und Gewicht der Person, d. h. haptisches Feedback für einen Menschen erfordert Ausrüstung mit Größe und Gewicht eines Menschen. Dieses Ausmaß an Gerätschaften ist oft limitiert auf Arkaden oder Laborumgebungen.

Der Schlüsselgedanke dieser Dissertation besteht darin, mechanische Geräte zu umgehen und stattdessen menschliche Muskelkraft zu nutzen. Wir erstellen Softwaresystem, die Menschen bei mechanischen Arbeiten orchestrieren, um anderen Menschen haptisches Feedback zu geben. Dies nennen wir „Human Actuation“ – menschliche Aktuierung. Ein möglicher Vorteil solcher Systeme ist es, dass Menschen generischer, flexibler und vielseitiger sind als gängige mechanische Ausrüstung. Dies bringt eine neue Bandbreite von haptischen Feedbackmöglichkeiten in moderne Virtual-Reality-Systeme.

Wir beginnen mit einem Proof-of-Concept-System– Haptic Turk, mit Schwerpunkt auf die Bewegungserlebnisse, die eine solche menschliche Bewegungsplattform liefert. Alle Haptic Turk Konfigurationen bestehen aus einem Nutzer, sowie einem oder mehreren Menschen, die den Nutzer unterstützen, den Aktuatoren. Der Nutzer genießt eine interaktive Bewegungssimulation wie zum Beispiel die Simulation eines Hängegleiters, jedoch wird die Bewegung von Menschen erzeugt, die die Gliedmaßen des Benutzers manuell heben, kippen und drücken. Um das Timing einzuhalten, folgen Sie den Anweisungen des Systems. Ein aus Rhythmusspielen bekanntes Format wird dabei dynamisch von dem System erzeugt.

Als nächstes erweitern wir das Konzept von „Human Actuation“ um 3-DoF auf 6-DoF Virtual Reality. Das heißt, Nutzer haben nun die Freiheit in der virtuellen Welt umherzugehen. TurkDeck löst dieses Problem, indem es eine Gruppe menschlicher Aktuatoren orchestriert, die eine Reihe von Requisiten rekonfigurieren, die der Nutzer fühlen kann, während er sich in der virtuellen Umgebung fortbewegt. TurkDeck plant die Positionierung der Menschen und weist sie zur richtigen Zeit an, die Requisiten an den richtigen Ort zu stellen. TurkDeck erreicht dies mit Hilfe von Laserprojektion und einer Anweisung gebender synthetischen Stimme.

Unsere Studien zu Haptic Turk und TurkDeck zeigen, dass menschliche Aktuatoren ihre Erfahrung zwar genießen, jedoch in dem Ausmaß wie der Nutzer selbst. Um menschliche Aktuatoren mehr einzubeziehen macht Mutual Turk aus jedem Aktuator einen Nutzer, d.h. mehrere Nutzer geben sich gegenseitig haptisches Feedback. Die Hauptfunktion von Mutual Turk besteht darin, dass es seine Nutzer so orchestriert, dass sie die richtigen Requisiten im richtigen Moment und im richtigen Ausmaß betätigen, um so das richtige Feedback in der Erfahrung des Anderen zu erzeugen.

Schlussendlich eliminieren wir die Notwendigkeit anderer Nutzer gänzlich und ermöglichen Erfahrungen für Einzelnutzer. iTurk lässt seinen Nutzer passive Requisiten neu konfigurieren und animieren. Dadurch kann iTurk virtuelle Welten mit stetig wechselnden Möglichkeiten bereitstellen oder sogar haptische Effekte generieren, obwohl jede Bewegung im System vom Nutzer selbst ausgelöst wird. Unsere Demo-Applikation verfügt über je ein Beispiel der von iTurk ermöglichten zwei Haupttypen von Requisiten - rekonfigurierbare Requisiten (eine faltbare Tafel aus TurkDeck) und animierter Requisiten (ein Pendel).

Wir schließen die Dissertation mit Verweisen auf mögliche Forschungsrichtungen ab, die sich durch die präsentierten Systeme ergeben. Wir diskutieren „Human Actuation“ sowohl im Vergleich zu herkömmlichen mechanischen Geräten, aber auch in der Kombination, da sich mechanische Geräte und Menschen gegenseitig ergänzen können. Zudem erkunden wir mögliche Interaktionsmodelle, die sich durch das Einbeziehen von menschlichen Aktuatoren ergeben.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

[1] Lung-Pan Cheng, Li Chang, Sebastian Marwecki, and Patrick Baudisch. "iTurk: Turning Passive Haptics into Active Haptics by Making Users Reconfigure Props in Virtual Reality." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 2018, 89:1–89:10. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173663. URL: http://doi.acm.org/10.1145/3173574.3173663.

[2] Lung-Pan Cheng, Patrick Luehne, Pedro Lopes, Christoph Sterz, and Patrick Baudisch. "Haptic Turk: A Motion Platform Based on People." In: *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: ACM, 2014, pp. 3463–3472. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557101. URL: http://doi.acm.org/10.1145/2556288.2557101.

[3] Lung-Pan Cheng, Sebastian Marwecki, and Patrick Baudisch. "Mutual Human Actuation." In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17. Quebec City, QC, Canada: ACM, 2017, pp. 797–805. ISBN: 978-1-4503-4981-9. DOI: 10.1145/3126594.3126667. URL: http://doi.acm.org/10.1145/3126594.3126667.

[4] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D. Wilson. "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. Denver, Colorado, USA: ACM, 2017, pp. 3718–3728. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025753. URL: http://doi.acm.org/10.1145/3025453.3025753.

[5] Lung-Pan Cheng, Thijs Roumen, Hannes Rantzsch, Sven Koehler, Patrick Schmidt, Robert Kovacs, Johannes Jasper, Jonas Kemper, and Patrick Baudisch. "TurkDeck: Physical Virtual Reality Based on People." In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. UIST '15. Charlotte, NC, USA: ACM, 2015, pp. 417–426. ISBN: 978-1-4503-3779-3. DOI: 10.1145/2807442.2807463. URL: http://doi.acm.org/10.1145/2807442.2807463.

[6] Pedro Lopes, Sijing You, Lung-Pan Cheng, Sebastian Marwecki, and Patrick Baudisch. “Providing Haptics to Walls and 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. Denver, Colorado, USA: ACM, 2017, pp. 1471–1482. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025600. URL: http://doi.acm.org/10.1145/3025453.3025600.

[7] Sebastian Marwecki, Maximilian Brehm, Lukas Wagner, Lung-Pan Cheng, Florian ’Floyd’ Mueller, and Patrick Baudisch. “VirtualSpace - Overloading Physical Space with Multiple Virtual Reality Users.” In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI ’18. Montreal QC, Canada: ACM, 2018, 241:1–241:10. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173815. URL: http://doi.acm.org/10.1145/3173574.3173815.

[8] Dominik Schmidt, Rob Kovacs, Vikram Mehta, Udayan Umapathi, Sven Koehler, Lung-Pan Cheng, and Patrick Baudisch. “Level-Ups: Motorized Stilts That Simulate Stair Steps in Virtual Reality.” In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI ’15. Seoul, Republic of Korea: ACM, 2015, pp. 2157–2160. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702253. URL: http://doi.acm.org/10.1145/2702123.2702253.

ACKNOWLEDGMENTS

First and foremost, I offer my sincerest gratitude to my supervisor, Patrick Baudisch, who has supported me throughout the course of this dissertation. This work would not have been accomplished without countless opinions and insightful inspirations from you. I have learned so much from our discussions as you never hesitate to share your thoughts, to ask bold questions and to explain details. I am deeply thankful for your patience, knowledge and experience.

I am also grateful for the opportunity to work with a few people outside school. Thanks Eyal Ofek, Christian Holz, Andy Wilson, Mike Sinclair and Hrvoje Benko for their support during the time I was working with them. Thanks Eyal for his friendly mentorship and numerous ideas. I really enjoyed the daily brainstorming with you. Thanks Christian for his sharp critics, useful career tips and impromptu jokes. I had a lot of fun whenever I took you to a new Chinese restaurant. Thanks Andy for his great advising. It was really nice to chat with you and to learn subtle conversation. Thanks Mike for the many discussions and the memorable boat trip. Thanks Benko for his insightful feedback on the work that we have done together.

I owe a great debt to all my collaborators for all the projects we worked on together. Thank you Sebastian Marwecki for, in addition to translating the abstract into German, all the hard work and thoughts that you contributed to Mutual Turk. We would not have gotten the best demo award without you. Thank you Lily for spending a summer on iTurk with me and flying over to help the demo at the conference. It was my great honor to be your mentor. Thanks Pedro for inviting me working with him on interactive electrical muscle stimulation. Thanks Thijs Roumen, Robert Kovacs, Hannes Rantzsch, Sven Koehler, Patrick Schmidt, Johannes Jasper and Jonas Kemper for working on the monstrous project—TurkDeck. It was an amazing group effort over two years, which allowed me not only to apply my research on a larger scale, but also to learn so much from you. Thanks Patrick Luehne, and Christoph Sterz for helping me get on the track with Haptic Turk.

I would like to acknowledge Alexandra Ion, David Lindlbauer, Oliver Schneider, Tim Chen, Jack Lindsay and Edward Wang for their comments and the good times, as well as Bing-Yu Chen, Mike Y. Chen, Rong-Hao Liang, and Liwei Chan for many brainstormings and feedback. I would also like to thank Jiajun Lu, Haijun Xia, Artem Dementyev Ken Pfeuffer and Gierad Laput for hanging out with me during my internships.

Finally, I would like to thank my family and my girlfriend Ling-Hsuan Huang. Thanks my parents for supporting me from thousands miles away in Taiwan. Thanks my big brother for taking care of all family emergencies so I don't have to frequently travel back and force. Thanks Ling-Hsuan Huang for proofreading the German abstract, for inspiring debates about art and research, for teaching me music, and for all the fun trips that took me out from the ivory tower.

CONTENTS

LIST OF FIGURES

1

INTRODUCTION

Ever since the conception of the virtual reality headset in 1968 by Sutherland [64], research have been through several stages to enhance immersion, such as more photorealistic graphics [52], spatial sound [16], and full-body interactions [12]. Many researchers argue that the next step to immersion is to bring physical feedback in so that the user can not only see and hear but also feel the virtual world [63].

1.1 NEXT STEP TO IMMERSION: PHYSICAL FEEDBACK

In the past, researchers have pursued two different approaches. On the one hand, researchers use mechanical machinery, such as motion platforms [61] and exoskeletons [35] to apply forces to the user. These have been used to add realism to flight simulators and theme park rides. While these approaches have been very successful at providing forces, they are not well suited for recreating the experience of touching objects, such as grabbing a door handle or slamming against a wall.

Researchers therefore proposed using physical props, also known as passive haptics [22, 38]. Simple prop-based systems use a single hand-held prop [46]. The more elaborate systems support "real walking" [68] in a space where all walls were physical (with projection [41] or head-mounted displays [26]) allowing users to experience the full physicality of the room. While passive haptics are easier to set up, their need for static positions limits their capability to represent virtual environments with complex geometries, large dimensions, or dynamic objects. Although there are techniques to allow confronting users with the same prop repeatedly [33], the immutability of the props makes them hard to reuse.

Researchers thus combined these two approaches together: to reconfigure props using robots [44] or shape displays [30] to dynamically render the physical world. This approach compensates each of the disadvantages, allowing the simulation of a wide range of physical feedback including animate objects in the virtual worlds.

These were all good approaches to provide physical feedback in the era when people had to travel miles away to enjoy immersive experience. Back then, virtual reality was only achievable with with specialized hardware such as visualization workstations [56], tracking cameras [42] and projectors or head-mounted displays.

However, providing physical feedback in virtual reality has been left behind as many underlying technologies have been brought to commercial market: graphics cards [11], head-mounted displays and tracking systems [10]. Currently people can run virtual reality at their living rooms, but without physical feedback the illusion of presence is broken whenever they physically interact with the virtual world.

The reason why physical feedback has not been brought to a broader audience is that, unlike other technologies, it heavily involves mechanical engineering. The size and weight of mechanical parts remains the same because of physical constraints, making it prohibitively large, heavy and stationary when simulating large-scale physical feedback. A machine that can lift up a person is roughly the size of a person, limiting their use to arcades or labs.

1.2 CONTRIBUTIONS AND STRUCTURE

In this dissertation, we contributes a systematic exploration of a new concept — *human actuation* — to bring large-scale physical feedback up to speed with other modern virtual reality technologies that have been brought to the commercial market. Our key idea is to bypass mechanical equipment by instead leveraging human power. Unlike machines, humans are generic, flexible, versatile and massively. To demonstrate the benefits and explore the limitations of human actuation, we consider humans as mechanical parts (or more specifically, *human actuators*) and implement software systems that orchestrate them to provide physical feedback in virtual reality. This dissertation is structured as follows:

RELATED WORK (Chapter 2)

We give an overview of related research that provides physical feedback in virtual reality.

PROOF OF CONCEPT—HAPTIC TURK (Chapter 3)

We start with a proof-of-concept system—*Haptic Turk*, focusing on delivering motion experiences just like a motion platform. All Haptic Turk setups consist of a user who is supported by one or more human actuators. The user enjoys an interactive motion simulation such as a hang glider experience, but the motion is generated by human actuators who manually lift, tilt, and push the user's limbs or torso. To get the timing and force right, timed motion instructions in a format familiar from rhythm games are generated by the system.

MORE DEGREES OF FREEDOM—TURKDECK (Chapter 4)

Next, we extend the concept of human actuation from 3-DoF to 6-DoF virtual reality where users have the freedom to walk around. TurkDeck tackled this problem by orchestrating a group of human actuators to reconfigure a set of passive props on the fly while the user is progressing in the virtual environment. TurkDeck schedules human actuators based on their distances from the user, and instructs them to reconfigure the props to the right place at the right time using laser projection and voice output.

MAKING EVERYONE A USER—MUTUAL TURK (Chapter 5)

Our studies in Haptic Turk and TurkDeck show that human actuators enjoy the experience but not as much as users. To eliminate the need of dedicated human actuators, Mutual Turk makes everyone a user by exchanging mechanical actuation between two or more users. Mutual Turk's main functionality is that it orchestrates the users so as to actuate props at just the right moment and with just the right force to produce the correct feedback in each other's experience.

ME, MYSELF, ITURK (Chapter 6)

We further eliminate the need of another user, bringing human actuation to single-user experience. iTurk makes the user constantly reconfigure and animate otherwise passive props. 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. Our demo experience features one example each of iTurk's two main types of props, i.e., reconfigurable props (the foldable board from TurkDeck) and animated props (the pendulum).

CONCLUSIONS AND NEXT STEPS (Chapter 7)

We conclude this dissertation by summarizing the findings of our explorations and pointing out future directions. We discuss the development of human actuation compare to traditional machine actuation, the possibility of combining human and machine actuators and interaction models that involve more human actuators.

2

RELATED WORK

Human actuation is an approach to larger-scale physical feedback. Research that related to physical or haptic feedback can be categorized into two:

PASSIVE HAPTICS Passive haptics are physical objects that are used to match mainly the shape but also the texture and the weight of virtual objects, giving users the exact tactile sensation whenever touching or manipulating them. These physical objects do not move by their own and thus are passive. People usually custom-make these physical objects and oftentimes call them *props* as in a play.

ACTIVE HAPTICS Active haptics are devices that are controlled by a computer to programmatically simulate haptic feedback, mainly forces and motions. These devices can dynamically interact with users and thus are active.

2.1 PASSIVE HAPTICS

Hinckley et al. [22] pioneered the concept of using passive props to as haptic feedback. They used a doll's head and a rectangle plate to allow users to explore three-dimensional neurosurgical visualizations. Similarly, users of the Personal Interaction Panel [65] use a pen and a pad to help manipulate objects in see-through augmented reality. Sheng, Balakrishnan, and Singh [55] used a sponge-like prop to create a range of interaction techniques. Lindeman, Sibert, and Hahn [38] used a window registered with a tracked, physical surface, to provide support for precise manipulation of interface widgets displayed in the virtual environment, and called it passive-haptic feedback.

Previous work shows passive haptics can enhance the sense of presence. In a study by Hoffman [23], 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 [26], 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 have been used in real walking environments. Low et al. [41], for example, use Styrofoam walls onto which they project augmented reality experiences. Similarly, mixed reality for military operations in urban terrain [25] uses passive haptics to add a haptic sense to otherwise virtual objects, terrain, and walls. FlatWorld [47] integrates large props into a physical world; between experiences these props can be rearranged to match the next virtual world.

2.1.1 *Repurposing Props*

While passive haptics are easier to set up, their need for static positions limits their capability to represent virtual environments with complex geometries, large dimensions, or dynamic objects. Researchers have thus examined reversing this approach and fitting the virtual scene to the geometry of existing physical objects. In Substitutional Reality [57], researchers conducted a study on how the mismatch between physical and virtual props can break believability. Annexing reality [21] analyzes the environment and opportunistically assigns objects as passive proxies, but a given physical environment may not support all the appliation needs, and the virtual geometry has to deform to fit existing geometries.

Kohli et al. [33] use redirected walking [49] to allow users to encounter a stationary prop at different virtual locations. Redirected walking is achieved by injecting additional translations and rotations to the user's head movements, causing users to walk on a physical path that is different from the perceived virtual path. By having the user walk back to the same physical object when moving from one virtual target to the next, the object provides haptic feedback for multiple virtual targets. Steinicke et al. [60] investigate the thresholds for imperceptible redirected walking through their user studies, indicating the limits of redirected walking.

Kohli et al. extend the concept of redirection to touching [32], manipulating the user's virtual hand-eye coordination. This results in discrepancies between a person's real and virtual hand motions, so that they reach the real and virtual objects simultaneously while they are actually not in the same location. Haptic Retargeting [2] uses the same concept but smoothes the transitions of the user's hand movement, making the user to manipulate any one virtual cube on the table while there is only a single physical cube prop. Spillmann, Tuchschmid, and Harders [59] proposed adaptive space warping to warp different organ geometries onto one physical mock-up for surgery training. Sparse Haptic Proxy [9] used a hemisphere prop combining with haptic retargeting [2] to give touch feedback whenever users touch the virtual object.

2.1.2 *Reconfigurable Props*

Reconfigurable props were a new class of passive haptics which has more flexible use. Aguerreche, Duval, and Lécuyer [1] proposed manually reconfigurable props for better manipulation of 3D objects in virtual environments. Self-actuated reconfigurable props such as Haptobend [43] and Shifty [73] used machine actuators to change the geometry and the weight distribution.

2.2 ACTIVE HAPTICS

Simulating haptic feedback using active machinery has been researched for decades since Goertz and Thompson [17] in 1954 proposed a robotic arm for teleoperation that transmitted the force feedback on the end effector to users. Project GROPE [7] in 1967 then used the robotic arm to provide force feedback when users were operating the arm to touch and grasp virtual objects in virtual reality. Since then a wide variety of force feedback devices have been used to enhance the realism of virtual reality interactions including virtual object manipulation and motion simulation [13].

2.2.1 *Simulating Motions*

In the class of devices that support whole-body experiences we find motion simulators; they shake, lift or tilt users sitting or standing on them. Motion simulators are used in driving and flight simulation for both training and entertainment purposes. Most of them are based on a Stewart platform [61], which offers six degrees of freedom driven, e.g., by six hydraulic cylinders as actuators.

A range of locomotion devices increases realism by simulating terrain geometry. The ground surface simulator [45], for example, is a treadmill equipped with individually height-adjustable elements of up to 6 cm that simulate bumpy terrain and virtual slopes. The torso force feedback system [24] pulls users walking on a treadmill using an active mechanical link, simulating a slope.

Gait Master [28] provides users with additional freedom in that it allows users to “walk”. The system, however, captures the user again on every step using motion platforms that position themselves where the user is expected to step next. CirculaFloor [29] builds on the same concept, but uses four robot units that place themselves under the user’s steps.

2.2.2 *Exoskeletons*

Exoskeletons are wearable machines not only for amplifying users’ motion but also providing force feedback. There are different kinds of exoskeletons, such as [20, 35] that use different actuators (pneumatic, hydraulic and strings) to help users perform six or more degrees of freedom tasks either in the real or the virtual world. They can provide large forces, but are also heavy and large because of their actuators and mechanical parts.

2.2.3 *Minimization*

To bring active haptics to a broader audience, researchers proposed devices with smaller form factors.

HapSeat [14] achieves motion simulation with lower cost and a more compact form factor by actuating the user's head and hands. The fact that users perceive motion mainly using their visual, auditory, vestibular, and kinesthetic systems [6, 19] allows this project to limit actuation to arm- and headrests.

The GLAD-IN-ART project [5] develops a glove-exoskeleton for manipulativing virtual objects. FlexTorque [67] offers an exoskeleton in a portable form factor; it provides force feedback to the arm of a user playing shooting games. Level-Ups [54] attach on the user's feet to simulate terrain in real-walking interfaces .

Researchers also showed that vibrotactile feedback may generate the illusion of self-motion [51]. Tactile Brush [27] uses this haptic illusion and renders vibrotactile strokes on the user's back using a grid of actuators in the chair. AIREAL [58] remotely sends vibrotactile feedback by shooting out air vortex to the user's body.

Lopes et al. proposed the use of electrical muscle stimulation as a mean to simulate the impact (Impacto [39]) and the resistance of objects in real-walking virtual reality (EMS Walls [40]). GyroTab [3] produces torque in a mobile form factor based on a gyroscopic effect. Yano et al. [71] proposed a handheld haptic device that uses a laser range finder to control a linear servomotor that touches users fingertips so that users can touch the object remotely. Han et al. [18] proposed a handheld haptic device that integrates a linear servomotor and a solenoid–magnet pair located under a membrane to simulate huge and subtle kinesthetic feedback when the finger touches the membrane.

2.2.4 *Combining With Passive Haptics*

Robotic Graphics [44] used a robotic arm to move a board to where users were about to touch, which physically "rendered" the entire geometry of the virtual world. Iwata et al. used shape displays [30] to render geometries and provide touch feedback.

3

HUMAN ACTUATION IN HAPTIC TURK

Haptic Turk is a motion platform based on people. The name is inspired by the 18th century chess automaton "The Turk" [36] that was powered by a human chess master.

The specific configuration shown in Figure 1 involves one user located in the center. The user wearing a head-mounted display (an Oculus Rift DK1 [69]) is enjoying an immersive experience, here a first-person simulation of flying a hang glider. The user is fully immersed with motion that is administered by *human actuators* who manually lift, tilt, and push the user around.

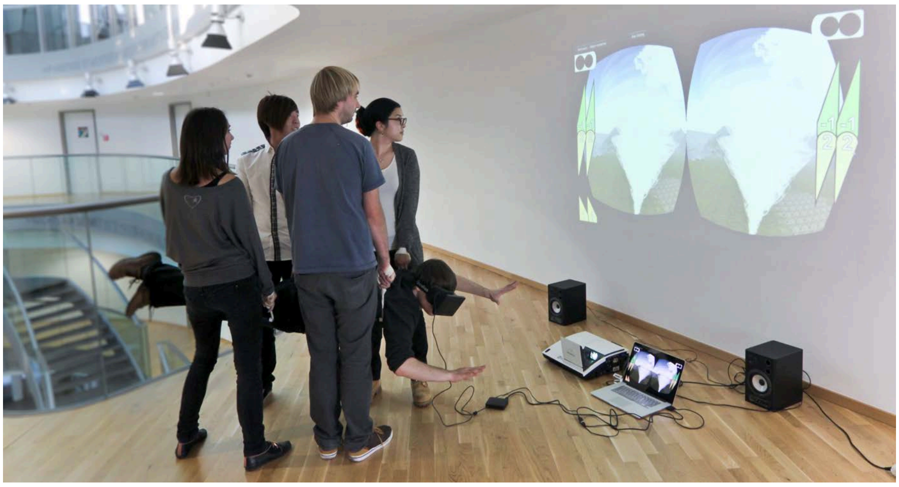

Figure 1: Four *human actuators* follow the instructions on the screen to create motion for the user's virtual hang glider experience.

To get the timing and force right, all human actuators receive timed motion instructions in a format familiar from rhythm games [37]. In the set-up shown in Figure 1, human actuators receive these motion instructions on the projection screen. During the experience, human actuators execute the motion instructions displayed on the screen by moving the part of the user's body assigned to them. At the moment shown in Figure 1, for example, the user is about to encounter a tornado that will shoot him up to the sky, while the four human actuators slowly lower the user then abruptly raise the user to match the drastic upward motion.

