Haptic Revolver: Touch, Shear, Texture, and Shape Rendering on a Reconfigurable Virtual Reality Controller

Eric Whitmire[1], **Hrvoje Benko**[2], **Christian Holz**[2], **Eyal Ofek**[2], **Mike Sinclair**[2]
[1]Paul G. Allen School, DUB Group, University of Washington, Seattle, WA
[2]Microsoft Research, Redmond, WA
emwhit@cs.washington.edu, {benko, cholz, eyalofek, sinclair}@microsoft.com

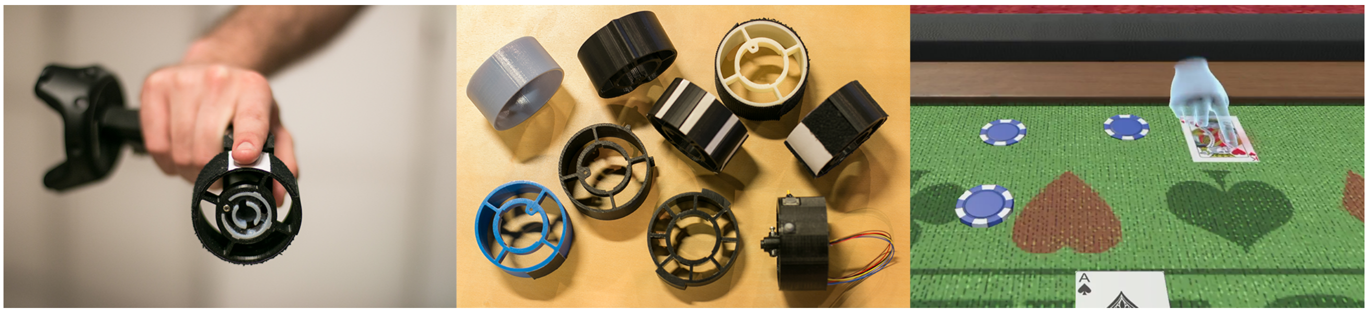

Figure 1. (left) Our Haptic Revolver device uses a wheel that raises and lowers and spins underneath the fingertip to render various haptic sensations. (center) The haptic wheels are interchangeable and can be customized to render arbitrary textures, shapes, or interactive elements. (right) Wheel features are spatially registered with the virtual environment, so the user can reach out and feel virtual surfaces.

ABSTRACT
We present Haptic Revolver, a handheld virtual reality controller that renders fingertip haptics when interacting with virtual surfaces. Haptic Revolver's core haptic element is an actuated wheel that raises and lowers underneath the finger to render contact with a virtual surface. As the user's finger moves along the surface of an object, the controller spins the wheel to render shear forces and motion under the fingertip. The wheel is interchangeable and can contain physical textures, shapes, edges, or active elements to provide different sensations to the user. Because the controller is spatially tracked, these physical features can be spatially registered with the geometry of the virtual environment and rendered on-demand. We evaluated Haptic Revolver in two studies to understand how wheel speed and direction impact perceived realism. We also report qualitative feedback from users who explored three application scenarios with our controller.

ACM Classification Keywords
H.5.1. Information Interfaces and Presentation: Multimedia Information Systems-Artificial, Augmented, and Virtual Realities; H.5.2 User Interfaces: Haptic I/O

Author Keywords
Virtual Reality; Haptics

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

ISBN 978-1-4503-5620-6/18/04...$15.00
DOI: https://doi.org/10.1145/3173574.3173660

INTRODUCTION
Recent advances in display technologies, computer graphics, and tracking have led to a resurgence in head mounted displays for virtual reality. Today's consumer VR devices are capable of rendering realistic visual and audio content and positionally tracked handheld controllers further improve the sense of presence by bringing the user's hands into the virtual world. Despite these advances, the ability of such devices to render the sense of touch is lacking. Haptics on handheld controllers are limited to vibrotactile stimulation, which is typically used for notification or binary touch events. This lack of cutaneous cues limits the user's ability to feel contact with a surface and to explore its texture and shape.