Haptic Turk generates these motion instructions so as to feed haptic feedback into the user's experience. As human actuators perform their motion instructions, they therefore contribute to the user's experience, making it richer and more immersive.

3.1 WALKTHROUGH: TEAM FLIGHT

We now illustrate the Haptic Turk platform at the example of the specific configuration from Figure 1, a hang glider experience we call *team flight*. The actual experience takes three minutes and contains 14 motion event groups (25 individual actuator movements overall). We present selected scenes that allow us to illustrate the design elements.

As illustrated by Figure 2, the experience starts out in calm weather and the user's hang glider is in a neutral, horizontal position. Accordingly, the instruction display is blank. This instructs all human actuators stand upright, relaxed, shoulders dropped—a position that human actuators can sustain with minimum effort. We design our experiences to bring human actuators back to this position frequently to avoid fatigue. Note that the display show the timelines of all human actuators so as to help them see "the big picture" and to synchronize their actions.

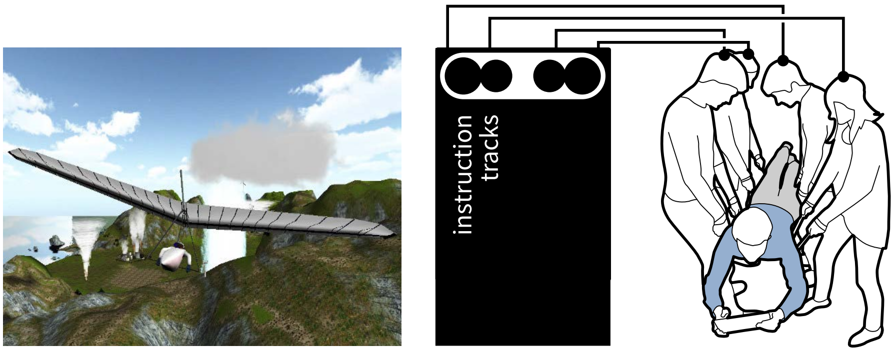

Figure 2: The *team flight* hang glider experience has started: the instruction display is blank, human actuators are relaxed.

up/down bars The user is approaching a big fan located on the right, and the hang glider is about to get caught by the fan's draft. Haptic Turk reflects this by rendering up/down motion instructions. The bars enter the display from below and travel up the screen as the hang glider is approaching the fan (Figure 3). As the bars reach the actuators' bullseyes at the top of the screen, the respective actuators execute the instructions. Motion instructions take about seven seconds to reach the bullseye. This is essential as it allows human actuators to get ready so as to perform their motion on time, i.e., in sync with the user's experience.

The shown up/down bars in the two right columns reach the bullseye exactly at the moment the user enters the draft of the fan. The up/down bars are labeled "1"; the two human actuators on the right therefore lift the user to position "1", which is waist height. This causes the user to be rotated or "rolled" to the left, as the user's hang glider is pushed sideways by the fan.

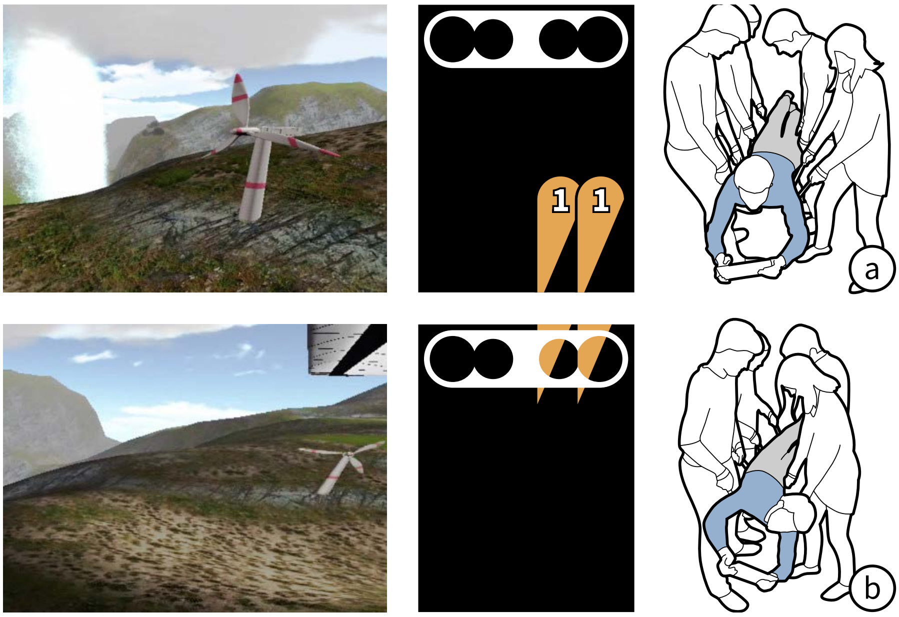

Figure 3: (a) The user is about to get caught by the draft of a fan, "lift to +1" motion instructions appear, and (b) are executed as they reach the bullseyes.

The two up/down bars have round heads, which demands actuators to move up abruptly, right at the moment the round head fills the round bullseye. Actuators continue to hold the user at +1 height, then return smoothly to the relaxed position, as demanded by the bars' diagonal tails.

Up/down bars are Haptic Turk's most versatile design element, and they are used to render the vast majority of motion events. As shown in Figure 4 , up/down bars come with values from -2 to +2, with +2 'chest level', +1 'waist level', blank 'relaxed', -1 'knee level', and -2 'foot level'. Redundant color-coding makes the display more glanceable (sun orange means up; grass green means down). The main idea behind up/down bars is that they show actuators exactly what to do, yet, with experience, allow them to see the big picture.

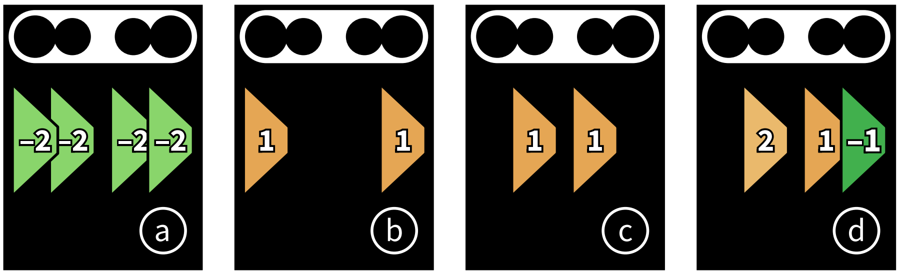

Figure 4: Up/down bar representations for (a) drop, (b) face down tilt, (c) face up tilt, and (d) up-left tilt-roll.

BUMPY RIDE In Figure 5 the user enters turbulences, here created by a group of fans pointed at the user from different directions. Haptic Turk responds by creating 'shake' instructions for all actuators, which each actuator executes as an unsynchronized vertical shake.

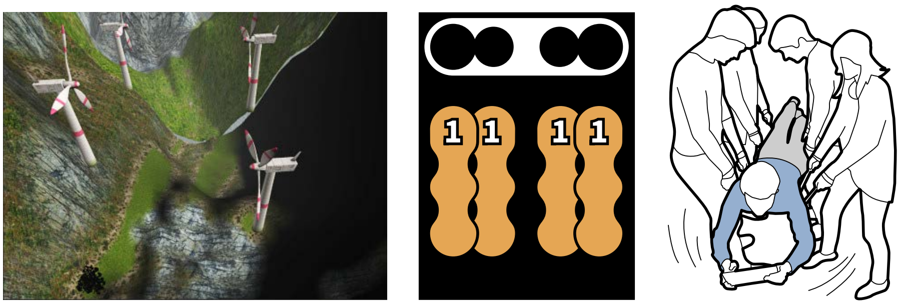

Figure 5: The fans cause an unsynchronized shake.

LARGE MOVEMENTS AND ANTICIPATION Figure 6 shows how Haptic Turk performs a large motion event. Here the user enters a tornado, which causes the user's hang glider to shoot up into the air. Haptic Turk renders this by instructing all four actuators to abruptly lift the user to '+2' the chest level. To make the user's experience even more intense, Haptic Turk instructs actuators to first lower the user slightly in anticipation and then to perform the vertical '+2' move.

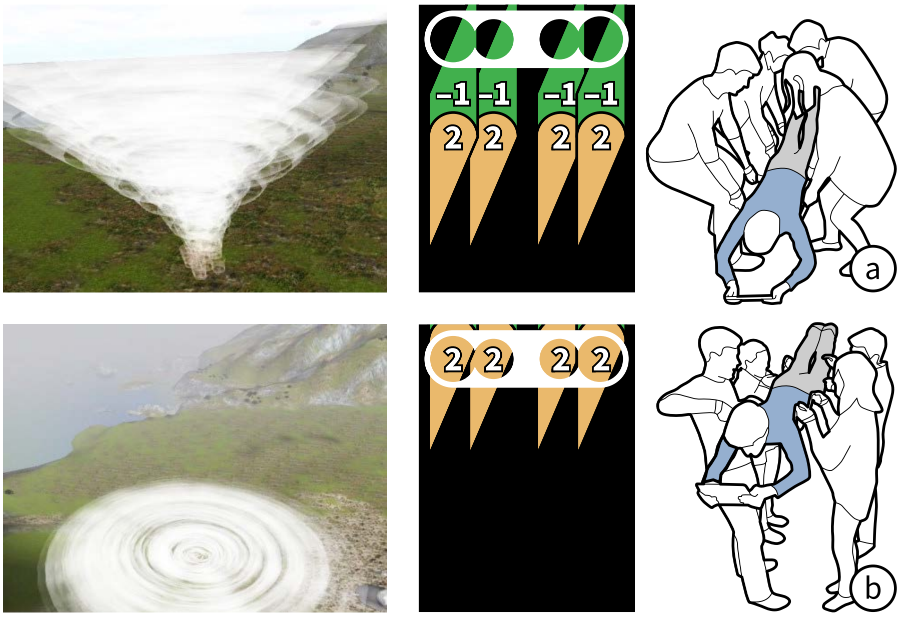

Figure 6: Right before the user enters the tornado, Haptic Turk anticipates the move by lowering the user.

COLLISIONS In Figure 7, the user's hang glider collides with an object, here a blimp approaching from the left. Haptic Turk renders the event as a pair of "bump" instructions on the left. As the bump

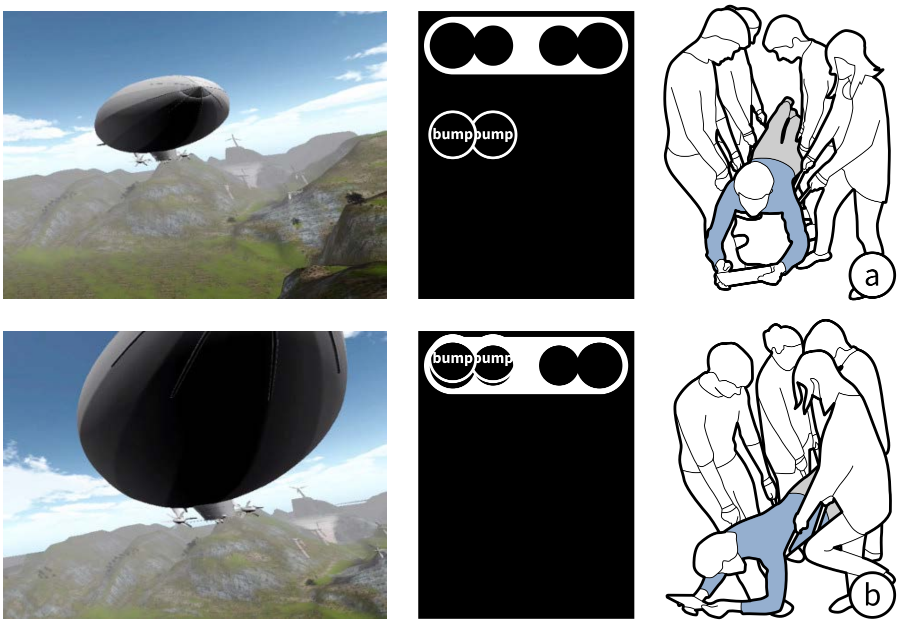

Figure 7: As the user collides with an object, here a blimp, human actuators bump the user using their thighs.

events reach the bulls¬eyes, the left two human actuators bump the user using their thighs.

HORIZONTAL MOTION In Figure 8, the user is entering the field of a very powerful fan that propels the hang glider quickly out over the desert. Haptic Turk emphasizes the onset of this movement with a horizontal move event, which human actuators execute by taking a step into the specified direction. After all, human actuators are flexible enough to walk around during the experience.

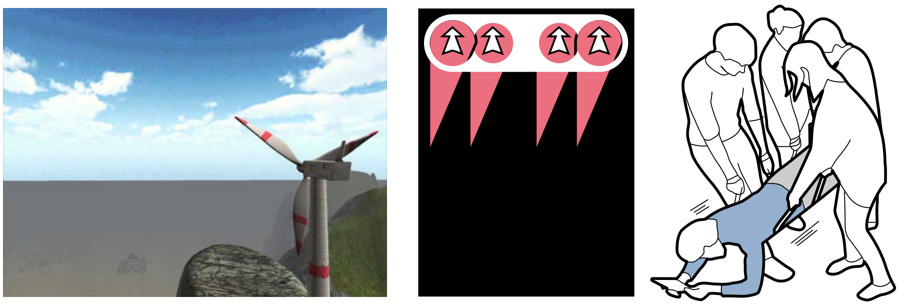

Figure 8: This fan produces a strong wind that shoots the user off horizontally, rendered as horizontal motion.

SPECIAL EFFECTS The concept of Haptic Turk is broader than just motion. Figure 9 shows two effects that we have explored, i.e., water and heat. They are administered by an additional "special effects" human actuator.

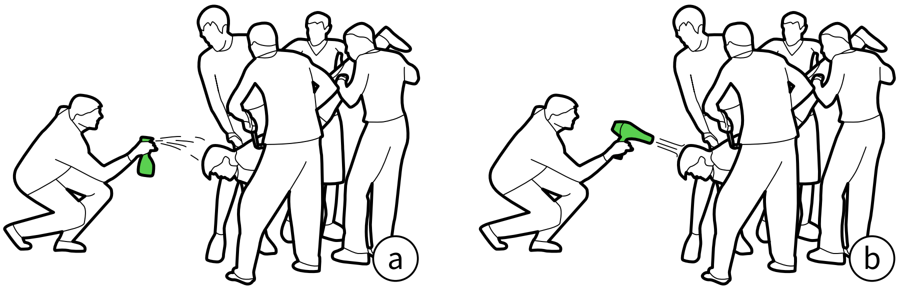

Figure 9: This human actuator (a) sprays water at the user as the user enters a waterfall and (b) uses a hair dryer to emulate a hot breeze, as the user crosses over into the desert terrain.

LANDING Finally, the user has reached the destination: a temple in the middle of the desert. As the user approaches the ground, Haptic Turk generates landing instructions, as shown in Figure 10.

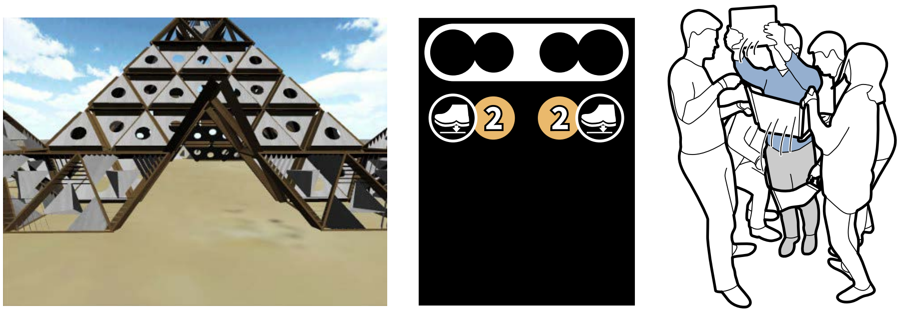

Figure 10: To land the user, the front human actuators whip up the user to +2, while the back two set down the user's feet.

These were all simple scenes, taken from an experience designed to be calm and serene, as one would expect a hang-glider experience to be. Figure 11 shows a more action-packed experience we created—note how the instruction display now handles multiple actions at once.

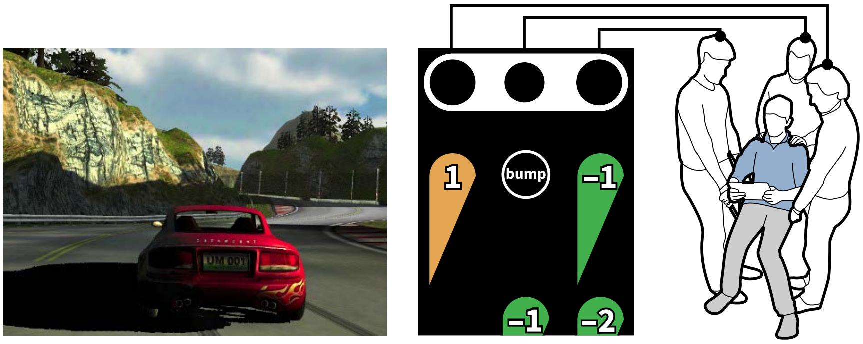

Figure 11: More action-packed experience, such as our car racing game contains faster sequences of motion instruction. (This experience is administered by three human actuators, thus three timelines)

3.2 GETTING THE TIMING RIGHT

The main technical challenge behind human actuation is to get the timing right, i.e., to make sure that human actuators provide force exactly at the moment required by the user's experience. This is a challenge because human actuators are inherently subject to human response time and thus delay. The mean response time for humans to react to a simple visual stimulus is 220ms [34]. This duration is substantially longer than the 50ms that humans tolerate in delayed haptic feedback [31].

For pose changes, we found this delay to not cause too much of an issue as long as we simulate a vehicle that moves smoothly. Hang gliders, for example, change their poses only gradually in response to their pilots shifting their weight around. Haptic Turk simply displays a preview of the expected goal pose as soon as the user starts to steer; this gives human actuators ample time.

The true challenge comes from instantaneous events, such as the forces resulting from a user colliding with an object. Haptic Turk addresses this challenge with two approaches.

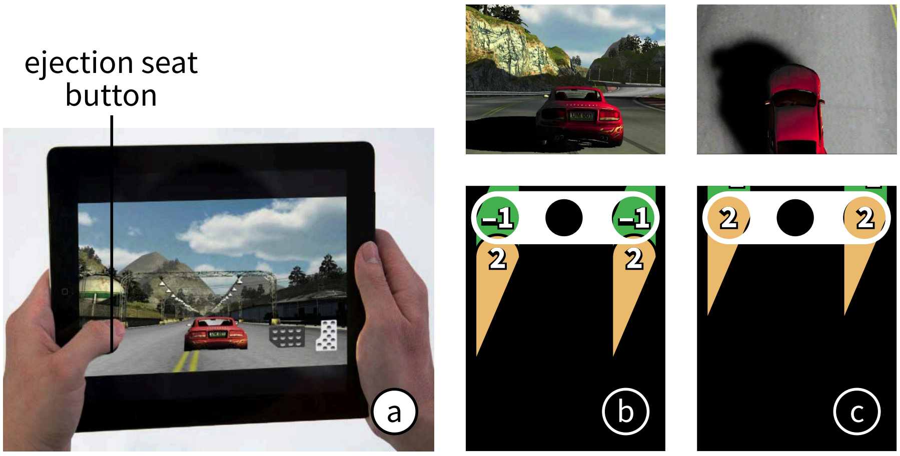

Figure 12: Delaying an event using a countdown: (a) In response to the user pressing the 'eject' button, (b) Haptic Turk renders up/down bars. (c) When the countdown runs down, the human actuators eject the user, well prepared.

1. COUNTDOWN EVENTS Haptic Turk cannot speed human actuators up, but it can slow the event down—by adding a countdown. We use this approach for all user actions where precision matters, either because they result in a high force or because that are performed by multiple human actuators in synchrony. The ejection seat button in the car of our racing game, for example, uses a three-second countdown before it fires. The countdown not only adds drama to the experience, but, more importantly, allows Haptic Turk to give human actuators advance warning. This is essential, as it allows human actuators to throw the user up in the air in synchrony.

2. ANTICIPATE COLLISIONS Only certain events can plausibly be fitted with a countdown. For other events, such as those resulting from physical collisions, Haptic Turk creates timely responses through "anticipation".

Rather than waiting for the collision to happen, Haptic Turk continuously checks for the possibility of an upcoming collision. Whenever a collision is likely, Haptic Turk displays a possible bump event, allowing human actuators to get ready. Haptic Turk probes the space that the user is likely to reach in the next seconds using probe lines (Figure 13, inspired by imaginary reality games [4]). Each probe line is as long as the user can travel in the seven seconds that the instruction display projects into the future.

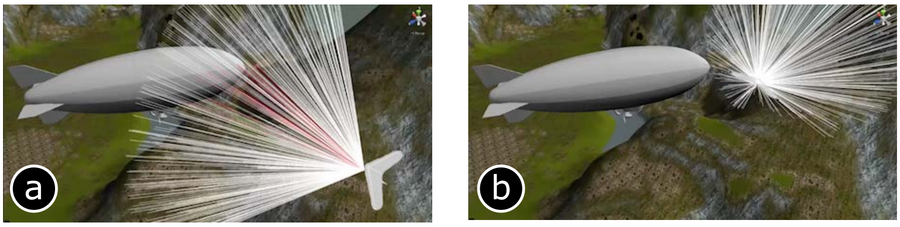

Figure 13: (a) Haptic Turk's debug view: the hang glider on the bottom right continuously sends out probe lines (1000 on a MacBook Pro). Here some of them detect a blimp (shown in red). (b) As the user turns right to avoid the blimp, no more probe lines are reaching the blimp.

At every frame, Haptic Turk counts how many probe lines predict a collision. If a collision seems likely (>30%), Haptic Turk injects a motion event into the instruction display (Figure 14). It renders the motion instruction's opacity so as to reflect the probability of the event, so 70% opacity represents a 70% probability. Haptic Turk then continuously updates the actuator display.

As the user navigates, the expected collision time tend to change. The motion instructions may thus speed up if the user steers into the obstacle or they may slow down if the user turns to hit the obstacle more tangentially. At the same time, the expected collision probability and thus the opacity of the motion instructions will vary. If the probability ever drops below a threshold (<15%), Haptic Turk removes the bump instruction—a "false alarm".

To enable probe lines, all the in-game objects have their own colliders and corresponding motion instruction. The tornado in the hang gliding game, for example, has a cylindrical collider. Probing it triggers the -1/+2 up/down bars shown earlier.

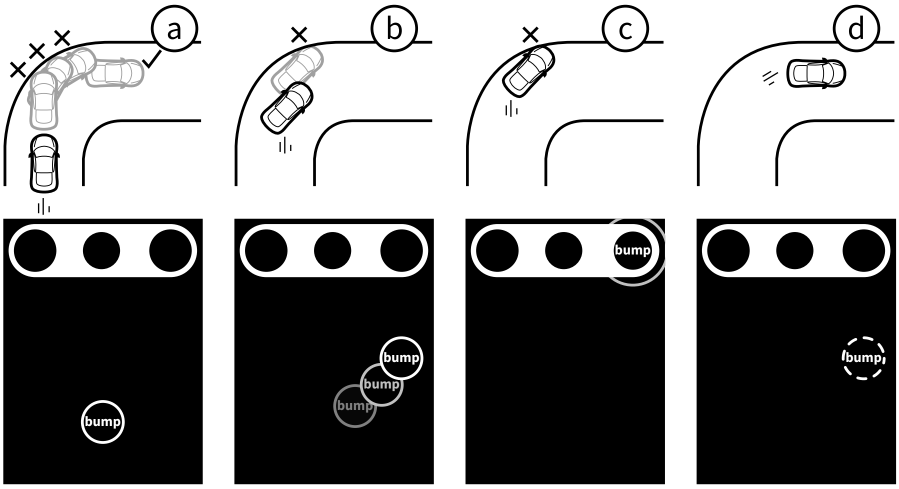

Figure 14: (a) Haptic turk anticipates a collision with the curb and crates a bump event. (b) As the user turns, the bump event changes to reflect the new expected orientation of the bump. (c) If the car stays the course, the collision takes place. Otherwise, (d) the bump event dissolves.

3.3 ONE PLATFORM, MANY CONFIGURATIONS

The Haptic Turk software is a general-purpose platform for creating motion experiences based on people. In the following, we show other hardware configurations we have explored.

3.3.1 *Display/Sensing Configurations*

Figure 15a shows a mobile set-up. It reduces hardware requirements by running on two iPads—one for the user and a single shared iPad for the human actuators.

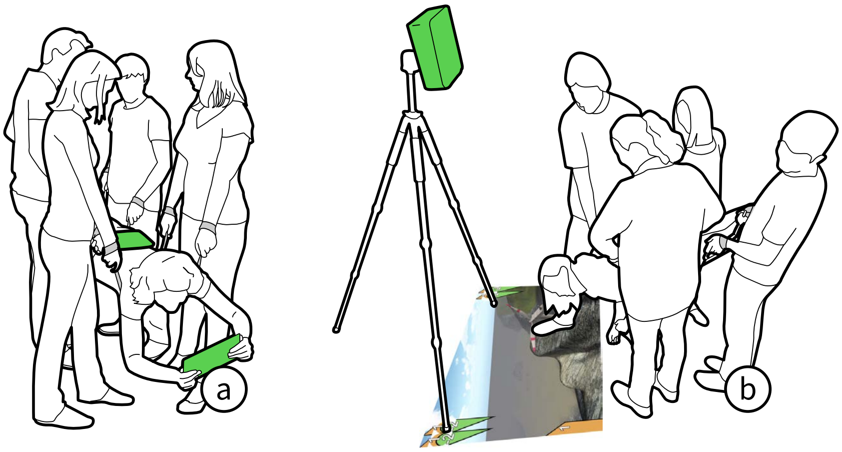

Figure 15: (a) Haptic Turk running on mobile devices (b) The walk-up installation is designed to minimize per-user start-up time.

Figure 15b shows the installation that we have shown in Figure 1. This is designed for use at tradeshows, museum exhibits, and art shows. Walk-up installations do not require participants to bring anything and we designed them to minimize per-user set-up time.

We implemented the Haptic Turk platform and the experiences (hang glider and racing car) in C# on Unity [66]. We deploy to notebook computers and mobile devices. We used the Oculus Rift DK1 [69] for our walk-up virtual reality set-up immersive version.

In the mobile version, Haptic Turk connects devices using Wi-Fi. Informal testing showed a mean latency of 3 ms, which is faster than the earlier discussed 50 ms delays that humans tolerate in a haptic response. To run an experience, one person starts the Haptic Turk app as "host", which makes this person the "user" for the first round (Figure 16). (b) All other users join the session as actuators. (c) Haptic Turk indicates each actuator where to stand with respect to the user. The actuators Velcro-strap their devices to the indicated part of the user's body.

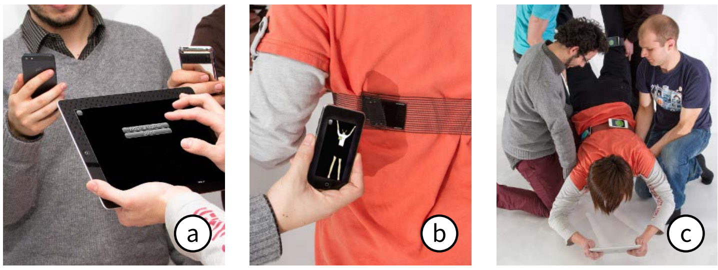

Figure 16: Setting up a haptic turk session via Wi-Fi: (a) the user starts a session, (b) joining human actuator are assigned a position on the user's body, (c) play.

Finally, we can record Haptic Turk experiences and share them as "Haptic Turk movies". We developed a web application that allows users to annotate existed videos on video-sharing websites such as Youtube [72] with our timed motion instructions (as in Figure 17). People can then render the experience on any installation they have access to.

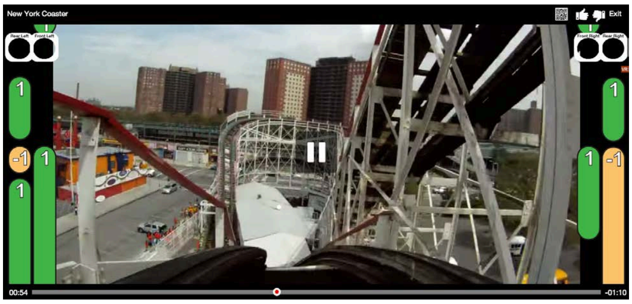

Figure 17: Users annotate and share their motion experiences on our web application.

3.3.2 *Mechanical Configurations*

All designs shown so far use four human actuators to keep the user suspended. We initially had human actuators hold the user directly, but then added the slingshots shown in Figure 18 to reduce human actuator fatigue, to increase user safety, and to soften potential proxemics issues. The shown design is made from the seat cushions of foldable chairs (Folding stool Stockholm II by Lectus), curtain bars, and linen ribbon.

Figure 18: We made these custom slingshots to reduce human actuator fatigue, increase user safety, and soften proxemics.

Figure 19 shows some of the alternative mechanical configurations we have explored. The shown designs (a, b, c) allow three experiences that require an upright user pose, such as car racing, (b, c, d) provide additional physical support to the user, thereby reducing human actuator fatigue. They allow fewer human actuators with a wider range of physical abilities.

Figure 19: Haptic Turk configurations: (a) racing game based on two human actuators, (b) chair and (c) swing allow for single-actuator use, and (d) in swimming pool.

3.3.3 *Instruction Interfaces*

The instruction interface presented earlier in 3.1 was the result of a series of iterations. Figure 20 shows two earlier designs for context. Our initial "level and arrows" design (Figure 20a) caused split attention, but most of all it caused ergonomic issues: Since this only indicated the desired user tilt, but no absolute height, human actuators tended to spend most of their time in non-neutral postures, resulting in substantial fatigue.

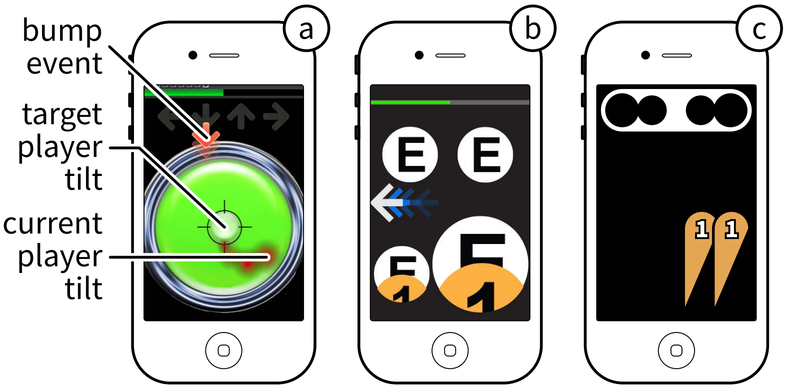

Figure 20: Earlier versions of the instruction interface: (a) Initial version based on the metaphor of a two-dimensional level. Only for bumping events did we use the time-line metaphor. (b) The second version helped us get human actuators into the neutral position, thus improved ergonomics. (c) The current interface.

We consequently switched to a model that instructed human actuators using a notion of absolute height (Figure 20b). The 2x2 layout also showed the instructions of up to four human actuators, allowing human actuators to better synchronize.

The interface, however, was limited in that it provided human actuators with too little preview of upcoming events, which is an essential requirement. We addressed this by lining up the four timelines at the top of the display and unifying all instructions to fit the timeline format—which brought us to the current version shown in Figure 20c. It performed very well throughout studies and deployment.

3.4 USER STUDY

To validate our approach, we conducted a user study. Our main objective was to verify that the haptic experience does indeed produce an enjoyable experience—for users and for human actuators.

We recruited 14 participants (4 females) from our university. Their age ranged from 20 to 28 years (*M*= 23.6, *SD*= 3.2), BMI from 19-25 (*M*=22.3, *SD*=2.3). We recruited participants in groups of two to four, i.e., participants were familiar with some of their human actuators, but never all of them.

TASK In teams of six (one user, four human actuators, one special effects actuator) participants went thorugh the team flight hang glider experience, parts of which we presented in the walkthrough section.

We ran the team flight experience on the walk-up virtual reality setup of Haptic Turk already shown in Figure 1, i.e., users experienced the world through a head mounted display (an Oculus Rift DK1 [69]). Human actuators saw their instruction on the computer screen in front of the user.

To assure that all participants enjoyed the same experience, we set the hang glider experience to autopilot. This limited user's interaction abilities to looking around, but assured that all participants encountered the same events. It also allowed us to always complete the experience in a fixed amount of time (three minutes), thereby creating a controlled experience also for the human actuators.

PROCEDURE We brought in participants in groups of 2, 3, or 4 and filled in experimenters so as to reach the 5 people required to play (plus we provided one experimenter for the water special effects). We then played one round of the team flight experience for each participant, so that each participant had the opportunity to be user exactly once.

With each new group, we provided two minutes of training during which we explained the handling of the slingshots, the instruction display and how high to lift the user for the four types of up/down bars, which track each human actuator was expected to follow, and how to wear the head mounted display.

Participants played once as a user and 1-3 times as a human actuator, according to their group size. Within these constraints, the order was counterbalanced. After all participants of a group were done playing, all participants filled in questionnaires about their experience. We then released the group of participants and brought in the next group. Running a group of 2-4 participants took 10 to 15 minutes.

RESULTS Figure 21 summarizes our results. As users, participants rated their experience on average as 6.1 (*SD*=0.7) on a 7-point Likert scale (1=unpleasant, 7=enjoyable)—so clearly as enjoyable. Overall, users preferred the large motion events. Five users stated that they liked the intense motion resulting from lifting, shaking and bumping. Another user stated that he particularly enjoyed whole-body movements, such as being lifted or being swung forward—more so than being tilted and rolled. Along the same lines, three participants stated that they did not enjoy the extensive landing period in which they were tilted down. One participant described tilting as uncomfortable.

Accordingly, when asked about the most impressive moment of their user experience, 11 participants picked the intense -1/+2 boost caused by the tornado. One user expressed that "the changes in altitude were amazing and immersive". Another user explained that he enjoyed the moment when he bumped into the blimp.

Participants rated the experience as human actuator as less enjoyable than as users, yet still on the "enjoyable" side (*M*=4.4, *SD*=1.2).

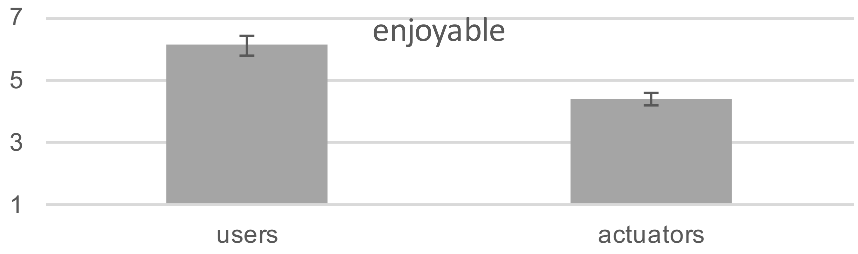

Figure 21: Users rated the experience as 6.1, so clearly enjoyable. Also as human actuators, participants enjoyed the experience. Error bars are +/-1 standard error of the mean.

The human actuators' experience was strongly driven by their perception of users' experience. Human actuators felt that their performance contributed to user's experience (*M*=5.4, *SD*=1.3). Our observations match this. Five human actuators said that they enjoyed seeing their users scream and giggle. One human actuator said "it's fun to play this with your friends and see their reactions as they fly." And one simply stated "it's fun to watch." One participant would have enjoyed an even better view of what the user is experiencing. While we thought of the special effects role as being less exciting, one human actuator said he would have also liked to take on that role.

The most likely reason for the lower score on enjoyment was fatigue. Seven human actuators mentioned fatigue. One human actuator mentioned that fatigue kicks in after two rounds. Another human actuator said "The person we moved was too heavy for me and I was smaller than the other users so my arms end at a lower height." Two human actuators mentioned that lifting the user to level +2 repeatedly had caused fatigue.

To learn more about proxemics, we asked participants who they would play team flight with. They indicated that they would play with friends (14/14) and family (10/14), but only 1/14 felt it was appropriate to play with the public. Given that this study had forced participants to play with a group of mostly strangers, this suggests that subjective satisfaction may improve further if experienced in a closer circle of friends and family. One participant explicitly said that she would enjoy playing Haptic Turk with her kids.

The human-human nature of Haptic Turk polarized participants. While 7/14 participants responded that they would have preferred an experience administered by a mechanical motion platform, 5/14 stated that they preferred being actuated by humans (Figure 22).

This suggests that these participants found an enjoyable quality in the human element. This is interesting, especially given that the human actuators were assigned to them for the purpose of the experiment and not the friends or family that participants would have liked to play with.

A Haptic Turk experience may incentivize human actuators to perform their best by measuring and scoring how well they match their motion instructions in terms of timing, position, and force—using the inertial measurement unit (IMU) in their mobile device. However, we obtain a better user experience by scoring human actuators as a group, as this encourages human actuators to synchronize during cooperative moves. But finally, several participants in our user study stated that they simply enjoyed supporting the user, suggesting that the act of providing an experience to the user is all the necessary incentive.

In summary, our simple study provides some initial validation for the Haptic Turk concept. Most importantly, participants very much enjoyed the user experience. The human actuator experience, while still enjoyable, could be improved by longer breaks and by giving the choice of who to play with.

Figure 22: Some participants preferred Haptic Turk, other would have preferred a mechanical motion platform.

3.5 EXPERIMENTAL DEPLOYMENT

Encouraged by the results of the study, we decided to try an experimental deployment of Haptic Turk and the team flight experience at an art festival in the Nevada desert (Burning Man [48]). Our goal was to learn more about the social dynamics of Haptic Turk—outside the lab.

We again opted for the walk-up VR set-up previously shown in Figure 1. As shown in Figure 23, we adapted the set-up to the hostile desert environment by placing the projector in a box equipped with air filters, etc. The entire installation (minus the generator, which we acquired on site) was transported, set up, and run by one of the authors, emphasizing Haptic Turk's potential to deliver motion experiences anytime anywhere.

We ran the installation only at night. While users wore the head mounted display, the projector projected a copy of the virtual scene with overlaid instruction display onto the desert ground, which served as actuator display (Figure 24). As before, we ran the team flight ex-

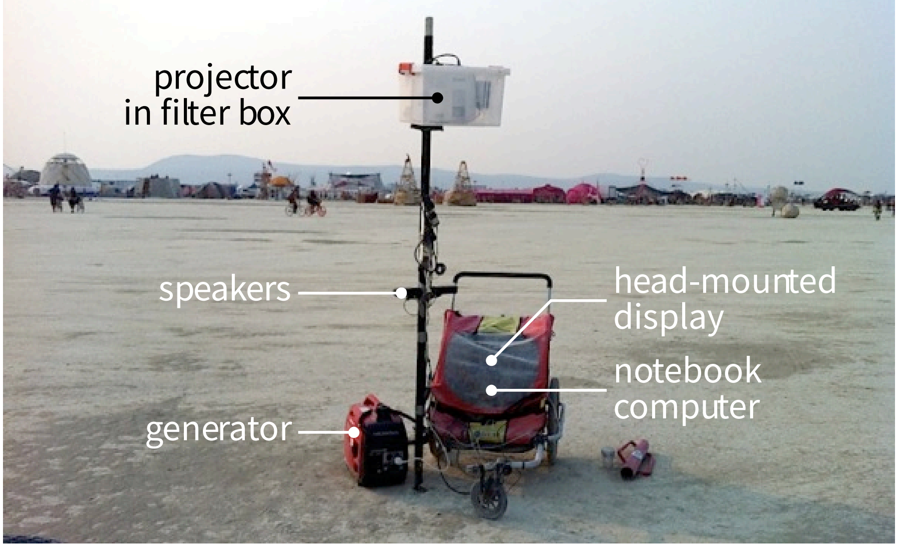

Figure 23: Our walk-up installation with head mounted display at the Burning Man art festival in the Nevada desert. A plastic box with fans protected the projector from desert dust.

perience in auto¬pilot and without additional incentives for human actuators.

Figure 24: Our installation projected the virtual scene overlaid with the instruction display onto the desert ground. This user weighed 200 pounds.

On three nights, we ran about 100 attendees. User weights ranged from an estimated 100 pounds to a self-declared 200 pounds. While the venue did not afford running questionnaires, we observed attendees and videotaped four of the runs for further analysis.

Unlike a lab study, our set-up had to begin by attracting its own audience. Every run unfolded as follows. An attendee or a small group of two to three attendees would walk by and inspect the installation. There was no particular attract mode, but the projection showed the game running.

Whenever an attendee was interested in playing, they would convince their friends to be the human actuators for them. Groups, however, were small—never large enough to play. We then encouraged attendees to recruit strangers as additional human actuators. This typically took them a minute or two (“can you help us play a video game?”) and was simplified by the friendly atmosphere of the event.

While most attendees recruited to be able to play, several attendees “gifted” the experience to a friend—oftentimes their boy/girlfriend—i.e., they recruited human actuators, but then stepped back and let their friend take on the role as user while they actuated.

Given the constant competition for visual attention at the festival, it was essential for us to get started as quickly as possible. To get started in about a minute, we assigned human actuator positions, helped human actuators into the slingshots, and taught them about how high to raise lower at the individual up/down bars. We then filled in one position ourselves, which allowed us to start right away and instead coach the other human actuators as the experience was unfolding.

As during the study, the large motion events were favorites for users and led to audible expressions of joy. Upon “landing”, several users were visible taken by the immersion of the experience and typically required a few seconds to find their way back to reality.

Among human actuators, the most popular scenes were again the large motion events, but also the rocky ride sequence in which human actuators had the opportunity to shake up the user a bit. One user reported motion sickness.

None of the human actuators mentioned fatigue this time. The reason could be that, unlike our study, most attendees actuated only once, very few of them twice. This suggests that the fatigue threshold for this particular four-person setup is around three to five minutes for this particular young athletic audience.

3.6 DISCUSSION

The main contribution of Haptic Turk is a new mechanism for creating a motion platform—based on humans. We present user interfaces and a system of motion instructions, two display and four mechanical configurations, interactive experiences, and two approaches to address the timing issue.

The main benefit of our approach is ubiquitous availability. (1) *Instantaneous*: Since the underlying units—people are incredibly flexible, users may obtain new experiences simply by downloading them from a network “into the human actuators”. This means that the human actuators that formed a hang glider a second ago may now serve as a car or battle robot. This way, Haptic Turk reduces the sharing of motion experiences to the sharing of data files. (2) *Available*: Haptic Turk runs on equipment that is orders of magnitude cheaper and more space-efficient than the technical equipment it emulates. In particular, the mobile versions of Haptic Turk could potentially reach millions of users by leveraging the existing install base of mobile devices. (3) *Everywhere*: We have deployed Haptic Turk not only in the lab and the student cafeteria but also at a desert festival, i.e., a hostile environment that would make it hard to deploy an actual motion platform. A single person was able to bring the equipment on the plane.

Limitations of our approach include that it requires multiple people, is tiring, and that human actuators cannot rival a computerized motion platform in terms of responsiveness and reliability. In particular, a real time unpredictable event is difficult to handle in our current system. For example, if users controlling a racing game swerve and hit a wall, this hit is difficult to predict. This makes the event difficult to haptically render by our system, as it does not leave enough time to properly prepare human actuators for their action. Haptic Turk runs in to similar issues if multiple events become possible at a given moment.

In addition, there are all the risks that come with motion equipment in the first place, such as motion sickness and the risk of injury. The approach may also raise proxemics considerations, the extent of which should be expected to vary across cultures. We address some of these issues with the design of slingshots that reduce actuator fatigue, increase user safety, and soften proxemics issues, but these are certainly only a partial solution.

In exchange to the proxemics issues, Haptic Turk delivers not only force feedback, but also a human-to-human experience that lets people interact in a new way. While we initially expected that competing to win the rhythm game would be the main incentive for actuators, the physical activity itself and, in particular, the social nature of the setup turned out to be the main driving force that made people participate in studies and experimental deployment.

We have presented Haptic Turk, the first proof-of-concept system of human actuation. Haptic Turk brings motion experiences to orders of magnitude cheaper, more space efficient, and faster-to-deploy than the mechanical equipment it is inspired by. Through Haptic Turk, we have demonstrated that human actuation works in 3DoF virtual reality where the user enjoys motion experiences in place. Our next step is to extend the concept of human actuation further to 6DoF virtual reality where the user has freedom to walk and touch around.

4

EXTENDING HUMAN ACTUATION TO REAL-WALKING: TURKDECK

While Haptic Turk provides immersive motion experiences, the user can only turn their head around but have no freedom to move and explore the surrounding. To achieve more immersive experiences, we bring human actuation to 6DoF virtual reality where the user can not only see and hear but also walk and touch around, which is TurkDeck.

Figure 25a shows a user immersed in a virtual reality experience in TurkDeck. The user can navigate freely, and can walk, sit, or lie, or turn and tilt their head. A head-mounted display (Oculus Rift DK1 [69]) continuously provides the user with a first-person stereoscopic view into the virtual world. A stereo headset provides spatial audio and prevents users from hearing what happens in the physical space around.

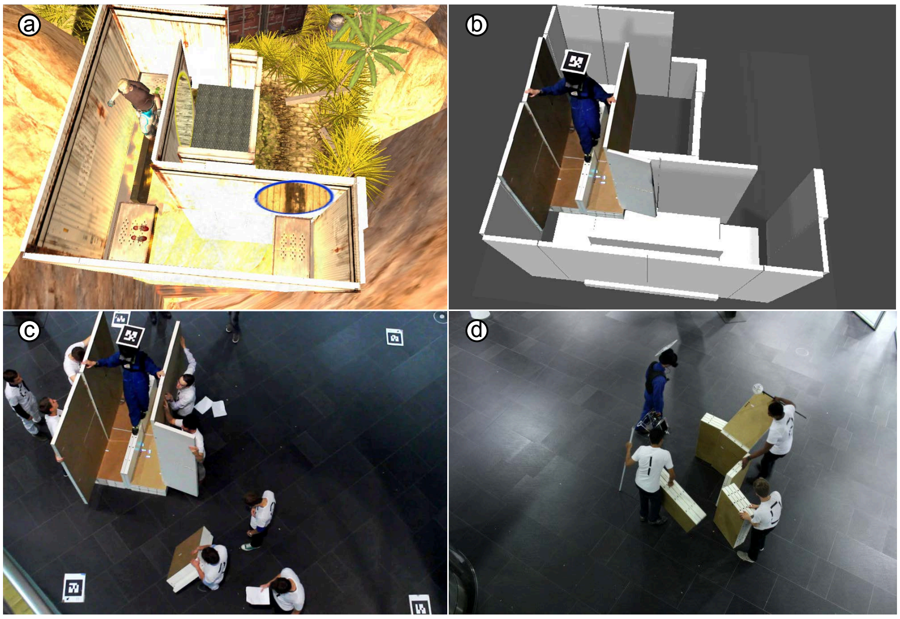

Figure 25: TurkDeck is a prop-based virtual reality system that let's users not only (a) see and hear a virtual world, but also (b) feel it. Conceptually, the user is in a fully populated physical world. (c) In reality, however, TurkDeck's physical room is almost empty. "Human actuators" present and operate props only when and where the user can actually reach them. (d) By reusing generic props, TurkDeck minimizes the required props to what human actuators can carry, still allows producing virtual worlds of arbitrary size.

The key element of TurkDeck, however, is that it provides the user not only with a visual and auditory experience, but also a physical/haptic experience. TurkDeck achieves this using a prop-based approach, i.e., whenever users touch an object in the virtual world, they also touch an object in the physical world. On a conceptual level, the user is thus immersed in the physical world as shown in Figure 25b, which illustrates how traditional prop-based systems have tackled the challenge (e.g., [26]).

Figure 25c, in contrast, shows the physical world actually created by TurkDeck. TurkDeck recreates the physical world only when and where the user can actually reach the elements. It creates this partial physical world on the fly with the help of human actuators. In the shown scene there are eight of them. Managed by TurkDeck, they position and operate props where required and just in time. TurkDeck's props are designed to be generic; human actuators can thus turn a prop that just served as a balancing beam into a wall, a table, a chair, etc. as necessary.

Figure 25d: The main benefit of our approach of constantly rearranging a small set of generic props is that it allows TurkDeck to create arbitrarily large, animated, and responsive virtual worlds in finite space and from limited physical resources. This is unlike previous prop-based systems that require physical space and physical props proportional to the amount of virtual space.

The central element of TurkDeck is the software system that manages human actuators to make them position and operate these props where required and just in time. TurkDeck does so by providing human actuators with timed instructions of when and where to place its custom props and how to actuate them.

4.1 REARRANGING PROPS BY HUMAN ACTUATORS

The following figures illustrate how TurkDeck uses human actuators to rearrange props—the main mechanism behind TurkDeck. As an example, we use the first few seconds of our demo experience called the *Lighthouse*.