For more nuanced haptic rendering, researchers have developed finger-mounted haptic devices [16, 21, 17, 6, 15, 29, 26, 20], glove-based exoskeletons [19, 9, 23, 4], and robotic arm solutions [1, 14, 13, 10, 12, 24] to render various haptic sensations. These devices either require users to mount or wear additional devices or require expensive robotic arms with a limited range. Researchers have also explored the use of handheld devices for haptic rendering [3, 31, 18]. Such devices are convenient to use and they are likely more compatible with existing VR systems because they can replace the functionality of existing controllers. However, previous controller-based devices have only focused on rendering a single haptic stimulus (e.g. normal forces or weight distribution).

In this paper, we present Haptic Revolver, a reconfigurable handheld haptic controller for virtual reality. The device uses an actuated wheel underneath the fingertip that moves up and down to render touch contact with a virtual surface and spins to render shear forces and motion as the user slides along a virtual surface. The device's wheel is interchangeable and it can contain a variety of physical haptic elements, such as ridges, textures, or custom shapes (Figure 1). These haptic features on the wheel's outer surface provide different sensations to the user as they explore the virtual environment. Because the device is spatially tracked, these haptic elements are spatially registered with the virtual environment. As the user explores a virtual environment, our rendering engine delivers the appropriate haptic element underneath the finger. For example, in a virtual card game environment, when a user touches a card, a poker chip, and a table, the device rotates the wheel to render the appropriate texture underneath the fingertip. As the user slides along one of these surfaces, the wheel moves underneath the finger to render shear forces and motion.

Unlike other haptic devices [3, 6, 23], which always maintain contact with the finger, our Haptic Revolver device can selectively contact the finger. When a user touches a virtual surface, the wheel rises to contact the fingertip. Because the haptic wheels on our device are interchangeable, Haptic Revolver can generalize to many applications. Applications can use custom wheels with the necessary haptic features. For example, a virtual petting zoo game might use a wheel containing various textures while a virtual cockpit environment might use a wheel with input elements such as buttons and switches.

In the following sections, we describe the design and implementation of our device, the techniques we use to render an arbitrary scene, and results from two perceptual studies that informed these decisions. Our results show that we can change the wheel speed and direction to render arbitrary scenes without compromising realism and support our technique of rendering 2D motion with a single wheel. We also show several example applications that highlight functionality of our device and qualitative feedback from users.

Specifically, our contributions include:

1. The design of Haptic Revolver, a handheld VR controller that renders touch contact, pressure, shear forces, textures, and shapes using a rotating wheel beneath the index finger;
2. Interchangeable haptic wheels that can be used to render surface features and techniques to haptically render any scene using an arbitrary wheel;
3. The results of two perceptual user studies that inform the design of our haptic rendering strategies.

By combining the fundamentals of touch contact, pressure, and shear rendering with the flexibility of haptic wheels that support arbitrary shapes and textures, Haptic Revolver enables more accurate haptic rendering for virtual environments.

RELATED WORK

While there is extensive literature on the field of haptics, we restrict our review of related work to haptic VR controllers, wearable haptic devices, and desktop haptic devices.

Haptic VR Controllers

The small form factor, low cost, and low power of vibrotactile actuators have led to their dominance in commercial VR controllers. The positionally tracked controllers offered by consumer VR systems (e.g. Oculus Touch or HTC VIVE controllers) include customizable vibrotactile feedback. However, the amount of information that can be conveyed by vibrotactile stimulation is limited and its usage is typically limited to simple touch events or notifications. Some research efforts have investigated how to use vibrotactile stimulation to render surface textures with a particular focus on how stimulation parameters impact users' perception of a surface [8, 22].

Recent academic and commercial efforts have attempted to move beyond vibrotactile feedback. For example, Benko et al. demonstrated NormalTouch and TextureTouch, controllers that render normal forces on the fingertip [3]. Like these devices, Haptic Revolver also targets haptic sensations at the fingertip, but we focus on shear forces and texture rendering instead of normal forces. Unlike NormalTouch and TextureTouch, which always maintain contact with the fingertip and modulate force to simulate touch, the movable wheel in Haptic Revolver is able to fully retract whenever there is no touch contact.

Other efforts have focused on using controllers to render the sensation of holding an object. Zenner and Krüger presented Shifty, a handheld device that shifts its weight distribution to simulate holding objects of different weights [31]. Tactical Haptics designed a controller that simulates friction forces in the palm due to holding an object using sliding tactors in the device handle [18]. By moving these tactors on opposite directions, the controller can simulate torsional forces as well.