The user enters the world on a thin ledge—apparently in the process of rock climbing. Through a head mounted display, the user experiences the world in first person. In the interest of visual clarity, however, we will show all remaining images in third person.

Figure 26 shows how TurkDeck allows the user to feel the vertical surface he is pressed against and the void under his heels using props operated by five human actuators. Two human actuators have laid down their boards to form the ledge the user stands on and four human actuators position their boards to simulate the segment of the cliff the user is holding on to.

Figure 26: (a) The user comes to, standing on a ledge, which is (b) simulated by five human actuators using six foldable boards.

In Figure 27, in order to locate a place to go, the user is inching towards the left, still pressed against the cliff. As shown, human actuators create two new segments of the cliff on the fly. At the same time, the wall segments on the right are now out of the user's reach; TurkDeck thus makes the respective human actuators tear them down.

Figure 27: As the user is inching towards the left, human actuators set up new wall segments there, tear down on the right.

4.1.1 *The Prop System*

The objective behind TurkDeck's prop design is to get by with the smallest possible number of generic props—as that makes the system small and potentially even portable.

Each human actuator carries only two main props, i.e., a *foldable board* and a *stick*. With the help of the human actuators, TurkDeck constantly reuses these props in different locations, orientations, and combinations, turning these few elements into a complete physical world.

We have created two sets of the foldable boards. Inspired by work in prop-based spatial augmented reality [41]), we made one version from Styrofoam; the resulting four-panel boards weigh one killogram. To offer users an even better experience, we also created the 12 kg high-fidelity version shown in Figure 28. These boards consist of the surfaces of four coffee tables, which we connected to one another using hinges.

Figure 28: The board folds in four. Joints and magnet connectors help human actuators reconfigure the boards. Finger holes enable actuators to lift boards easily.

The high-fidelity boards are very sturdy and their hardness and surface conveys the sensation of walls and steps convincingly, while the use of finger holes keeps the weight manageable for the human actuators.

Each board features a four-segment folding mechanism and magnetic connectors along all relevant edges. Hinges and magnets along the inside edges allows the boards to assume a wide range of shapes, some of which are shown in Figure 29.

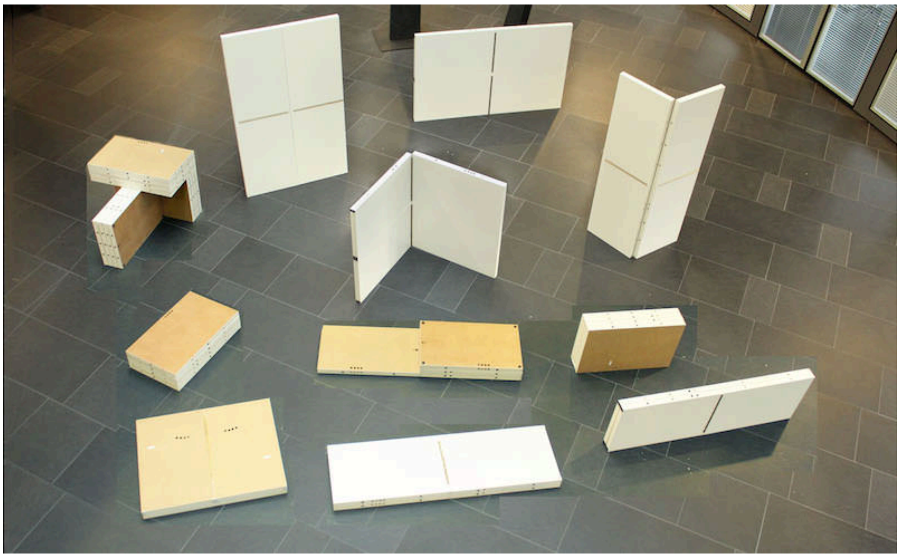

Figure 29: The foldable boards can be arranged into many different shapes.

Magnetic connectors along the outside edges allow human actuators to combine the boards into larger surfaces, such as the multi-segment walls in Figure 27.

The same mechanism allows boards to collapse into the 90 cm x 55 cm x 20 cm package shown in Figure 30. A built-in handle allows human actuators to carry their board and stick like a suitcase. As each human actuator has only one board and stick, the TurkDeck props can travel with the team.

Figure 30: The foldable board and the stick pack into a 90x55x20cm package that human actuators can carry like a suitcase.

Some mechanisms in our demo experience also use the stick shown in Figure 31. It is made from a metal pipe and it bears a male and a female mount at its two ends, allowing it to attach to other sticks. (b) This allows combining sticks into larger structures, such as the zip line mechanism shown in Figure 39. (c) Optional experience-specific attachments allow sticks to perform a wide range of functions; a Styrofoam knob, for example, turns a stick into a lever.

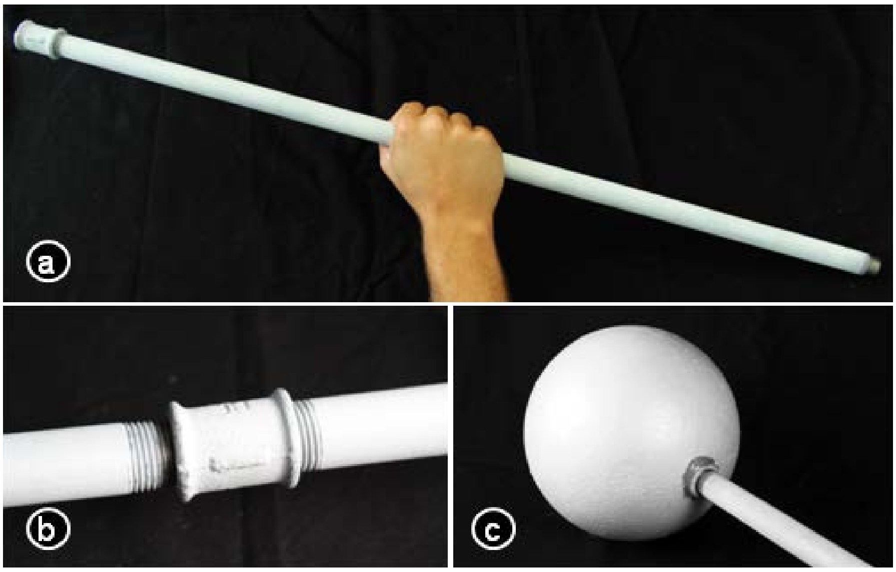

Figure 31: (a) Sticks bear mounts at their two ends. (b) This allows sticks to connect (c) to experience-specific attachments.

4.1.2 *The Display System*

As illustrated by Figure 32, a projector mounted eight meters above projects instructions for the human actuators onto the interaction area. We use a laser projector for this purpose (SmartLine RGB 1800), i.e., a vector display consisting of a laser, mirror, and galvanometer scanners, as used in laser shows. We opted for this type of display device because it delivers sufficient brightness to work in daylight, independently of the size of interaction area. We measured the power per square inch that reaches human actuators as the result of our specific display configuration and found safe levels. We encourage readers replicating our work to conduct similar measurements before using such a projection system. In accordance with our display system, we designed the visual language of the human actuator instructions to consist of vectors.

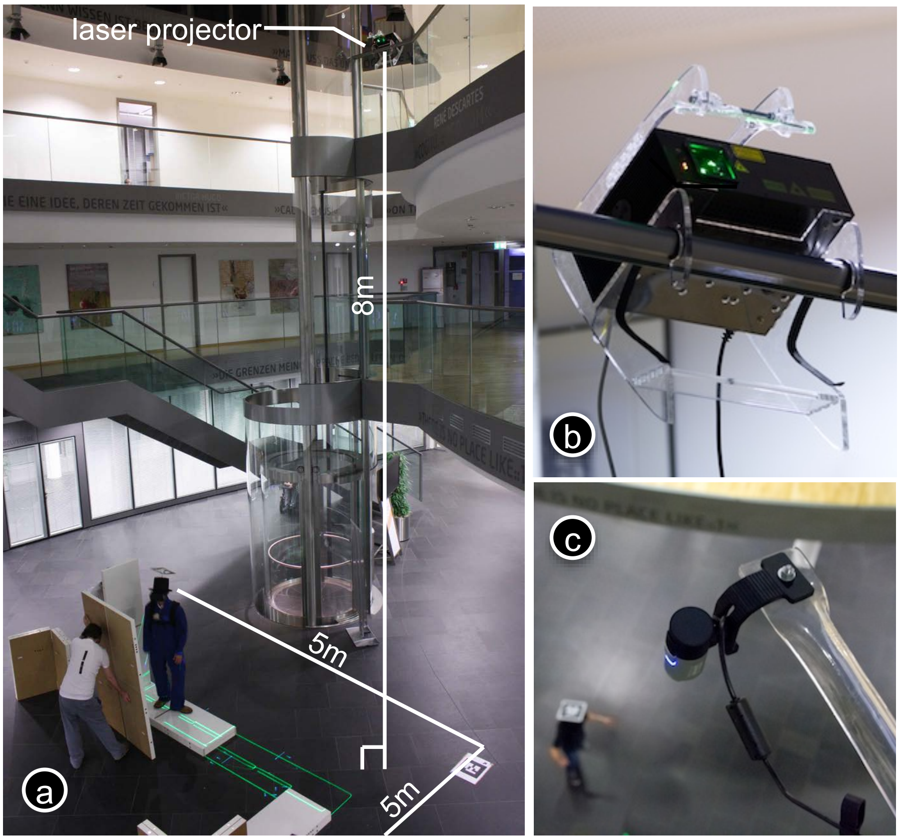

Figure 32: TurkDeck projects instructions human actuators onto the ground using a 200 mW laser projector.

4.1.3 *The TurkDeck Instruction System*

Human actuators perform their actions under the management of TurkDeck. As shown in Figure 33, TurkDeck talks to human actuators by projecting visual instructions on the ground. The use of a public projection system gives all human actuators shared understanding of the system status (unlike, for example, individual head mounted displays).

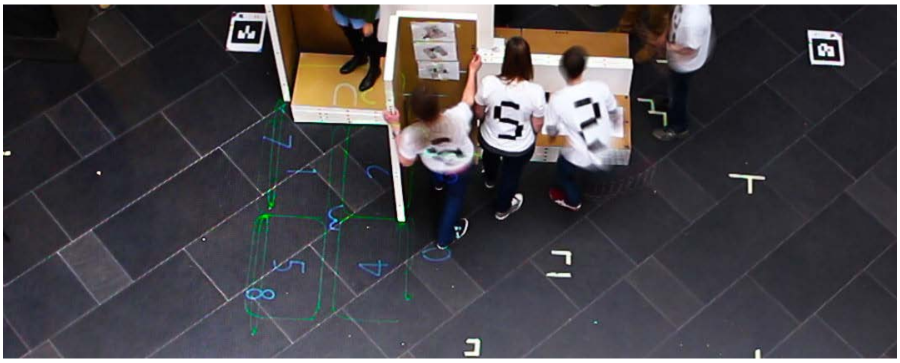

Figure 33: The projected human actuator interface consists of green board outlines and blue human actuator IDs.

Each instruction is a pair consisting of (1) an outline representing the requested prop, shown in the desired position and orientation and (2) the ID of the human actuator intended to perform the job, shown in the position and orientation the in which the human actuator is supposed to perform the task. The respective human actuator responds by standing on the ID, picking the respective prop from his/her personal prop repertoire, and transforming it to the shown shape. As illustrated by Figure 34, TurkDeck depicts each board as their rectangular projection and all sticks as lines, for a simple, yet effective visual language.

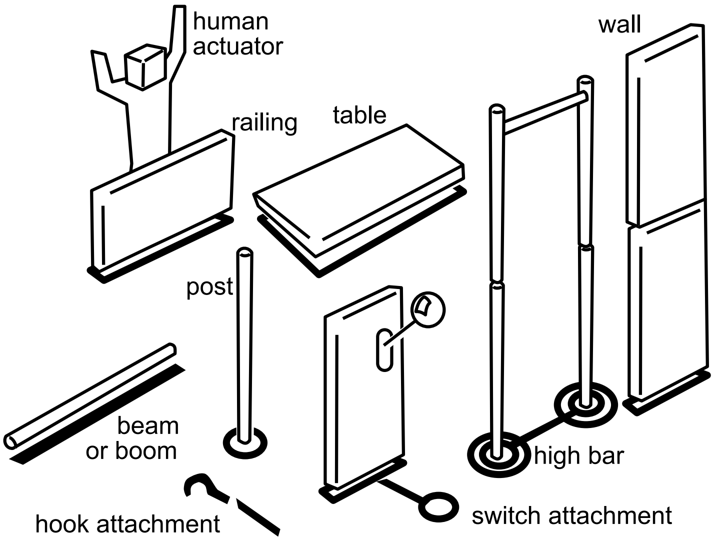

Figure 34: TurkDeck renders props as their outline.

In order to deal with occlusion and to maximize speed, TurkDeck starts projecting instructions a few seconds before they are required.

Furthermore, TurkDeck uses redundant cues when directing human actuators. In addition to the projected instructions, (a) a small number of visual landmarks on the ground (masking tape, see Figure 33) serve as visual reference whenever projection is occluded, (b) overview maps of the experience give human actuators a preview of their upcoming tasks, and (c) auditory alerts instruct actuators to move to the next position. The audio instruction "3B", for example, instructs human actuator "3" to move on to the location labeled "B".

4.2 WALKTHROUGH SELECTED MECHANISMS

We now demonstrate the resulting functionality of TurkDeck by showing selected mechanisms from the *Lighthouse* demo experience. The actual experience takes about 10 minutes and requires 65 physical props of 12 different types to be placed.

INTERACTIVE MECHANISMS Following the ledge from Figure 27 leads to a little alcove where the user finds a switch to open a door (Figure 35).

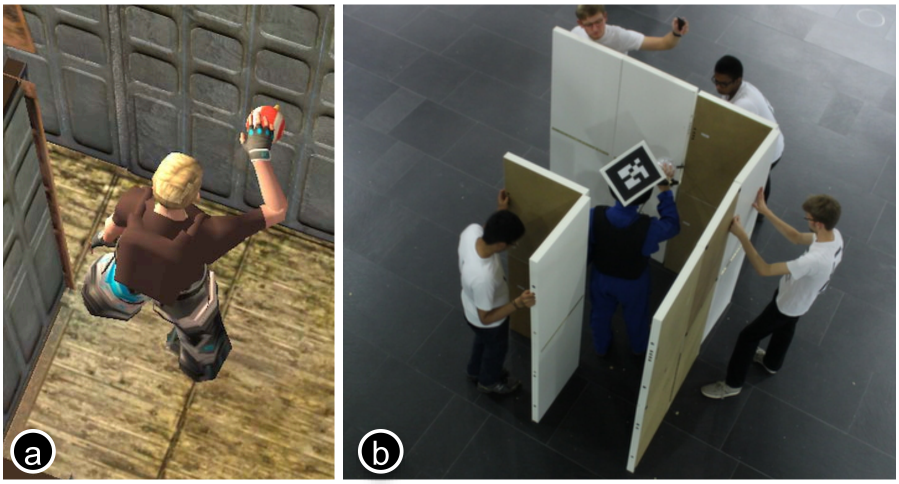

Figure 35: A switch; this human actuator also serves as sensor.

TurkDeck renders the switch using a stick with a ball attachment held by a human actuator. As the user pushes down, the human actuator follows the motion, simulating the path a physical knob would perform. The use of human actuators provides TurkDeck with substantial freedom in rendering the knob's response, including a wide range of motion paths, animated behavior, and force feedback. Marks on the stick and the adjacent boards serve as visual reference for knob height and the default angle.

HUMAN ACTUATOR AS SENSING SYSTEM Conceptually, interactive mechanisms such as the switch require a sensing mechanism that detects the state of the prop and updates the virtual world accordingly, e.g., an accelerometer built into the stick. While that is of course an option, TurkDeck bypasses sensing/tracking/recognition questions by letting human actuators perform the job. In the case of the switch, TurkDeck tasks one of the human actuators to watch the human actuator enacting the switch. This human actuator simply triggers a predefined animation sequence by pressing a button on a wireless presenter tool.

BODIES AS PROPS Next, the user finds a dead body and strips it off two bracelets (Figure 36). TurkDeck renders the body using a human actuator itself as prop. To help the human actuator assume the correct pose it displays the body's outline on the board.

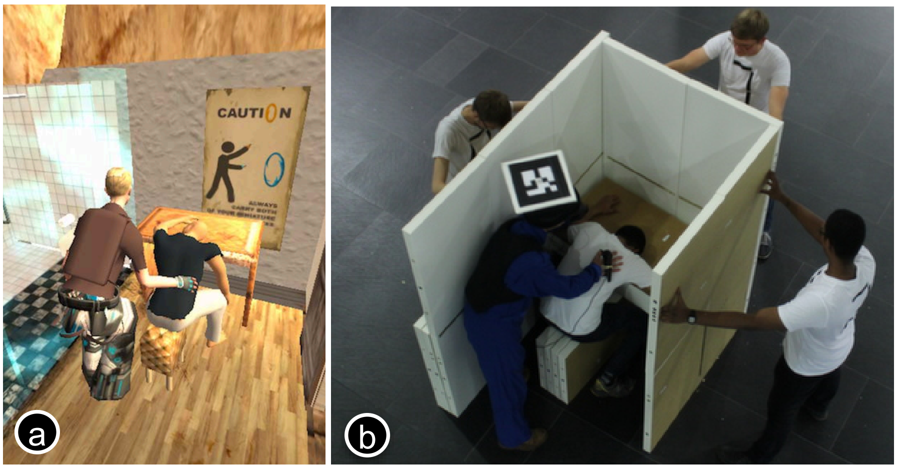

Figure 36: TurkDeck using body of a human actuator as prop.

ANIMATED MECHANISMS Next, the user has to get past the security mechanism shown in Figure 37: a "stomper". This is an example of an animated mechanism. When the user enters into the security mechanism, the stomper starts actuating, pushing the user against the wall. TurkDeck implements this using four human actuators, as shown.

Figure 38 shows how TurkDeck coordinates the human animation. (a) TurkDeck prepares human actuators by showing a pair of arrow icons that indicate the upcoming animation. Right before the animation starts the arrows perform a visible countdown (they blink three times). (b) Now the wall icon animates along the trajectory shown using the arrows. Because human actuators had the advance warning, they are ready to follow the movement of the wall in correct timing. TurkDeck uses the same visual language for all animations. (c), for example, shows a door (next to a switch) that will later rotate to open.

Figure 37: This "stomper" is an animated mechanism.

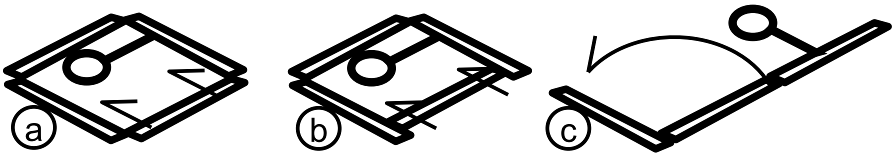

Figure 38: Animation based on human actuators: (a) the pair of (half-) arrows indicates that this wall will animate at some point in time. (b) During the animation, the wall moves along the arrows. (c) TurkDeck uses the same visual language for all animations; this is a door that will later open to the left.

STATIONARY MOVEMENT ("VEHICLES") Figure 39 shows a different type of animation. (a) In the shown scene, the user climbs a step, grabs the handle overhead, and rides the zip line. (b) TurkDeck implements the ride by making human actuators remove the step from under the user's feet, leaving the user dangling in the air. TurkDeck now plays the visuals of the world animating past the user—while the user remains stationary. (c) A few seconds later, the user reaches the end of the zip line, and typically bumps into the wall. TurkDeck implements this by two human actuators carrying wall segments walking into the user.

The instructions for the human actuators use the visual language described earlier, i.e., first arrows indicate that the wall at the end of the zip line is going to animate, then the blueprint of the entire world animates past the user.

The zip line is an example of a class of mechanisms we call "vehicles". Normally, users move though a stationary world ("real-walking" [68]). While in a vehicle, in contrast, users remain stationary and the world moves past them. TurkDeck implements vehicles this way, as it allows transporting users by an arbitrary distance—including vertical

Figure 39: The zip line is one instance of a vehicle.

displacement—with constant space requirements and constant effort for props.

On an abstract level, vehicles create an experience as it could have been created using Haptic Turk [8]. Another way of thinking of TurkDeck's vehicle mechanism is thus as a means for embedding Haptic Turk experiences into a larger, overarching TurkDeck experience. The resulting worlds have the form of node-link diagrams with real-walking areas as nodes and vehicles forming the links.

PHYSICAL EFFECTS Any TurkDeck mechanism can be and typically is complemented with physical effects. Riding the zip line, for example, is accompanied by wind, which TurkDeck implements using human actuators waving their boards. Other physical effects, such as spraying water from waterfalls, brushing leaves, etc. can be added to as required by the experience.

4.3 SCHEDULING AND TRACKING

The key objective of TurkDeck is to get the timing and the location of physical actuation right. TurkDeck's underlying system accomplishes this by scheduling human actuators, tracking users, and by coordinating the two with each other. Figure 40 summarizes TurkDeck's system architecture.

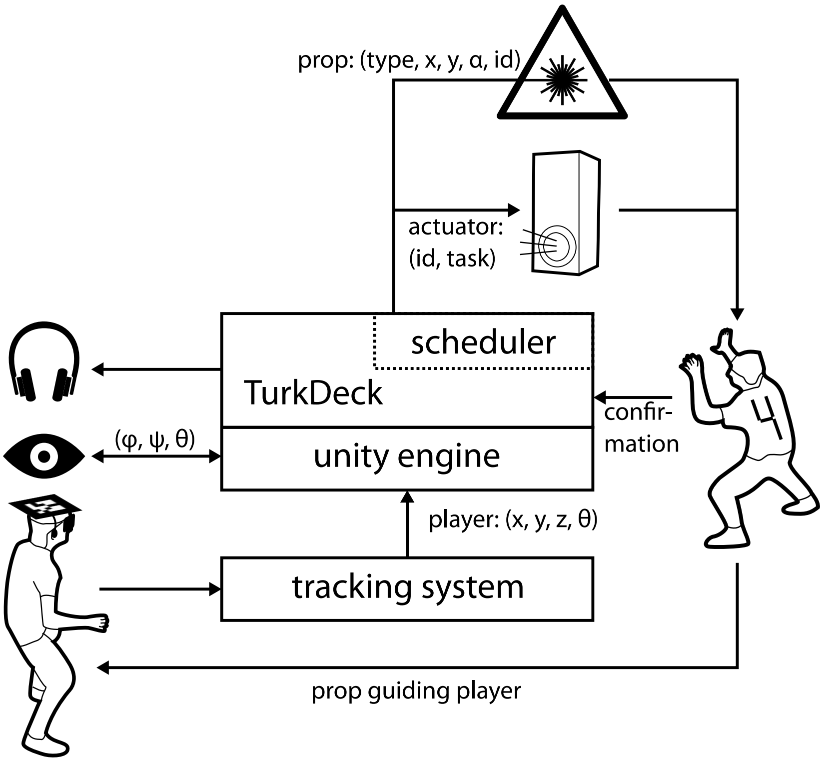

Figure 40: TurkDeck system diagram.

4.3.1 *The Scheduler*

TurkDeck's scheduler makes sure that human actuators physically "render" mechanisms before the user can touch them. Since this just-in-time approach is the key to TurkDeck's contribution of delivering large physical worlds with limited resources, getting the timing right is a key objective for the system.

We made good experiences with this very simple queue algorithm. (1) When the experience is launched, all mechanisms register with the scheduler. The system attaches a "collider" object to every mechanism, i.e., an invisible hull surrounding the object that triggers an event whenever the user enters or leaves. The scheduler sorts mechanisms by their expected chronological order. (2) At the beginning of the experience, the scheduler assigns the n human actuators to the n mechanisms the user will encounter first. (3) During operation, whenever the user leaves a collider, the scheduler frees up the associated human actuator and instantly reuses it for the next upcoming mechanism.

This scheduler also works across rooms. While TurkDeck uses portals to "fold" large virtual worlds into small physical spaces as in imposible spaces [62], the scheduler already starts tearing down the current room and sets up the room behind the portal, as early as the user starts walking towards the portal. The moment the user passes the portal, the room on the other side is already "there".

Note how our algorithm very aggressively allocates all its resources for future events, maximizing the time human actuators have to set up. This maximizes the speed at which a user can traverse the virtual world, at the obvious expense of reduced responsiveness for unexpected user behavior, such as walking backwards. Giving users additional freedom, such as the option to instantaneously turn around and go back, is possible, but requires additional human actuators. The same holds for bifurcations and large rooms. We can reduce this need for human actuators by trading it in for responsiveness, as we describe in the following.

4.3.2 *Delay Mechanisms*

If a room contains more mechanisms than the human actuators can set up in time, we protect the room with what we call a delay mechanism, i.e., a simple mechanism that can be set up quickly, but that requires users a specified time to get past. It consists of (1) a barrier that prevents access to the complex mechanism and (2) a mechanism that keeps the user occupied until a predefined duration has passed.

Figure 41 shows an example. A user reaching the zip line finds access blocked by a glass door. In order to unlock the door, the user has to crank the hydraulic mechanism that powers the door. Once oil pressure has been restored, the door slides open and the user can access the zip line.

Figure 41: This delay mechanism allows human actuators to delay the user until the zip line mechanism has been set up.

Note that the speed at which pressure builds up is designed to be non-obvious to the user. In reality, a human actuator determines the build-up of pressure—as with all interactive elements. This gives the human actuators control over the timing, allowing the user's task to complete just as the human actuators have completed the zip line mechanism. In essence, the delay element thus serves as a progress bar. One that, however, feels like it is under the user's control. Within limits, adding delay mechanisms allows adapting an experience to the number of human actuators available.

4.3.3 *Tracking the User with Precision*

TurkDeck uses a hybrid approach to track the user. First, a ceiling-mounted camera observes the user from above, tracking the user's head-worn ALVAR marker [15] (Figure 32c). This perspective from above works reliably, while human actuators and their boards tend to block almost any other angle. Second, the inertial measurement unit (IMU) in the head mounted display determines head rotation. We aggregate this IMU data with the ALVAR data in order to track the user during fast movements.

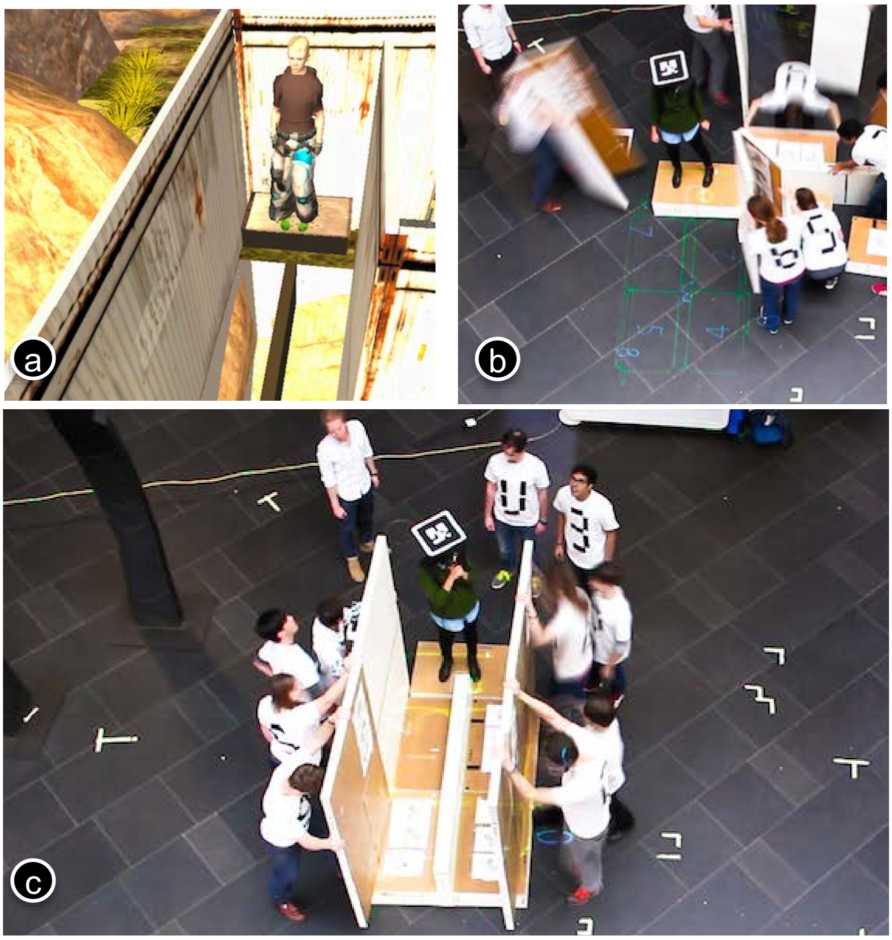

Figure 42: This position and delay mechanism allows human actuators to place the physical beam in reference to the user.

Some mechanisms require additional accuracy. Figure 43 illustrates this at the example of the user stepping onto a physical prop; accuracy matters here to make sure the user does not trip. TurkDeck addresses this with a variant of the delay mechanism. (a) When the user arrives at the pit, there is no way to cross. This forces the user to stand on a virtual foot-switch until a balancing beam is slowly raised into position. (b) In the physical world, there is no balancing beam. As the human actuators are setting it up, the well-defined position of the user serves as additional reference, allowing the actuators to place the beam in precise reference to the user's feet. (c) When the physical beam is set up, also the physical beam reaches its goal position. The user can now step forward and reliably hit the center of the beam without risk of tripping.

4.4 USER STUDY

To validate our approach, we conducted a user study. Every participant explored a sequence out of the *Lighthouse* experience in two conditions: with props placed and actuated by human actuators and without. Participants rated both experiences on a series of Likert scales. Our main hypothesis was that the props and human actuation would contribute to a more realistic and enjoyable user experience.

TASK Participants explored a sequence out of the *Lighthouse* experience and their task was to simply reach the other end of the experience. The experience included a range of elements and mechanisms, including the ledge, a door, a dead body, multiple portals, a stomper, two balancing beams

There were two conditions. In both conditions, participants could walk around the world freely and they experienced the virtual world by means of a head mounted display and earphones.

- The TurkDeck condition: the *Lighthouse* experience encompassed a simulation of its physicality, administered by ten human actuators. Combined, human actuators placed and operated a total of 65 props.
- The baseline condition: participants experienced the *Lighthouse* through the head-mounted display and the closed earphones only; there were neither props nor human actuation.

We used a within-subject design, so that each participant experienced the *Lighthouse* world twice, i.e., once in the TurkDeck condition and once in the Baseline condition, in counterbalanced order.

PARTICIPANTS We recruited 11 participants (three females) from our institution. Their age ranged from 19 to 29 years (*M*=23.0, *SD*=3.4). They had never experienced TurkDeck before. One participant lacked the ability to see in stereo.

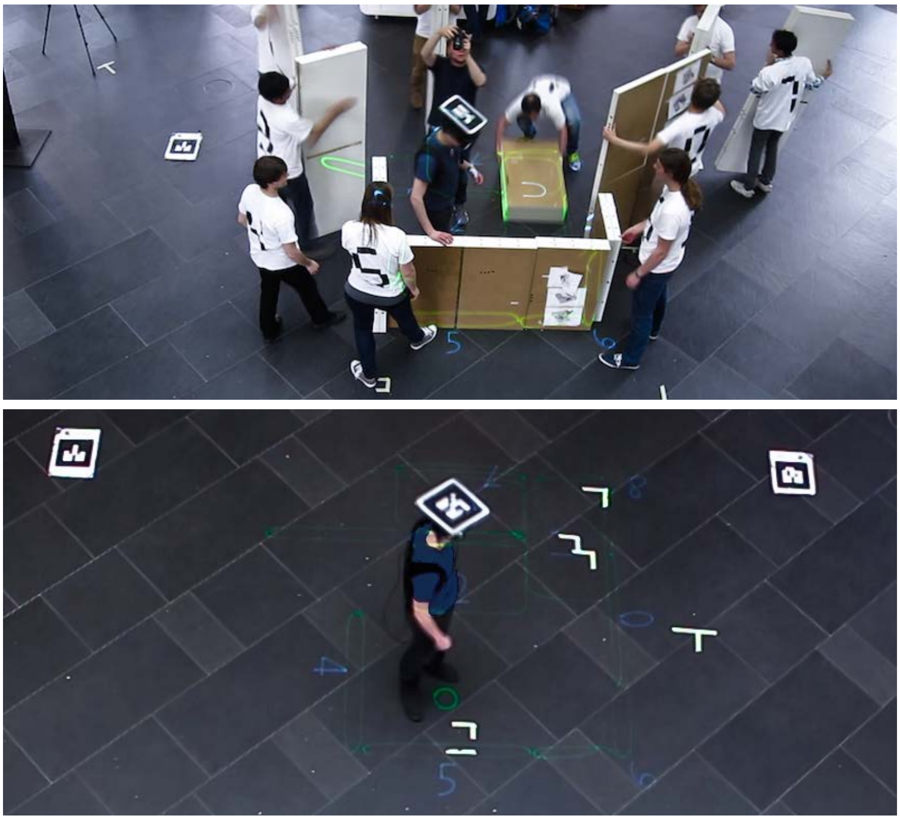

Figure 43: Participants went through the *Lighthouse* experience in the Turkdeck and the baseline conditions.

In addition, we recruited two groups of actuator participants to serve as human actuators in two sessions, i.e., one group of four and one group of three. The experimenter filled in additional human actuators to fill the ten-actuator slots required for the experience.

The two groups of participants were distinct, i.e., participants either served as participant or as actuator participant.

PROCEDURE For each participant the experiment started by dressing up in the head-mounted display, a set of earphones, a marker hat, and a backpack that contained the MacBook. After we calibrated the system, participants explored the *Lighthouse* experience until they reached the other end, which took between six and eight minutes. They then repeated the experience in the other interface condition, and finally filled in a questionnaire.

For each of the two groups of actuator participants, we provided 30 minutes of training ahead of time during which we explained the handling of the boards and the individual actuator displays (projection, sheets, audio). Then actuator participants assisted in three to eight TurkDeck experiences; they used the in-between baseline conditions to rest (10-15 minutes each). After they were done actuating their respective group of participants, all actuator participants filled in questionnaires about their experience.

RESULTS User participants rated their experience on average as 5.82 (*SD*=1.16) on a 7-point Likert scale (7=fun)—so clearly as enjoyable. This was also reflected by their ratings of realism 4.91 vs. 3.63.

Feedback about the TurkDeck condition was overall very positive. One participant explained with respect to the TurkDeck condition "What I mean is, that it was that realistic, that I was truly fearful standing on the edge or before the fire." Another participant said, "I really felt I was there, and I could only calm myself down by telling me it was not real." Some participant mentioned favorite moments, such as standing on the ledge. User participants agreed that the human actuators were responsible for the added realism (6.0/7).

The limiting factor in the experience turned out to be the calibration of the tracking system, which occasionally was off by up to 15cm. In the TurkDeck condition, this affected the sync between props and visuals, while it left the Baseline condition largely unaffected. This caused one participant to rate the TurkDeck condition lower than the baseline (3 vs. 5) while all participants agreed that "better tracking would make this more realistic" (6.0/7). An optical motion capture system can fix these issues, as mentioned earlier.

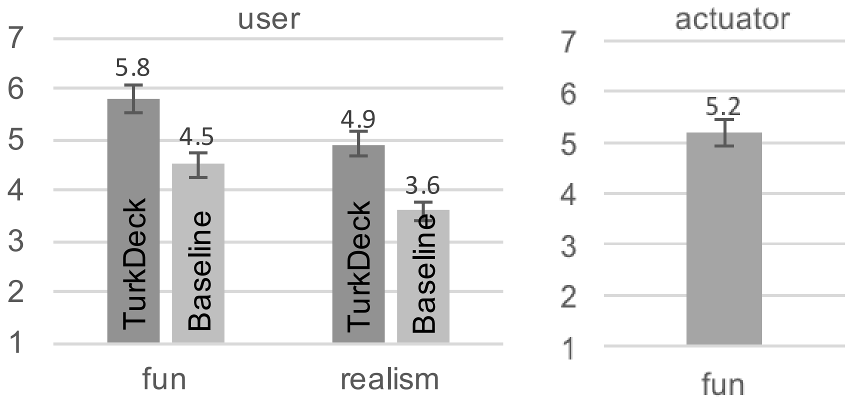

Figure 44: User participants rated their experience as more fun and realistic when in the TurkDeck condition. Error bars are +/-1 standard error of the mean.

As actuators, participants rated the experience as less enjoyable than as user participants, yet still as "enjoyable" (*M*=5.2, *SD*=1.9). The actuator participants explained that their experience was driven by two factors: their desire to support of users' experience and the team experience among the human actuators.

Actuator participants rated all four components of the instruction system as useful (projection 6.1/7, auditory 6.0/7, sheets 5.0/7, landmarks 3.8/7) confirming our design decisions. In addition, we observed human actuators use boards placed by other human actuators as a spatial reference. Five actuator participants mentioned fatigue—an expected effect given the use of the heavier high-fidelity boards.

In summary, our user study provides initial validation for the TurkDeck concept. Most importantly, participants enjoyed their experience in the TurkDeck condition more than baseline and rated it as more realistic. The actuator experience turned out to be enjoyable as well, primarily caused by the social nature of the system.

4.5 DISCUSSION

The main contribution of TurkDeck is that it “scales” prop-based virtual reality, i.e., it can produce arbitrarily large virtual worlds in finite space and with a finite amount of props. TurkDeck achieves this as follows: (1) By using human actuators to rearrange props, TurkDeck reuses props, allowing it to simulate large virtual worlds with a small set of physical props. (2) By designing props to be generic, TurkDeck is able to use the same props to represent a wide range of physical objects and effects. (3) By physically rendering only the user’s immediate periphery, TurkDeck can tear down or rebuilt the world in a constant amount of time. This allows TurkDeck to handle situations of abrupt change to the virtual world, e.g., when the user walks through a portal or when loading a new experience. (4) By exploiting the smarts of human actuators, TurkDeck can animate objects and sense user actions.

On the flip-side, experiences created by TurkDeck are limited by the speed and accuracy of the human actuators and in our case also the tracking system. Readers attempting to replicate our work can achieve better results by tracking the user using an optical motion capture system instead. Although the system is limited in that our *Lighthouse* experience was created by hand, the mechanisms that the experience is composed of are reusable. They are also intended to give readers a sense of how to design their own experiences for a given number of human actuators.

We have presented TurkDeck, a physical virtual reality based on human actuation, demonstrating that that human actuation not only works in 3DoF but also in 6DoF virtual reality. Since humans are versatile, they can provide various actuations from simulating motions to rearranging props. While TurkDeck enables more physical interactions in virtual reality, it employs more human actuators. Although our study shows that the experience of human actuators was decent, it was still inferior to the experience these people could have had, had they participated as a user, reducing the willingness of being human actuators. To resolve the issue, our next idea is to bring the same enjoyment to human actuators—by making them users.

5

NO MORE HUMAN ACTUATORS, ONLY USERS: MUTUAL TURK

From our previous studies with Haptic Turk and TurkDeck, while the experience of human actuators was decent, it was still inferior to the experience these people could have had, had they participated as a user. We address this issue by making everyone a user. We introduce *mutual human actuation*, a version of human actuation that works without dedicated human actuators.

Mutual Turk is a real walking virtual reality system that implements mutual human actuation. The key idea behind Mutual Turk is that it runs two users at the same time, synchronizing their experience so that every time one user is manipulating an object in her virtual world, the other user is subjected to forces presumably caused by something in his virtual world.

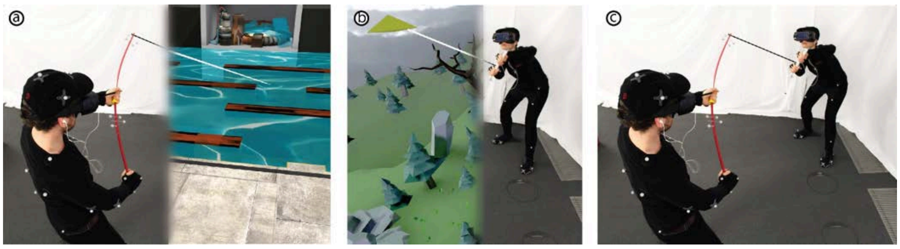

Figure 45: (a) This user, alone in his virtual world, is trying to pull a huge creature out of the water. He feels how the creature is struggling and pulling on his fishing rod. (b) At the same time, this other user, also alone in her virtual world, is struggling to control her kite during a heavy storm, which is whipping her kite through the air. (c) While users' experiences of force might suggest the presence of a force feedback machine, Mutual Turk achieves force feedback instead using shared props that transmit forces between users. The system orchestrates users so as to actuate their prop at just the right moment and with just the right force to produce the correct experience for the other user.

Figure 45 shows an example. (a) One of the two users, alone in his virtual world, is trying to pull a huge creature out of the water. Through the fishing rod he feels how the creature is struggling. (b) At the same time, the other user, also alone in her virtual world, is struggling to control her kite during a heavy storm, which she feels pulling at her kite. (c) In reality, both users are connected by means of a shared prop, so that all forces they output become input to the other user. This is the main concept behind Mutual Turk. Mutual Turk's main functionality is that it orchestrates users so as to actuate props just at the right moment and just with the right force to produce the correct experience for the other user.

To maximize immersion, we generally design Mutual Turk experiences so as to set users' expectations of upcoming haptic effects. The kite/fishing rod prop, for example, achieves this as illustrated by Figure 46a. The particular arrangement of logs in this pool gives the fishing rod user a steering task to perform, i.e., in order to reel in and save the struggling creature at the end of the fishing line, the user has to pull the fishing rod left and right in just the right order so as to make the creature pass in between the logs.

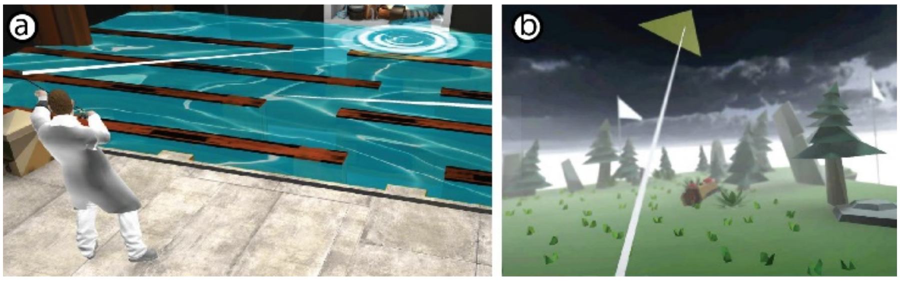

Figure 46: Visual puzzles allow us to pre-define the sequence of forces the respective user will produce.

Meanwhile, the up-front knowledge about the fishing rod's motion sequence allows Mutual Turk to manage the kite user's expectations of what she is about to feel. As shown in Figure 46b, Mutual Turk makes the white flags in the background fly in the direction of where the other user is about to steer, suggesting to the kite user that she wind is about to change. Once the fishing rod user starts pulling and the kite user experiences the directional pull, this pull matches the kite user's expectations.

5.1 SHARED PROPS

The key to enabling mutual human actuation is the use of *shared props*—props allow users to exchange forces without revealing that these forces are generated by another human. If users' hands or any other part of the users' bodies were ever brought into immediate physical contact, this would instantaneously give away that the other side is human (through, e.g., skin softness, temperature, moisture, shape of hands). This is what the shared prop prevents. Thus, one way to think of Mutual Turk's shared props is as a means of masking the information exchanged between the users.

Different prop designs enable different levels of expressivity. We explored five: continuous force (most expressive), moving, impact, contactless sensations, and rearranging props (least expressive).

1. CONTINUOUS EXCHANGE OF FORCE BETWEEN USERS' HANDS The kite/fishing-rod prop from Figure 46 uses string to connect the two handles. This specific design allows the prop to eliminate much of the information about what is located at the other end of the prop—the only information that is transmitted is the direction and magnitude of the tension. This allows the system to re-envision the many dimensions that were filtered out, such as to render the kite at the end of a 100x longer tether.

The reason why the fishing rod prop uses the additional tether for "obfuscation" is that the interaction is particularly expressive: the two users exchange forces over an extended period of time during which they modify their force output hundreds of times. And users feel each other's force "signal" well, as they hold the prop in their hands.

In contrast, less expressive interactions require less obfuscation, thus allowing the use of shared props that couple users more rigidly. This thereby allows transmitting additional degrees of freedom, such as translation etc. In the remainder of this section, we show four different classes of such less expressive interactions.

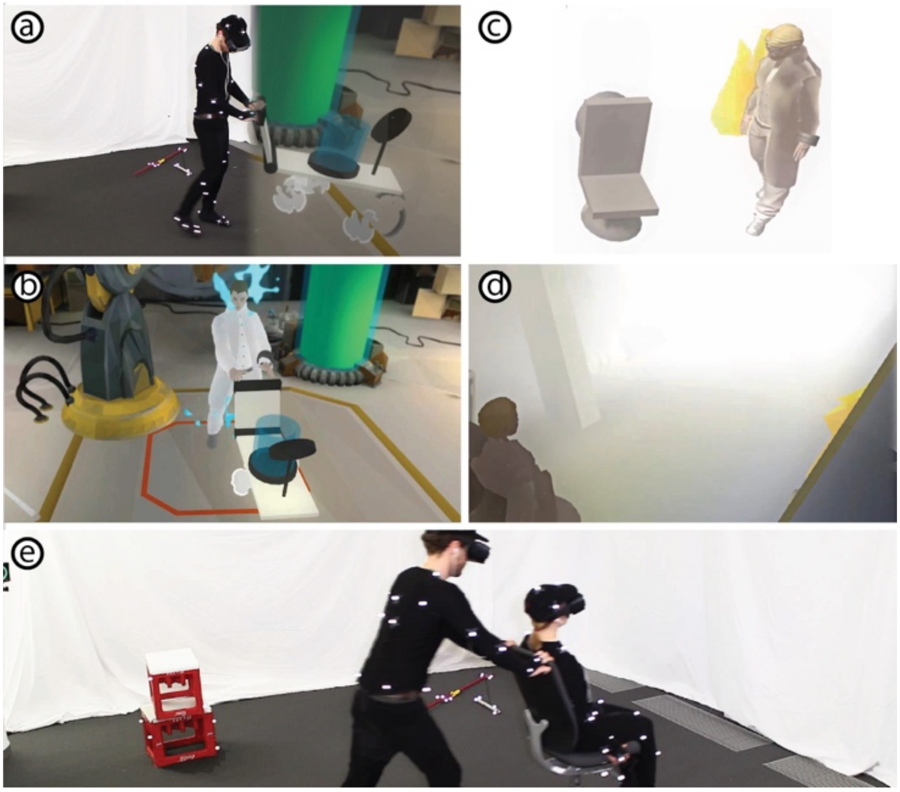

Figure 47: One user pushes cart around while the other enters and rides an escape pod.

2. CONTINUOUS MOTION In Figure 47a, the user pushes an empty cart under a faucet. (b) He watches as the faucet drops water into the tank. (c) Meanwhile, in the other user's world, the world is collapsing and our user is making a dash for the escape pod. She hops onto the escape pod (thereby giving weight to the water tank in the first user's world). (d) She then rides the automated pod down an evacuation tunnel—(propelled by the first user pushing his cart), (e) just as the first user starts to push his (now much heavier) cart on to next destination.

The office chair prop transmits movement and rotation in one direction. In return, the user sitting on the chair affects the chair's inertia. Unlike the interactions in the previous category, only one user's hands are involved in driving the office chair. This allowed us to drop the tether and use a rigid prop instead.

3.IMPACT The user in Figure 48 sees himself walking in stormy weather; he sees huge hailstones shooting down from the sky at an angle, hitting his body at various locations. (b) In the meanwhile, the other user is fighting back a monster using an improvised weapon made from a plastic tube she found at the lab.

The foam stick used in this scene touches the other user for only very brief periods of time, which properly obfuscates the origin of the force.

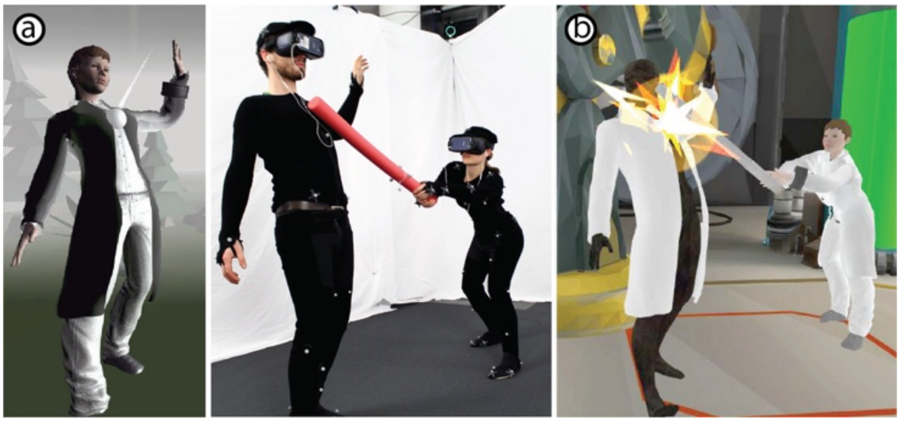

Figure 48: One user is getting bombarded by hail, as the other user is fighting a monster.

4. CONTACTLESS SENSATIONS In Figure 49a, our user is trying to fight her way back to lab against very heavy wind. (b) Meanwhile, our user in the other world is trying to get a fire going to distill the emulsion created earlier.

The forces exchanged in this scene are obviously minimal. However, the interaction produces a strong tactile sensation (and certainly properly obfuscated).

5. REARRANGING PROPS Finally, Figure 50a shows a user waiting for a series of pillars to rise in order to allow her to cross the pit ahead of her. As she lowers her right foot to probe the space below, she can

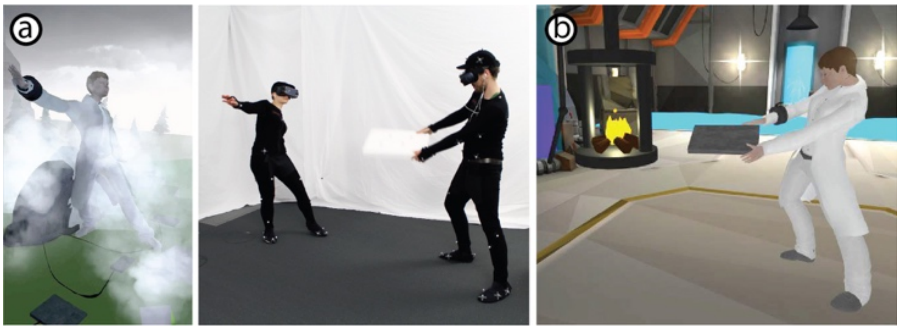

Figure 49: One user is trying to fight her way through heavy winds, while the other is trying to get a fire going.

feel the void. Once she sees that the pillar has fully risen, she can step on it. (b) In the meanwhile, the other user is solving a puzzle that requires him to place numbered boxes on matching tiles.

This is the least expressive type of exchange between two users as no physical contact between the two users is ever established. It thus is also the most obfuscated type of interaction.

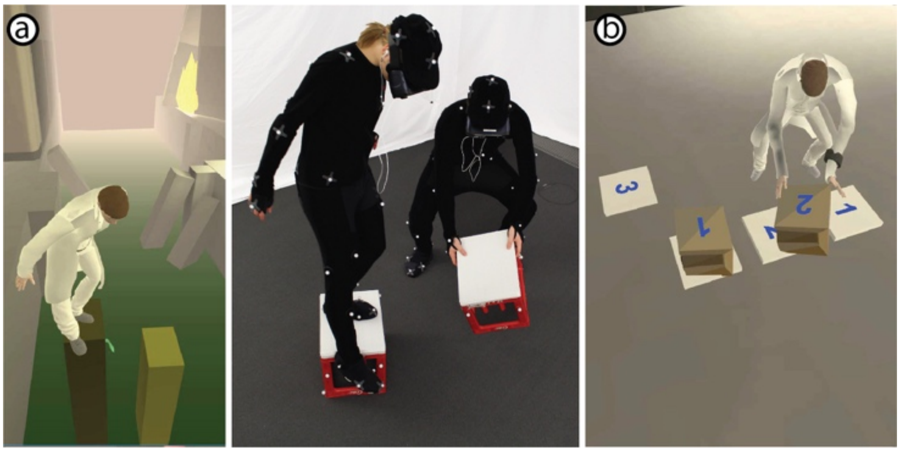

Figure 50: One user is waiting for the next pillar to rise, while the other user rearranges boxes to solve a puzzle.

5.2 SYNCHRONIZING USERS

As discussed earlier, the main function of the Mutual Turk system is to serve as scheduler, i.e., to orchestrate the two users in a way that their experiences are properly synchronized in time and space.

So far, we only looked at what we call action sequences, i.e., sequences during which the users already hold the shared prop and the subsequent interaction emerges largely from the use of this prop.

As illustrated by Figure 51, complete Mutual Turk experiences are more encompassing than this. Mutual Turk must not only synchronize the use of the shared props, but also their acquisition and disposal. A typical scene consists of a period of real walking within a designated area, the acquisition of a prop, the use of the prop forming an action sequence, the disposal of the prop, and return to unencumbered real walking. Experiences are then sequences of such scenes.

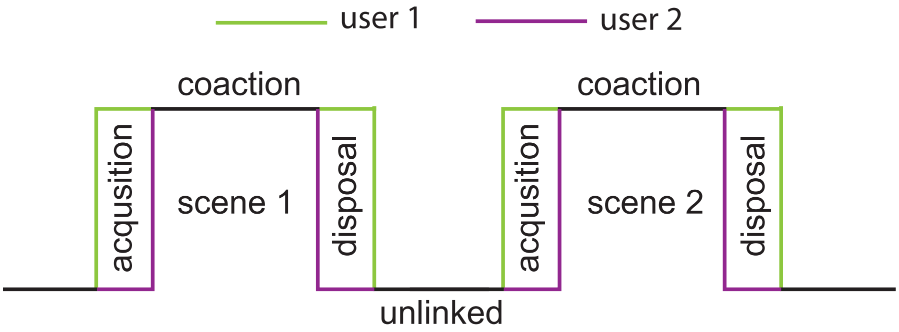

Figure 51: Mutual Turk experiences typically consist of multiple scenes, each of which consists of prop acquisition, use, and disposal.

PROP ACQUISITION To show an example, we returns one more time to the kite/fishing rod example. This time we are joining the two users early. They are still real walking unencumbered; all props are located on the ground and within the tracking volume.

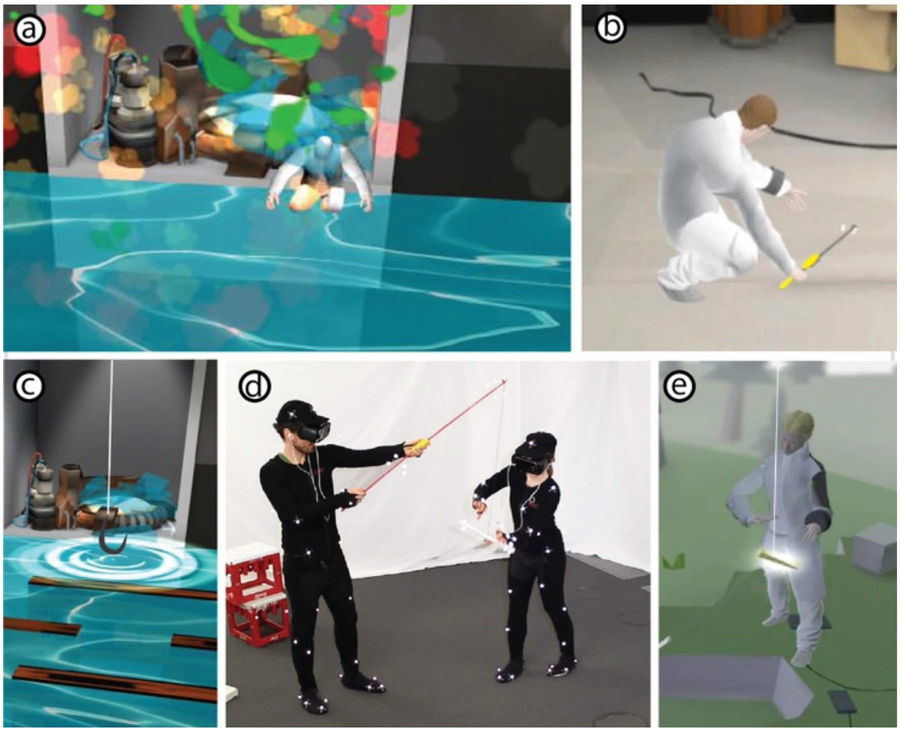

Figure 52: Acquisition of props

Figure 52: (a) The scene starts by the creature across the pool crying for help as it falls into the water. (b) The fishing rod, lying on the ground next to our user, might just be the tool to rescue the creature.

Our user picks it up and (c) holds his its end over the spot where the creature just fell in, waiting for the creature to reach for it. (d) As the position of the kite handle in the physical world stabilizes, (e) our user in the other world notices a kite stuck in a tree. The kite's handle is hanging down; she reaches out and grabs it.

Once both users have acquired their props, the action sequence begins and the fishing rod user can rescue the creature by steering it through the gaps between the logs, as already shown in Figure 46.

PROP DISPOSAL Towards the end of the action sequence, the fishing rod user has reeled in the creature and pulled it out of the pool. Figure 53 : (a) He now slowly lowers his fishing rod, ready to drop it.

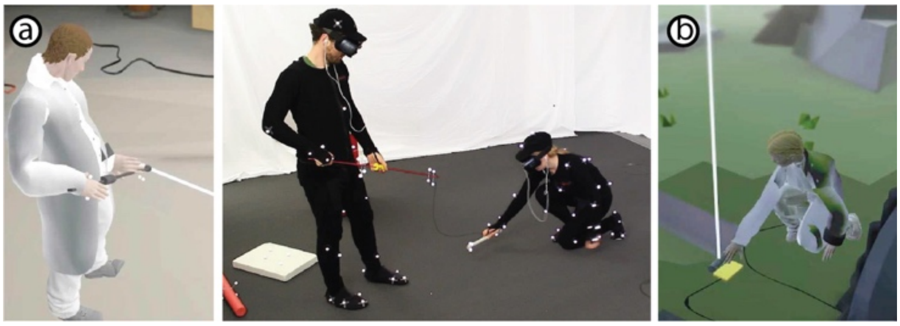

Figure 53: Disposal of props

(b) Meanwhile in the other world, the kite user has succeeded at collecting enough lightning energy using her kite. She returns to the tree, tugs the kite under one of its roots, and leaves it there.

Now that the kite user has let go of the prop, the fishing rod user has sole control and can put the fishing rod away. This completes the disposal sequence and both players engage in the next real walking sequence, walking towards the next prop acquisition.

5.3 TRACKING USERS AND THEIR PROGRESS

As illustrated by Figure 54, Mutual Turk runs inside of Unity 3D and is written in C#. This includes (1) a native OptiTrack NatNet Unity plug-in that receives the tracking data directly from OptiTrack Motive, (2) Petri net server and client and (3) our demo experience called "*Edison, Jr.*" (in which users have to perform a series of experiments to help their ancestor regain physical form).

Headsets To allow for unencumbered real walking, we used Samsung S6s mounted into GearVR headsets with earphones attached. Both headsets run their own Unity app where a Mutual Turk client and the adventure experience are embedded. Via our wireless network, the Mutual Turk client receives the tracking data and communicates with the Mutual Turk server to synchronize its Petri net with the other Mutual Turk clients.

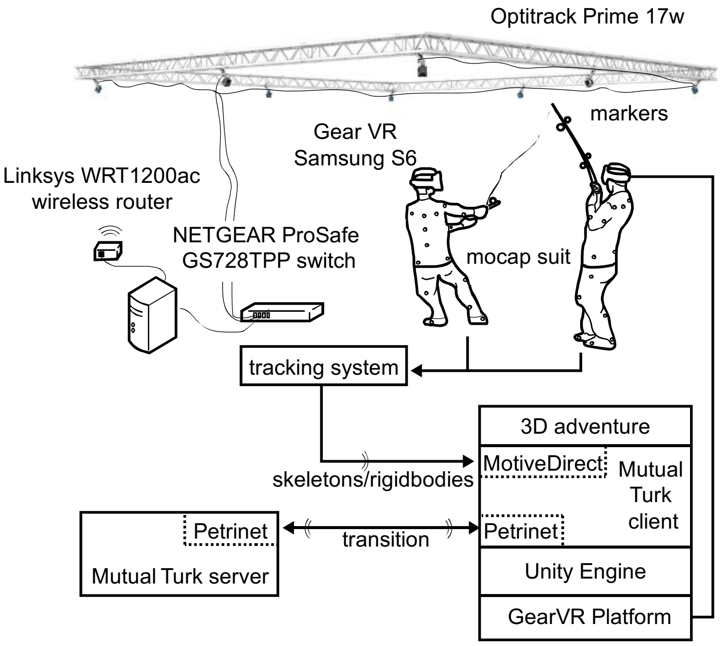

Figure 54: The Mutual Turk system diagram.

Tracking We use nine OptiTrack Prime 17w cameras to track a 5 m x 5 m tracking space, running the OptiTrack Motive 2.10 tracking software. Users wear motion capture suits. To make props trackable, we attached rigid body markers, 6.7 mm to 9.5 mm. Figure 55 shows where the markers are attached to our shared prop.

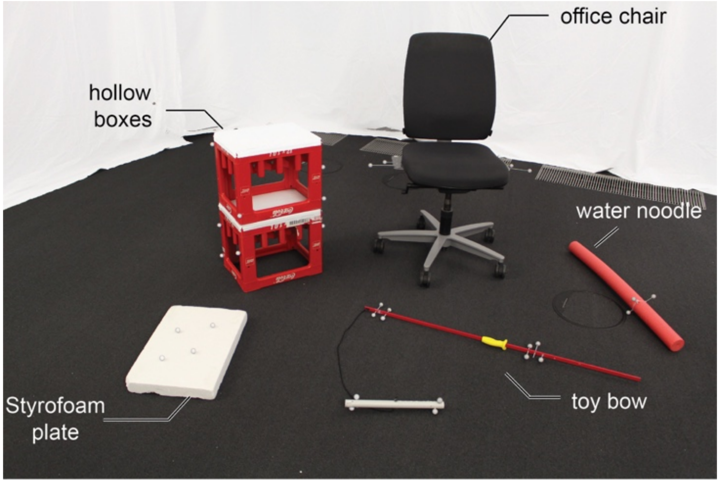

Figure 55: All props used in the Edison, Jr. experience.

Mutual Turk runs in real time with two users. We achieved 40+ fps by making VR scenes low-poly, simple lights, etc. In addition, we enabled time warping to guarantee interactive rates. The maximum delay between visual and haptics was around 25ms. The devices receive tracking updates wirelessly at 120 Hz with ping interval 5ms in average.

5.3.1 *Tracking Acquisition and Disposal*

Mutual Turk determines when to advance the global timeline using simple rules, such as "fishing rod user is touching the fishing rod and the fishing rod prop has started to move".

In order to detect acquisition and disposal with additional accuracy, we overwrite Unity's collider with the following custom code. (1) Our system approximates the volume of the user's body by padding the mo-cap suit marker locations with volumetric primitives as shown in Figure 56. (2) Our system determines collisions by ray casting from the body primitives to the props (down-sampled to 10 rays per cubic meter in order to run on the mobile phones in the GearVR headsets). (3) Our system determines that an object has been picked up, if the standard deviation of the position offset history in the past 0.5 seconds is > 1cm.

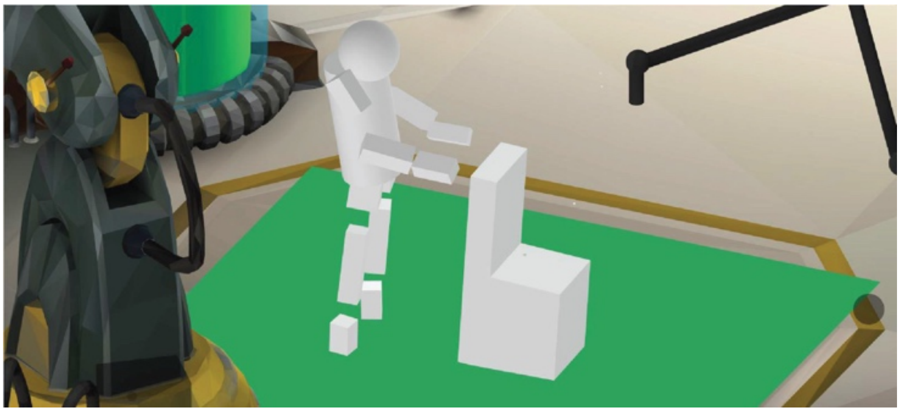

Figure 56: All props used in the Edison, Jr. experience.

5.3.2 *Petri net*

Internally, Mutual Turk considers the two users and their acquisition and disposal of props as a concurrent state machine. It manages this state machine as a Petri net [50]. This allows Mutual Turk to ensure that the overall story arc does not progress until both users are ready for it. The Petri net is also useful for level designers to detect and avoid potential dead locks, i.e., situations where both users would be waiting for each other.

Figure 57 shows the Petri net of the fishing rod vs. kite acquisition sequence described earlier. As we see, in the first half of the Petri net, the fishing user and the kite user have no influence on each other. The fishing user has the freedom to pick up or drop the fishing rod anytime and the kite user has the freedom to walk around as well. Only when they both are in their correct respective locations and the kite user has grabbed the handle, both users get to move on. One can extend the Petri net to continue the experience of the remaining user

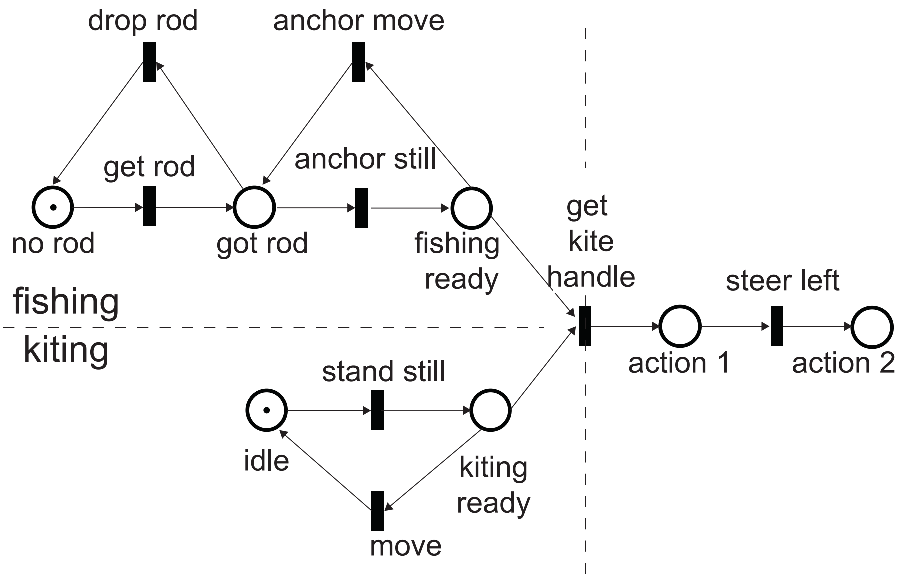

Figure 57: The Petri net diagram that governs the acquisition sequence of the fishing-vs.-kite scene.

as a single-user without mutual haptics experience if one of the users refuse to progress.

5.3.3 *Tracking and Extrapolation During Action Sequences*

Mutual Turk tracks users' props at all times. During action sequences, it extrapolates the props' movement, which allows Mutual Turk to anticipate interactions between the prop and the other user. In the hail vs. zombie scene (Figure 48), for example, Mutual Turk extrapolates the movement of the foam stick to determine when and where it will hit the other user. Based on this, the system either generates a new hailstone with and send it off towards the anticipated collision point or it alters the movement path of an existing hailstone, so as to hit the predicted location. Until the impact actually occurs, Mutual Turk continues to track prop and user and readjust the movement path of the hailstone accordingly.

In the fishing rod vs. kite scene, Mutual Turk needs to know the amount of tension on the tether, e.g., in order to determine whether the user is pulling hard enough to reel in the creature, but also to render the kite and fishing line visuals properly. Figure 58 illustrates how the fishing rod/kite prop allows Mutual Turk to sense this tension. The key idea is that the prop bears two markers on the fishing rod side. The angle between the two markers indicates how much the rod is currently bent, which indicates the applied force.

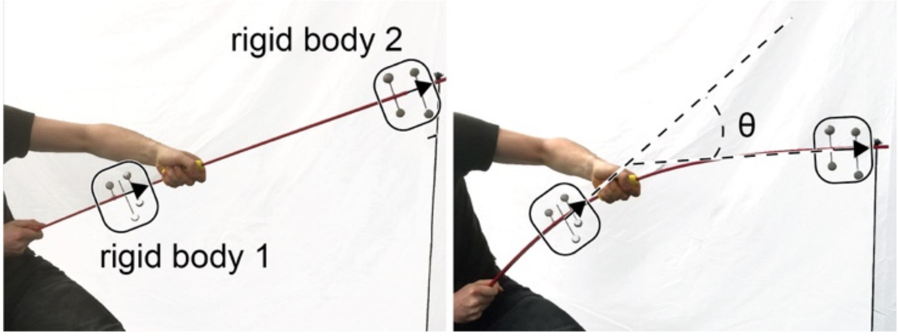

Figure 58: Mutual Turk computes the tension applied to the fishing line based on how much the prop is bent.

5.3.4 *Placing Props in a Limited Tracking Volume*

We generally design our experiences so that props are located along the edge of the tracking space and make users return props to their original place after use. This keeps the center area free of obstacles, allowing us to use that space for real walking. To prevent users from accidentally stepping on any props, Mutual Turk camouflages the props that are not currently available with virtual objects, such as by placing a big virtual robotic arm in the same location where the boxes are.

5.4 DESIGNING A MUTUAL TURK EXPERIENCE

Designing Mutual Turk requires additional effort, because any interaction has to satisfy two user experiences at the same time. When we designed the *Edison Jr.* experience described throughout this paper, we proceeded as follows.

1. DESIGNING A SINGLE USER EXPERIENCE We started by designing a user experience for a single user. In order to balance both users' experiences, we composed half of the experience from scenes where the first user actively actuates the environment (and thus the other user) and the other half from scenes where the first user primarily experiences actuation by the environment (and thus the other user). Based on this experience, we create matching passive props. We then tested the experience using dedicated human actuators.

2. IDEATING THE OTHER EXPERIENCE BASED ON THE FIRST ONE We then ideated multiple scenes for the second user that might take place during the first single user's experience, while considering the first user's experience as design constraint. We combined the respective props into shared props and tested them. In many cases, we succeeded at finding second user experiences without modifying the first

user's experience; in some cases, we revisited the first user's experience in order to improve the second user's experience.

3. RESTRUCTURING THE TWO TIMELINES We then tested the experience and swapped some scenes and tweaked scenes until both experiences flowed well.

In the particular case of the *Edison Jr.* experience presented throughout this paper, we then doubled the length of the overall experience by combining the first user's experience and the second user's experience into a "canon", i.e. while one user is on the 1st half of his or her experience, the other is in the 2nd half.

5.5 USER STUDY

We conducted a user study to validate our Mutual Turk system. We recruited 12 participants (age 19-23) in six pairs from our institute. Three participants had never worn a head mounted display and none had experienced full body motion capture in VR before.

TASK AND PROCEDURE Each participant experienced a 10-min subset of the *Edison Jr.* experience (flying kite, pushing cart, using fishing rod, riding escape pod) using Mutual Turk and a control condition that only had passive haptics. The order was counter-balanced. After each condition, they filled in a custom questionnaire (measuring overall enjoyment and perceived realism) and the Presence Questionnaire [70].

RESULTS To better understand what caused the higher enjoyment, Figure 59 shows all the presence scores. Mutual Turk received higher overall presence score (5.1/7) vs. (4.7/7) ($t(22) = 2.1$, $p < 0.05$).

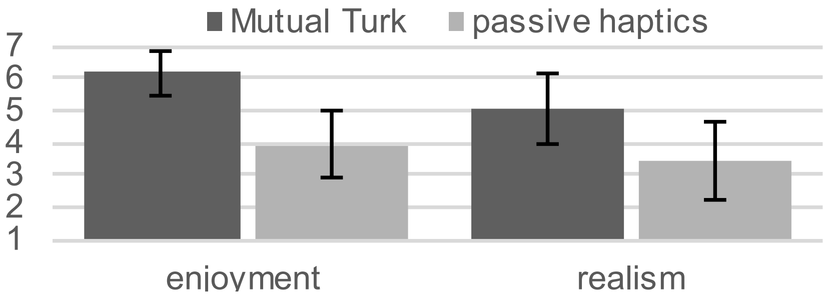

Figure 59: Participants enjoyed their experience more in the Mutual Turk condition

Figure 60 shows the main result of our study: participants enjoyed their experience significantly more in the Mutual Turk condition than in the passive haptics baseline condition (6.2/7) vs. (3.9/7) (Student's t(22)= 6.0, p<0.01). In terms of perceived realism, Mutual Turk received significantly higher rating as well (5.1/7) vs. (3.4/7) (t(22)= 3.4, p<0.01).

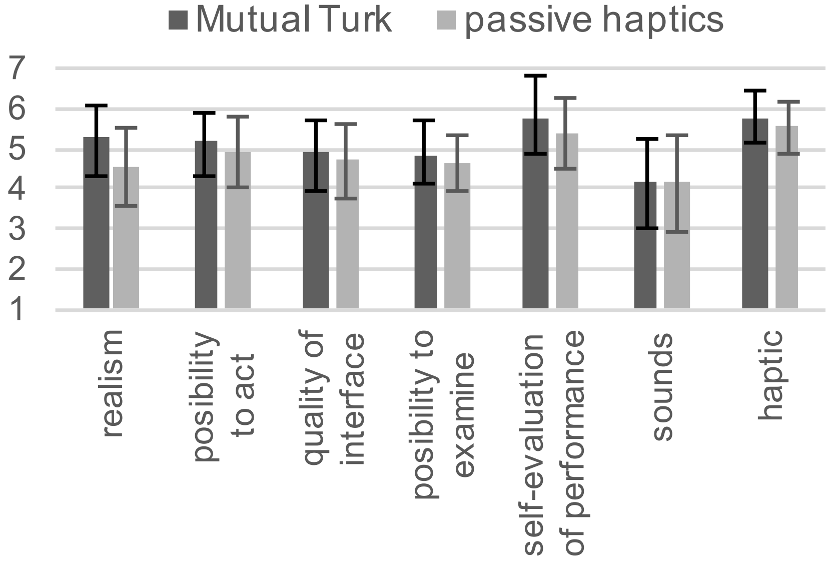

Figure 60: The presence scores in the two conditions

All participants said that they enjoyed the experience more with Mutual Turk because they could feel the force feedback. “It was very crucial for me to have the force feedback when flying a kite and fishing”, said p1. Participants did feel the force feedback was matching to their expectation. “It felt exactly as what would happen in the virtual world”, said p7. No participant experienced any performance issues during the test. “The system is responsive and real-time both on tracking and haptics”, said p2. Although all the participants had never experienced full-body motion captured VR, p10 explicitly said “the real-walking VR only amazed me in the beginning for a couple of minutes, but later it was all about the haptic feedback”.

5.6 DISCUSSION

The main contribution is the concept of mutual human actuation. The key idea is to run pairs of users at the same time and have them provide human actuation to each other. Our system, Mutual Turk, achieves this by (1) offering shared props through which users can exchange forces while obscuring the fact that there is a human on the other side, and (2) synchronizing the two users’ timelines such that their way of manipulating the shared props is consistent across both virtual worlds. We demonstrate mutual human actuation with an example experience in which users pilot kites though storms, tug fish out of ponds, are pummeled by hail, battle monsters, hop across chasms, push loaded carts, and ride in moving vehicles.

The main benefit of this approach is that it eliminates the need for dedicated human actuators, instead allowing everyone to enjoy the experience of a user. At the same time, Mutual Turk still offers the benefits of human actuating system, i.e., it allows creating human-scale force feedback without mechanical machines.

The main limitation of Mutual Turk is that designing experiences for mutual human actuation requires additional care and design skill, as each scene is subject to at least twice the number of design requirements as regular virtual reality scenes, which tend to be designed around a single user. While designing for Mutual Turk requires extra care, it does indeed allow telling encompassing stories. The 10 interactions are in fact the snapshots from a single 30 min experience *Edison Jr.*: the user awakes, meets the ghost of Thomas Edison's, who instructs the player to harvest energy from a thunderstorm through a kite, save his new body from a pool using a rod, and finally collect items with a cart to revive him. We see mutual human actuation is a tool to add active haptics to any type of VR experience, e.g., theater plays, circus shows, theme park rides, etc.

Furthermore, Mutual Turk users can make mistakes while working on their tasks or they may choose to explore the world differently (e.g., leave the tracking volume) from how the system incentivizes them to. Mutual Turk handles this not that differently from regular real-walking VR systems. Mutual Turk uses visual guides to discourage users from touching one another/leaving the tracking space. That said, in our user study all participants followed the story arc.

We have presented Mutual Turk, a version of human actuation that works without dedicated human actuators. Mutual Turk resolves the issue that we have found in the previous studies with Haptic Turk and TurkDeck—human actuators did not enjoy as much as users and thus less volunteers. With Mutual Turk, we bring human actuation to a more applicable level where all users are fairly treated, having immersive experiences while providing forces to each other in the meantime.

6

HUMAN ACTUATION FOR A SINGLE USER: ITURK

In previous chapters, we have explored the concept of human actuation in many aspects such as physical interactions, degrees of freedom, etc. One of the aspects is the number of human actuators. While Mutual Turk requires only two users/human actuators to run the system, we further lower the bound to one user in order to showcase that human actuation is applicable to single user experiences which is one of the big genres of current VR applications.

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 [10]) while physically walking through the tracking space.

Figure 61: (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.

Figure 61 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.

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 62, i.e., one foldable board and one pendulum.

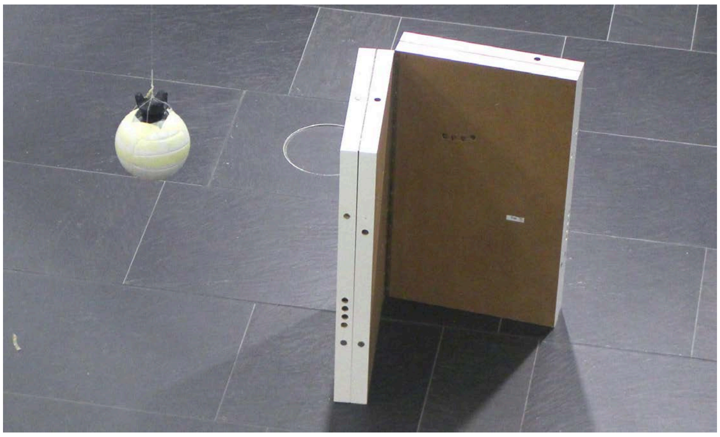

Figure 62: Our demo experience is built around only two props: (a) a board that can be reconfigured by the user and (b) a round prop suspended from the ceiling that starts to swing when pushed or hit by the user.

6.1 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** bold. We typeset every step in which the user reconfigures a prop in ***bold italics***.

As shown in Figure 63, (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, textbf*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.

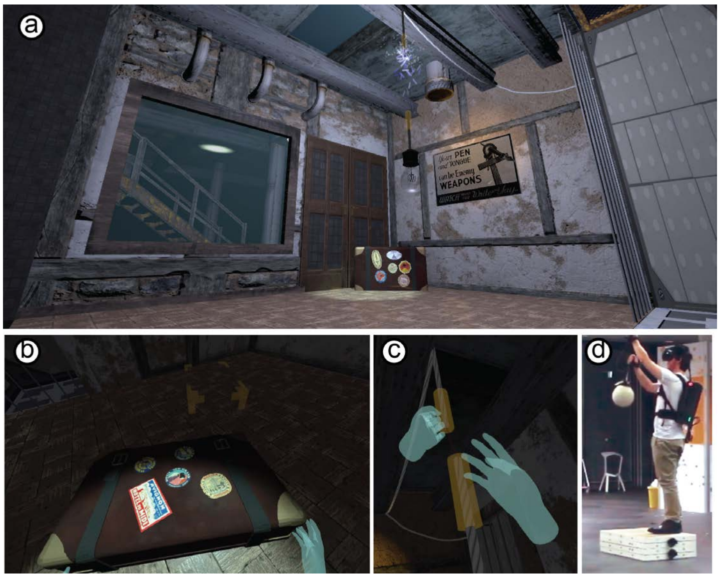

Figure 63: (a) The first room of our example experience requires users to (b) move a suitcase, lay it flat, and (c) step on to short circuit a cable. (d) This is supported by two physical props, i.e., a folded board and a pendulum.

As shown in Figure 64, the adjacent room is physically overlapping with the first room. We achieve this by implementing the impossible spaces [62], 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. 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.

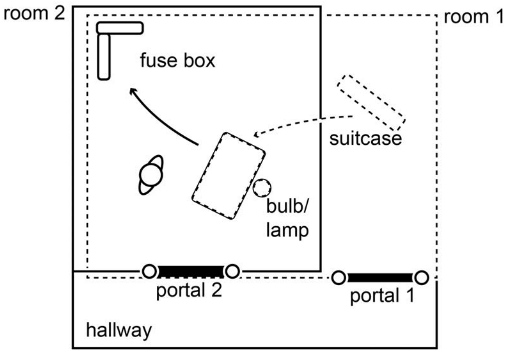

Figure 64: The fuse room is designed to geometrically overlap with the first room. This allows the two physical props to be located in both rooms.

As shown in Figure 65, (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.

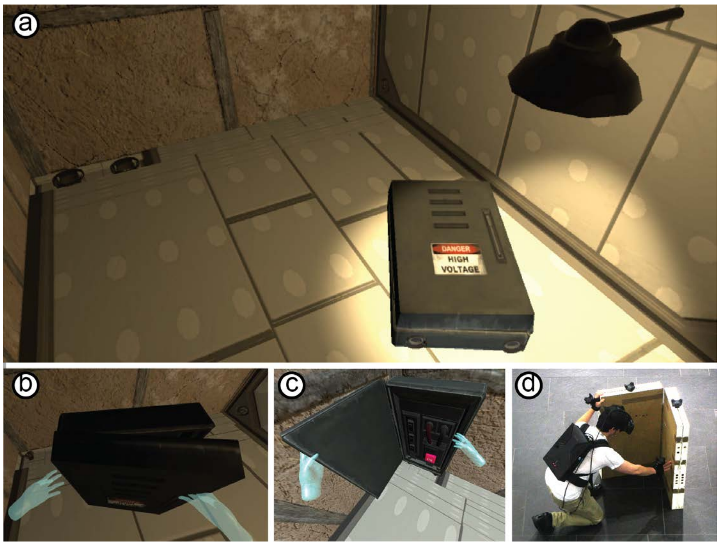

Figure 65: (a) The second room requires users to move and erect a fuse box, (b) open it, and (c) push the "on" button. (d) This is supported by the same physical prop that served as suitcase in the previous room.

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 66, 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.

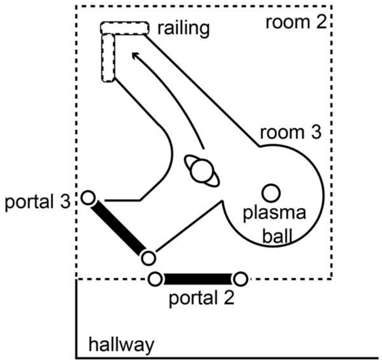

Figure 66: The reactor room also overlaps with the other rooms.

As shown in Figure 67a, the user walks to the end of the walkway and ***presses*** 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.

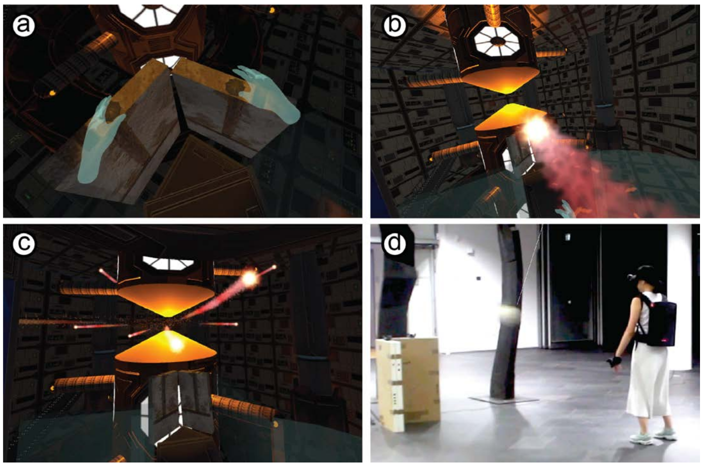

Figure 67: (a) The reactor requires users to press and hold two buttons on the railing, in order to reveal the plasma ball, which (b) users launch by hitting it towards the reactor. (c) The rector explosion causes shrapnel to fly towards the user. (d) The plasma ball and all shrapnel are rendered using the pendulum prop.

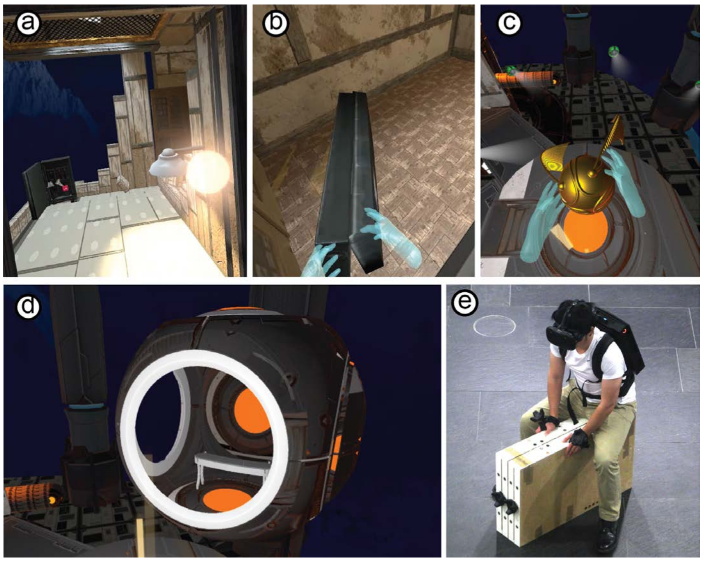

Figure 68: The user places the fuse box to a safe region which later becomes the chair in the escaping pod.

Figure 68 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.

6.2 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.

6.2.1 *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.

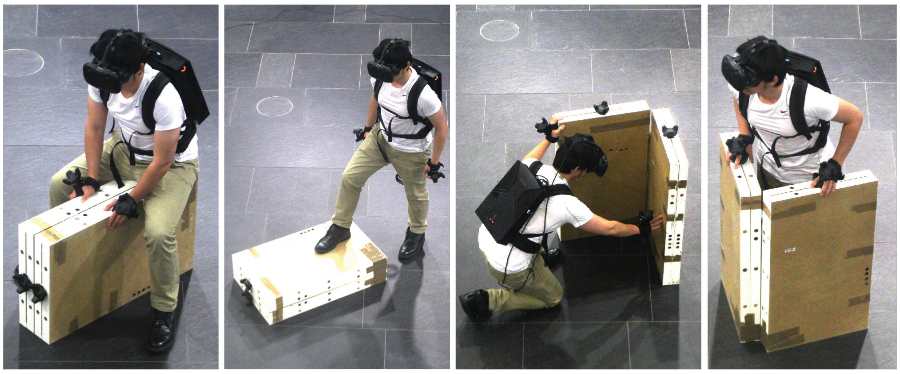

Figure 69: Some of the uses of the foldable board in our demo experience.

In our demo experience, we used a foldable board as prop in order to illustrate the intentional analogy to TurkDeck. 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 69).

(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 [49]. The reason we instead used portals disguised as doors, was primarily to allow our experience to run in a very compact tracking volume (4 m x 4 m).

6.2.2 *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 70a). 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 70, i.e., (b) animate the prop, (c) keep the prop animated, and (d) stop the prop.

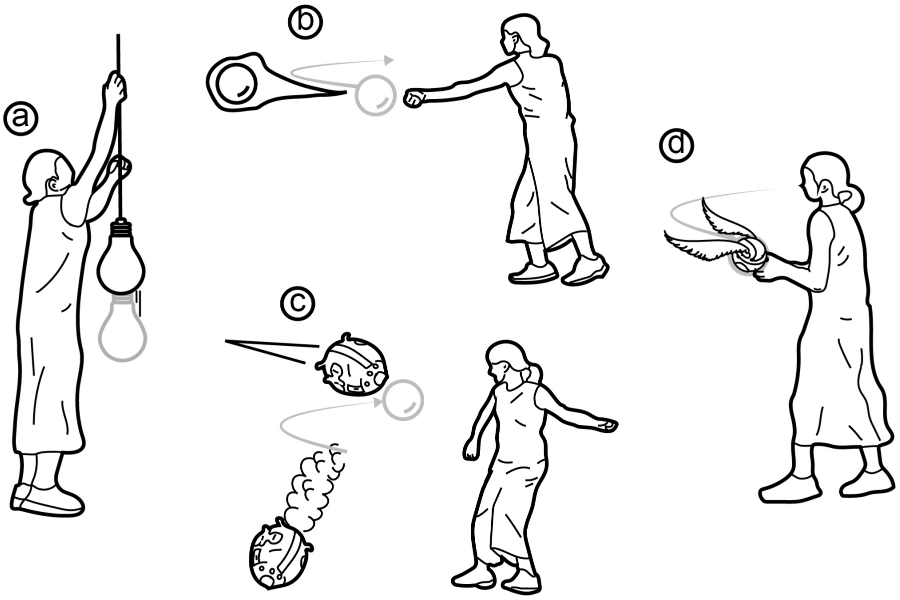

Figure 70: Our demo experience contains examples of (a) inanimate use, (b) animate prop, (c) keep prop animated, and (d) stop the prop.

(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 70c and d) are probably experience designers' best tools for conveying a virtual world that is alive.

6.2.3 *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. 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 61a, we used this to make the plasma ball find its way into the reactor. In Figure 61c, 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.

Figure 71: A football experience by replacing the end effector of the pendulum with a punching bag,

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 61 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.

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 71)

6.3 TRACKING PROPS AND THE PENDULUM EXTRPOLATION

We developed iTurk software system in Unity 3D, running on a Zotac VR go backpack PC. We use an HTC Vive system [10] for tracking as well as 5 Vive trackers for the props and the user's hands.

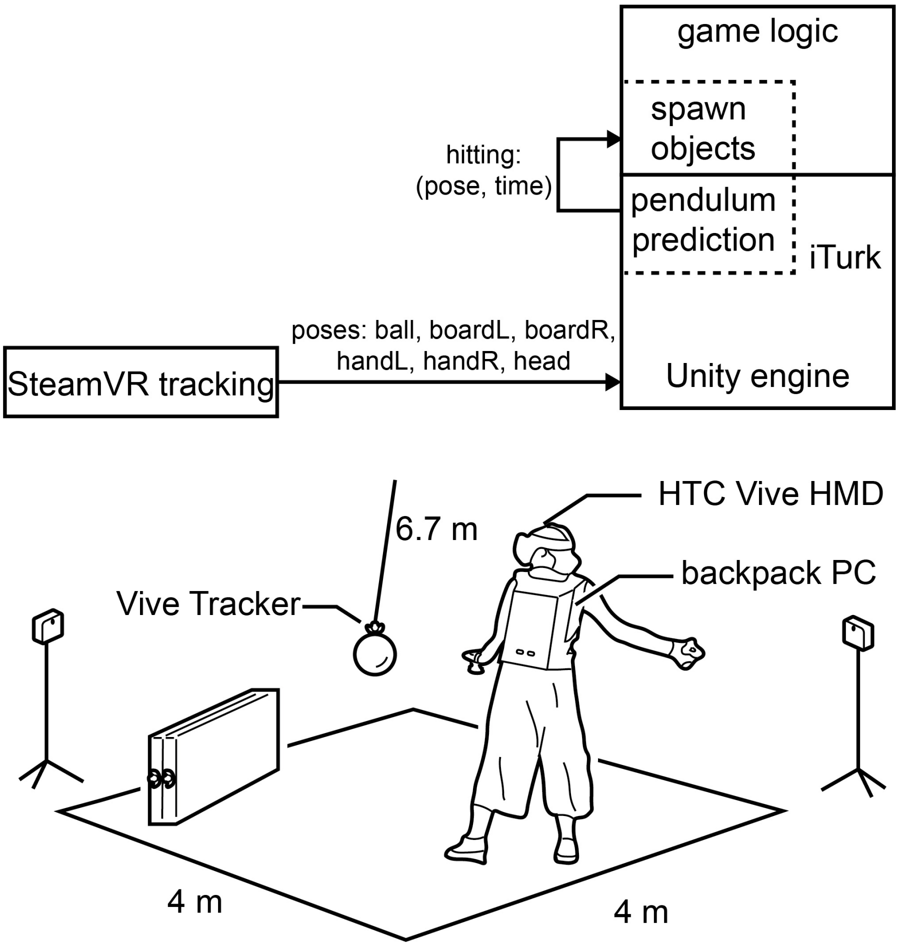

Figure 72: The system diagram of iTurk.

Figure 62 has 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 1 m 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.

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 73, the system uses two user interactions to get a stationary pendulum to orbit.

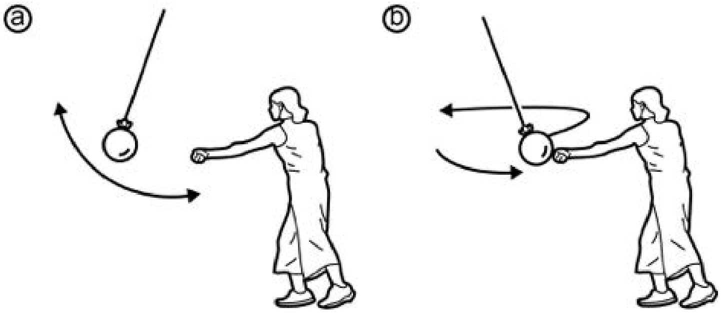

Figure 73: Making the pendulum reach its two degrees of freedom requires two user interactions: (a) The first hit makes the prop swing back and forth and (b) the second hit allows it to orbit.

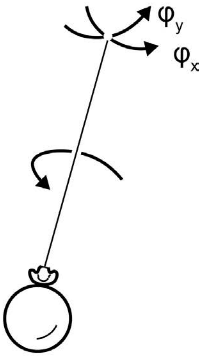

Figure 74: The state of the spherical pendulum is described by rotation angles on the pivot.

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 x and y as shown in Figure 74.

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

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

It then extrapolates using ordinary differential equations solver from the boost library [53] 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 75 shows the hit trajectory.

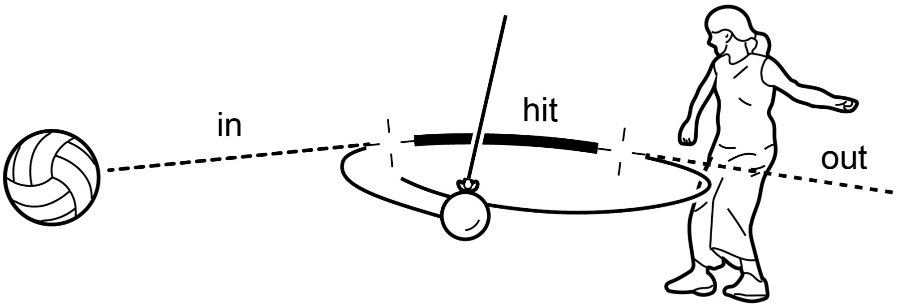

Figure 75: iTurk's pendulum movement behind the scene.

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 75, 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 61. 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 (5 m in 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.

6.4 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 hypothesized that iTurk's reconfigurable props would contribute to participants' experience and that the reconfigure step would not affect participants' sense of realism. We recruited 12 participants from our institute and 3 of them had used HMD before.

Figure 76: A participant hit the pendulum during the study.

TASK 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 counterbalanced order and filled in a questionnaire.

RESULT Figure 77 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.

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.

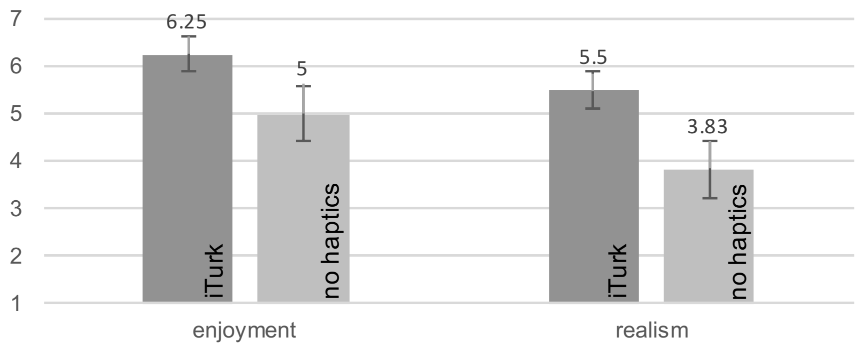

Figure 77: Participants rated their experience in the iTurk condition as significantly more enjoyable and realistic than in the no-haptics control.

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.

6.5 DISCUSSION

The main contribution of iTurk 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.

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.

We have presented iTurk, a single-user human actuation system. With iTurk, we open up a new design space for single-user VR experiences that come with physical feedback. iTurk naturally requires no dedicated human actuator as there is only one user, and it could be see as the minimum version of Mutual Turk.

7

CONCLUSION AND NEXT STEPS

We have explored the concept of human actuation, a new take on large-scale physical feedback that leverages human power instead of machines. Through a series of system implementations and validations, we have demonstrated the strength — humans are generic, flexible and versatile, allowing our systems to provide a range of haptic experiences from motions and forces to tactile sensations. We have dealt with the weakness — humans are slow and inaccurate — by creating software systems that predict, schedule and synchronize human actuators with the user's experience. Finally, we have proposed designs and strategies that bring human actuation to a more applicable level where every one enjoys the immersion, even when there is only one user.

7.1 MACHINES VS. HUMANS

Arguably, the key remaining question is: "shouldn't this all be automated in the future; shouldn't all actuation be performed by machines?" We argue 'no'. The idea of automation is to reduce the cost of a heavily repeated process. For immersive experiences, however, this assumption is simply not justified. Users may repeat an experience once or twice, but certainly not hundreds of times. Since the reason for automation does not apply to immersive experiences, we argue that going "back" to manual labor is the valid approach.

In one extreme case, we think of human actuation as complementary to traditional approaches that have been deployed, for example, by the military. The military use case is characterized by the availability of large spaces, large budgets, and the availability of thousands of users who can be brought to the installation to experience the exact same experience. We see human actuation fits well in the ecosystem resulting from the currently ongoing mass-availability of virtual reality consumer headsets. These are democratizing the field by bringing VR to consumers. The resulting consumer use cases lack all the aspects that characterized the military use case in that space and budgets are limited and travel to an installation just does not seem justified. Consequently, we expect to see many VR installations in the future, all of which serve only very few users. This new ecosystem requires a new take on VR. Where the traditional approaches build on flying in thousands of users, to a military base or Disney World, the new eco system will bring immersive virtual experiences to people.

7.2 MACHINES + HUMANS

While we see human actuation is the valid approach for large-scale physical feedback, there might be a synergy between human and machine actuators. Human actuators are slow but generic. Machine actuators are fast but specific. We may borrow the idea of sensor fusion to combine these two. For example, in a vehicle simulation, it involves not only low frequency motions but also high frequency vibration. The same effect may be achieved by letting human actuators generate the low frequency motions and vibration motors generate the high frequency vibration.

7.3 MORE USERS/HUMAN ACTUATORS

There is still a challenge of orchestrating larger group of human actuators. So far, the maximum number of human actuators that have been used in our systems are ten in TurkDeck, and they were are dedicated for just one user. There are all possibilities of orchestrating n dedicated human actuators and m users. More research questions around this topic would be how to balance users and human actuators and how much performance gain or loss from adding more hum actuators.

BIBLIOGRAPHY

[1] Laurent Aguerreche, Thierry Duval, and Anatole Lécuyer. "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. Hong Kong: ACM, 2010, pp. 227–230. ISBN: 978-1-4503-0441-2. DOI: 10.1145/1889863.1889913. URL: http://doi.acm.org/10.1145/1889863.1889913.

[2] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. "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. San Jose, California, USA: ACM, 2016, pp. 1968–1979. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858226. URL: http://doi.acm.org/10.1145/2858036.2858226.

[3] Akash Badshah, Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. "GyroTab: A Handheld Device That Provides Reactive Torque Feedback." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 3153–3156. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2208731. URL: http://doi.acm.org/10.1145/2207676.2208731.

[4] Patrick Baudisch, Henning Pohl, Stefanie Reinicke, Emilia Wittmers, Patrick Lühne, Marius Knaust, Sven Köhler, Patrick Schmidt, and Christian Holz. "Imaginary Reality Gaming: Ball Games Without a Ball." In: *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. UIST '13. St. Andrews, Scotland, United Kingdom: ACM, 2013, pp. 405–410. ISBN: 978-1-4503-2268-3. DOI: 10.1145/2501988.2502012. URL: http://doi.acm.org/10.1145/2501988.2502012.

[5] Massimo Bergamasco. "The GLAD-IN-ART Project." In: *Virtual Reality*. Ed. by H. J. Warnecke and H.-J. Bullinger. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 251–258. ISBN: 978-3-642-88650-8.

[6] A. Berthoz. *The Brain's Sense of Movement*. Perspectives in cognitive neuroscience. Harvard University Press, 2000. ISBN: 9780674971097. URL: https://books.google.de/books?id=4VlEAQAACAAJ.

[7] Frederick P. Brooks Jr., Ming Ouh-Young, James J. Batter, and P. Jerome Kilpatrick. "Project GROPEHaptic Displays for Scientific Visualization." In: *Proceedings of the 17th Annual Conference*

on Computer Graphics and Interactive Techniques. SIGGRAPH '90. Dallas, TX, USA: ACM, 1990, pp. 177–185. ISBN: 0-89791-344-2. DOI: 10.1145/97879.97899. URL: http://doi.acm.org/10.1145/97879.97899.

[8] Lung-Pan Cheng, Patrick Luehne, Pedro Lopes, Christoph Sterz, and Patrick Baudisch. "Haptic Turk: A Motion Platform Based on People." In: *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: ACM, 2014, pp. 3463–3472. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557101. URL: http://doi.acm.org/10.1145/2556288.2557101.

[9] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D. Wilson. "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. Denver, Colorado, USA: ACM, 2017, pp. 3718–3728. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025753. URL: http://doi.acm.org/10.1145/3025453.3025753.

[10] HTC Corporation. *Vive*. URL: https://www.vive.com (visited on 05/30/2018).

[11] Nvidia Corporation. *GeForce*. URL: https://www.nvidia.com/en-in/geforce (visited on 05/30/2018).

[12] Carolina Cruz-Neira, Daniel J. Sandin, Thomas A. DeFanti, Robert V. Kenyon, and John C. Hart. "The CAVE: Audio Visual Experience Automatic Virtual Environment." In: *Commun. ACM* 35.6 (June 1992), pp. 64–72. ISSN: 0001-0782. DOI: 10.1145/129888.129892. URL: http://doi.acm.org/10.1145/129888.129892.

[13] Laycock S. D. and Day A. M. "Recent Developments and Applications of Haptic Devices." In: *Computer Graphics Forum* 22.2 (2003), pp. 117–132. DOI: 10.1111/1467-8659.00654. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8659.00654. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8659.00654.

[14] Fabien Danieau, Julien Fleureau, Philippe Guillotel, Nicolas Mollet, Marc Christie, and Anatole Lécuyer. "HapSeat: A Novel Approach to Simulate Motion in Audiovisual Experiences." In: *ACM SIGGRAPH 2013 Emerging Technologies*. SIGGRAPH '13. Anaheim, California: ACM, 2013, 6:1–6:1. ISBN: 978-1-4503-2340-6. DOI: 10.1145/2503368.2503374. URL: http://doi.acm.org/10.1145/2503368.2503374.

[15] VTT Technical Research Centre of Finland Ltd. *ALVAR*. URL: http://alvar.erve.vtt.fi/ (visited on 05/30/2018).

[16] S. H. Foster, E. M. Wenzel, and R. M. Tayior. "Real Time Synthesis of Complex Acoustic Environments." In: *Final Program and Paper Summaries 1991 IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*. Oct. 1991, pp. 47–48. DOI: 10.1109/ASPAA.1991.634098.

[17] Raymond C. Goertz and William M. Thompson. "Electronically controlled manipulator." In: *Nucleonics* 12.11 (), pp. 46–47.

[18] B. K. Han, S. C. Kim, S. C. Lim, D. Pyo, and D. S. Kwon. "Physical mobile interaction with kinesthetic feedback." In: *2012 IEEE Haptics Symposium (HAPTICS)*. Mar. 2012, pp. 571–575. DOI: 10.1109/HAPTIC.2012.6183849.

[19] L. R. Harris, M. R. Jenkin, D. Zikovitz, F. Redlick, P. Jaekl, U. T. Jasiobedzka, H. L. Jenkin, and R. S. Allison. "Simulating Self-Motion I: Cues for the Perception of Motion." In: *Virtual Reality* 6.2 (Sept. 2002), pp. 75–85. ISSN: 1434-9957. DOI: 10.1007/s100550200008. URL: https://doi.org/10.1007/s100550200008.

[20] T. Hayashi, H. Kawamoto, and Y. Sankai. "Control method of robot suit HAL working as operator's muscle using biological and dynamical information." In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Aug. 2005, pp. 3063–3068. DOI: 10.1109/IROS.2005.1545505.

[21] Anuruddha Hettiarachchi and Daniel Wigdor. "Annexing Reality: Enabling Opportunistic Use of Everyday Objects As Tangible Proxies in Augmented Reality." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: ACM, 2016, pp. 1957–1967. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858134. URL: http://doi.acm.org/10.1145/2858036.2858134.

[22] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. "Passive Real-world Interface Props for Neurosurgical Visualization." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '94. Boston, Massachusetts, USA: ACM, 1994, pp. 452–458. ISBN: 0-89791-650-6. DOI: 10.1145/191666.191821. URL: http://doi.acm.org/10.1145/191666.191821.

[23] H. G. Hoffman. "Physically Touching Virtual Objects Using Tactile Augmentation Enhances the Realism of Virtual Environments." In: *Proceedings of the Virtual Reality Annual International Symposium*. VRAIS '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 59–. ISBN: 0-8186-8362-7. URL: http://dl.acm.org/citation.cfm?id=522258.836098.

[24] John M. Hollerbach, Rose Mills, David Tristano, Robert R. Christensen, William B. Thompson, and Yangming Xu. "Torso Force Feedback Realistically Simulates Slope on Treadmill-Style Lo-

comotion Interfaces." In: *The International Journal of Robotics Research* 20.12 (2001), pp. 939–952. DOI: 10.1177/02783640122068209. eprint: https://doi.org/10.1177/02783640122068209. URL: https://doi.org/10.1177/02783640122068209.

[25] Charles E. Hughes, Christopher B. Stapleton, Darin E. Hughes, and Eileen M. Smith. "Mixed Reality in Education, Entertainment, and Training." In: *IEEE Comput. Graph. Appl.* 25.6 (Nov. 2005), pp. 24–30. ISSN: 0272-1716. DOI: 10.1109/MCG.2005.139. URL: https://doi.org/10.1109/MCG.2005.139.

[26] Brent Edward Insko. "Passive Haptics Significantly Enhances Virtual Environments." AAI3007820. PhD thesis. 2001. ISBN: 0-493-17286-6.

[27] Ali Israr and Ivan Poupyrev. "Tactile Brush: Drawing on Skin with a Tactile Grid Display." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: ACM, 2011, pp. 2019–2028. ISBN: 978-1-4503-0228-9. DOI: 10.1145/1978942.1979235. URL: http://doi.acm.org/10.1145/1978942.1979235.

[28] H. Iwata, H. Yano, and F. Nakaizumi. "Gait Master: a versatile locomotion interface for uneven virtual terrain." In: *Proceedings IEEE Virtual Reality 2001*. Mar. 2001, pp. 131–137. DOI: 10.1109/VR.2001.913779.

[29] Hiroo Iwata, Hiroaki Yano, Hiroyuki Fukushima, and Haruo Noma. "CirculaFloor: A Locomotion Interface Using Circulation of Movable Tiles." In: *Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*. VR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 223–230. ISBN: 0-7803-8929-8. DOI: 10.1109/VR.2005.11. URL: https://doi.org/10.1109/VR.2005.11.

[30] Hiroo Iwata, Hiroaki Yano, Fumitaka Nakaizumi, and Ryo Kawamura. "Project FEELEX: Adding Haptic Surface to Graphics." In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 469–476. ISBN: 1-58113-374-X. DOI: 10.1145/383259.383314. URL: http://doi.acm.org/10.1145/383259.383314.

[31] Caroline Jay, Mashhuda Glencross, and Roger Hubbold. "Modeling the Effects of Delayed Haptic and Visual Feedback in a Collaborative Virtual Environment." In: *ACM Trans. Comput.-Hum. Interact.* 14.2 (Aug. 2007). ISSN: 1073-0516. DOI: 10.1145/1275511.1275514. URL: http://doi.acm.org/10.1145/1275511.1275514.

[32] L. Kohli. “Redirected touching: Warping space to remap passive haptics.” In: *2010 IEEE Symposium on 3D User Interfaces (3DUI)*. Mar. 2010, pp. 129–130. DOI: 10.1109/3DUI.2010.5444703.

[33] Luv Kohli, Eric Burns, Dorian Miller, and Henry Fuchs. “Combining Passive Haptics with Redirected Walking.” In: *Proceedings of the 2005 International Conference on Augmented Tele-existence*. ICAT ’05. Christchurch, New Zealand: ACM, 2005, pp. 253–254. ISBN: 0-473-10657-4. DOI: 10.1145/1152399.1152451. URL: http://doi.acm.org/10.1145/1152399.1152451.

[34] Robert J. Kosinski. “A Literature Review on Reaction Time.” In: 2012.

[35] Sooyong Lee, Sangmin Park, Munsang Kim, and Chong-Won Lee. “Design of a force reflecting master arm and master hand using pneumatic actuators.” In: *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*. Vol. 3. May 1998, 2574–2579 vol.3. DOI: 10.1109/ROBOT.1998.680729.

[36] Gerald M. Levitt. *The Turk, Chess Automation*. McFarland & Company, Inc. Publishers, 2000. ISBN: 0786407786.

[37] Debra A. Lieberman. *Dance Games and Other Exergames: What the Research Says*. 2006.

[38] R. W. Lindeman, J. L. Sibert, and J. K. Hahn. “Hand-held windows: towards effective 2D interaction in immersive virtual environments.” In: *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*. Mar. 1999, pp. 205–212. DOI: 10.1109/VR.1999.756952.

[39] Pedro Lopes, Alexandra Ion, and Patrick Baudisch. “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. Charlotte, NC, USA: ACM, 2015, pp. 11–19. ISBN: 978-1-4503-3779-3. DOI: 10.1145/2807442.2807443. URL: http://doi.acm.org/10.1145/2807442.2807443.

[40] Pedro Lopes, Sijing You, Lung-Pan Cheng, Sebastian Marwecki, and Patrick Baudisch. “Providing Haptics to Walls and 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. Denver, Colorado, USA: ACM, 2017, pp. 1471–1482. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025600. URL: http://doi.acm.org/10.1145/3025453.3025600.

[41] Kok-Lim Low, Greg Welch, Anselmo Lastra, and Henry Fuchs. “Life-sized Projector-based Dioramas.” In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. VRST ’01.

Baniff, Alberta, Canada: ACM, 2001, pp. 93–101. ISBN: 1-58113-427-4. DOI: 10.1145/505008.505026. URL: http://doi.acm.org/10.1145/505008.505026.

[42] Vicon Motion Systems Ltd. *Vicon*. URL: https://www.vicon.com (visited on 05/30/2018).

[43] John C. McClelland, Robert J. Teather, and Audrey Girouard. "Haptobend: Shape-changing Passive Haptic Feedback in Virtual Reality." In: *Proceedings of the 5th Symposium on Spatial User Interaction*. SUI '17. Brighton, United Kingdom: ACM, 2017, pp. 82–90. ISBN: 978-1-4503-5486-8. DOI: 10.1145/3131277.3132179. URL: http://doi.acm.org/10.1145/3131277.3132179.

[44] W. A. McNeely. "Robotic graphics: a new approach to force feedback for virtual reality." In: *Proceedings of IEEE Virtual Reality Annual International Symposium*. Sept. 1993, pp. 336–341. DOI: 10.1109/VRAIS.1993.380761.

[45] H. Noma, T. Sugihara, and T. Miyasato. "Development of Ground Surface Simulator for Tel-E-Merge system." In: *Proceedings IEEE Virtual Reality 2000 (Cat. No.00CB37048)*. 2000, pp. 217–224. DOI: 10.1109/VR.2000.840501.

[46] Michael Ortega and Sabine Coquillart. "Prop-Based Haptic Interaction with Co-location and Immersion: an Automotive Application." In: *CoRR* abs/cs/0601025 (2006). arXiv: cs/0601025. URL: http://arxiv.org/abs/cs/0601025.

[47] Jarrell Pair, Ulrich Neumann, Diane Piepol, and Bill Swartout. "FlatWorld: Combining Hollywood Set-Design Techniques with VR." In: *IEEE Comput. Graph. Appl.* 23.1 (Jan. 2003), pp. 12–15. ISSN: 0272-1716. DOI: 10.1109/MCG.2003.1159607. URL: http://dx.doi.org/10.1109/MCG.2003.1159607.

[48] burning man project. *Burning Man*. URL: https://burningman.org (visited on 05/30/2018).

[49] Sharif Razzaque. "Redirected Walking." AAI3190299. PhD thesis. Chapel Hill, NC, USA, 2005. ISBN: 0-542-34053-4.

[50] Wolfgang Reisig. "Petri Nets and Algebraic Specifications." In: *Theor. Comput. Sci.* 80.1 (Mar. 1991), pp. 1–34. ISSN: 0304-3975. DOI: 10.1016/0304-3975(91)90203-E. URL: http://dx.doi.org/10.1016/0304-3975(91)90203-E.

[51] Bernhard E. Riecke, Jorg Schulte-Pelkum, Franck Caniard, and Heinrich H. Bulthoff. "Towards Lean and Elegant Self-Motion Simulation in Virtual Reality." In: *Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*. VR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 131–138. ISBN: 0-7803-8929-8. DOI: 10.1109/VR.2005.83. URL: https://doi.org/10.1109/VR.2005.83.

[52] Andreas Schilling. "Towards Real-time Photorealistic Rendering: Challenges and Solutions." In: *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware*. HWWS '97. Los Angeles, California, USA: ACM, 1997, pp. 7–15. ISBN: 0-89791-961-0. DOI: 10.1145/258694.258701. URL: http://doi.acm.org/10.1145/258694.258701.

[53] Boris Schling. *The Boost C++ Libraries*. XML Press, 2011. ISBN: 0982219199, 9780982219195.

[54] Dominik Schmidt, Rob Kovacs, Vikram Mehta, Udayan Umapathi, Sven Koehler, Lung-Pan Cheng, and Patrick Baudisch. "Level-Ups: Motorized Stilts That Simulate Stair Steps in Virtual Reality." In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: ACM, 2015, pp. 2157–2160. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702253. URL: http://doi.acm.org/10.1145/2702123.2702253.

[55] Jia Sheng, Ravin Balakrishnan, and Karan Singh. "An Interface for Virtual 3D Sculpting via Physical Proxy." In: *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*. GRAPHITE '06. Kuala Lumpur, Malaysia: ACM, 2006, pp. 213–220. ISBN: 1-59593-564-9. DOI: 10.1145/1174429.1174467. URL: http://doi.acm.org/10.1145/1174429.1174467.

[56] Inc Silicon Graphics. *Onyx 3000*.

[57] Adalberto L. Simeone, Eduardo Velloso, and Hans Gellersen. "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. Seoul, Republic of Korea: ACM, 2015, pp. 3307–3316. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702389. URL: http://doi.acm.org/10.1145/2702123.2702389.

[58] Rajinder Sodhi, Ivan Poupyrev, Matthew Glisson, and Ali Israr. "AIREAL: Interactive Tactile Experiences in Free Air." In: *ACM Trans. Graph.* 32.4 (July 2013), 134:1–134:10. ISSN: 0730-0301. DOI: 10.1145/2461912.2462007. URL: http://doi.acm.org/10.1145/2461912.2462007.

[59] Jonas Spillmann, Stefan Tuchschmid, and Matthias Harders. "Adaptive Space Warping to Enhance Passive Haptics in an Arthroscopy Surgical Simulator." In: *IEEE Transactions on Visualization and Computer Graphics* 19.4 (Apr. 2013), pp. 626–633. ISSN: 1077-2626. DOI: 10.1109/TVCG.2013.23. URL: https://doi.org/10.1109/TVCG.2013.23.

[60] Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz, and Markus Lappe. “Estimation of Detection Thresholds for Redirected Walking Techniques.” In: *IEEE Transactions on Visualization and Computer Graphics* 16.1 (Jan. 2010), pp. 17–27. ISSN: 1077-2626. DOI: `10.1109/TVCG.2009.62`. URL: `http://dx.doi.org/10.1109/TVCG.2009.62`.

[61] D. Stewart. “A Platform with Six Degrees of Freedom.” In: *Proceedings of the Institution of Mechanical Engineers* 180.1 (1965), pp. 371–386. DOI: `10.1243/PIME\_PROC\_1965\_180\_029\_02`. eprint: `https://doi.org/10.1243/PIME_PROC_1965_180_029_02`. URL: `https://doi.org/10.1243/PIME_PROC_1965_180_029_02`.

[62] Evan A. Suma, Zachary Lipps, Samantha Finkelstein, David M. Krum, and Mark Bolas. “Impossible Spaces: Maximizing Natural Walking in Virtual Environments with Self-Overlapping Architecture.” In: *IEEE Transactions on Visualization and Computer Graphics* 18.4 (Apr. 2012), pp. 555–564. ISSN: 1077-2626. DOI: `10.1109/TVCG.2012.47`. URL: `http://dx.doi.org/10.1109/TVCG.2012.47`.

[63] Ivan E. Sutherland. “The Ultimate Display.” In: *Proceedings of the IFIP Congress*. 1965, pp. 506–508.

[64] Ivan E. Sutherland. “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). San Francisco, California: ACM, 1968, pp. 757–764. DOI: `10.1145/1476589.1476686`. URL: `http://doi.acm.org/10.1145/1476589.1476686`.

[65] Zsolt Szalavari and Michael Gervautz. “The Personal Interaction Panel - a Two-Handed Interface for Augmented Reality.” In: 16 (Sept. 1997), pp. 335–346.

[66] Unity Technology. *Unity*. URL: `https://www.unity3d.com` (visited on 05/30/2018).

[67] Dzmitry Tsetserukou, Katsunari Sato, and Susumu Tachi. “FlexTorque: Exoskeleton Interface for Haptic Interaction with the Digital World.” In: *Proceedings of the 2010 International Conference on Haptics - Generating and Perceiving Tangible Sensations: Part II*. EuroHaptics’10. Amsterdam: Springer-Verlag, 2010, pp. 166–171. ISBN: 3-642-14074-2, 978-3-642-14074-7. URL: `http://dl.acm.org/citation.cfm?id=1893760.1893786`.

[68] Martin Usoh, Kevin Arthur, Mary C. Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P. Brooks Jr. “Walking > Walking-in-place > Flying, in Virtual Environments.” In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 359–364. ISBN:

0-201-48560-5. DOI: 10.1145/311535.311589. URL: http://dx.doi.org/10.1145/311535.311589.

[69] Oculus VR. *Oculus Rift DK1*. URL: https://www.oculus.com (visited on 05/30/2018).

[70] Bob G. Witmer and Michael J. Singer. "Measuring Presence in Virtual Environments: A Presence Questionnaire." In: *Presence: Teleoper. Virtual Environ.* 7.3 (June 1998), pp. 225–240. ISSN: 1054-7460. DOI: 10.1162/105474698565686. URL: http://dx.doi.org/10.1162/105474698565686.

[71] H. Yano, Y. Miyamoto, and H. Iwata. "Haptic interface for perceiving remote object using a laser range finder." In: *World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. Mar. 2009, pp. 196–201. DOI: 10.1109/WHC.2009.4810889.

[72] YouTube. *YouTube*. URL: https://www.youtube.com (visited on 05/30/2018).

[73] Andre Zenner and Antonio Kruger. "Shifty: A Weight-Shifting Dynamic Passive Haptic Proxy to Enhance Object Perception in Virtual Reality." In: *IEEE Transactions on Visualization and Computer Graphics* 23.4 (Apr. 2017), pp. 1285–1294. ISSN: 1077-2626. DOI: 10.1109/TVCG.2017.2656978. URL: https://doi.org/10.1109/TVCG.2017.2656978.

DECLARATION

Ich erkläre hiermit, dass

- ich die vorliegende Dissertationsschrift "Human Actuation" selbständig und ohne unerlaubte Hilfe angefertigt sowie nur die angegebene Literatur verwendet habe,
- ich mich nicht bereits anderwärts um einen Doktorgrad beworben habe oder ich einen solchen besitze und
- mir die Promotionsordnung der Mathematisch-Naturwissenschaftlichen Fakultät der Universität Potsdam vom 18. September 2013 bekannt ist.

Potsdam, Germany, June 2018

Lung-Pan Cheng

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

`https://bitbucket.org/amiede/classicthesis/`

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

`http://postcards.miede.de/`

Final Version as of June 15, 2018 (`classicthesis` version 1.0).