Wearable Haptic Devices

In addition to handheld haptic controllers, there are several options for wearable haptic devices. Glove-based exoskeletons such as the Exos [9], Dexmo [19], Cyber Grasp [23], and Rutgers Master II [4] all use actuators at the fingers to resist grasping forces. Though these devices have the advantage of grounding at the wrist, they tend to be bulky and require a nontrivial amount of setup compared to a handheld controller. Such devices are also unable to render shear forces and motion underneath the fingertip. In contrast, our device can render the sensation of sliding across a virtual surface.

Finger-mounted haptic devices are another common form factor. These are often strapped or clipped onto one or more fingers to provide fingertip sensations as the hand moves. In recent years, researchers have explored rendering contact [21], pressure [17], tilt [6], and shear forces [15, 20] with these devices. Yem and Kajimoto developed the FinGAR device, which uses a DC motor and electrodes to provide skin deformation and vibration [29]. Wolverine is a four-finger device that uses braking to simulate grasping rigid objects [7]. Go Touch VR designed a device that clips onto three fingers and renders contact and pressure as a user grasps a virtual object [26]. Like the exoskeleton devices, these devices require careful mounting to the fingers before use. Our Haptic Revolver device renders many of the same sensations as these devices as well as motion under the fingertip. Moreover, with our device, one can simply pick it up and begin using it without having to strap anything to the fingers. However, unlike most controllers, including our device, these finger-mounted devices do not restrict hand posture during use. We note that there are many more examples of finger-mounted haptic devices. For a more comprehensive review of such devices, we direct the reader to Pacchierotti et al. [16].

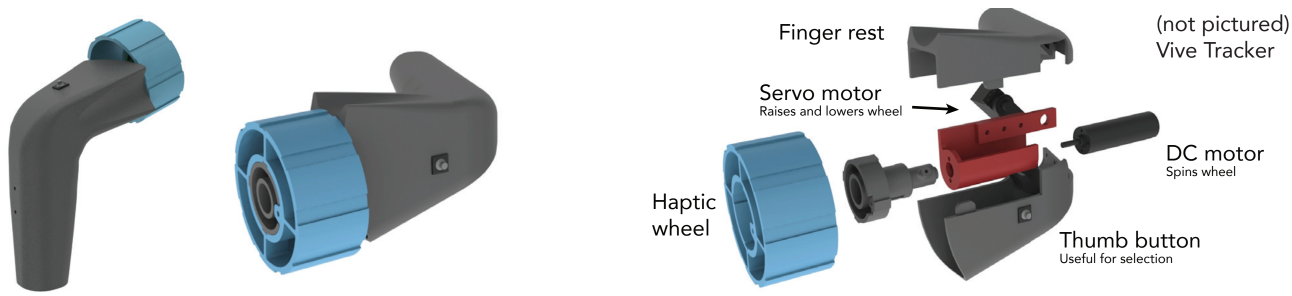

Figure 2. (left) 3D model of Haptic Revolver with a textured wheel attached. (right) Exploded view showing internal components. A servo motor raises and lowers the wheel, while a DC motor spins the wheel to render motion and shear forces. A button on the side allows the user to select objects and navigate. The VIVE tracker (not shown) enables 6-DOF spatial tracking of our device

Desktop Haptic Devices

In addition to handheld and wearable devices, there are a number of efforts exploring haptics using environmentally grounded systems. Robotic arm actuators such as the PHANToM [13], Haptic Master [24], Novint Falcon [14], and Haption Virtuose [10] excel at rendering larger, externally grounded forces against the finger or hand. These are often used in applications such as tele-operated surgery, 3D sculpting, gaming, and interactive training. Recently Araujo et al.'s Snake Charmer [1] used a robotic arm with custom attachments to render various surface features on demand. Like Snake Charmer, our Haptic Revolver controller uses interchangeable attachments to render textures, shapes, and active elements. However, because Haptic Revolver is integrated into a handheld controller, we remove the need for precise finger tracking and have no range limitations. We also use the wheel to render shear forces and motion under the fingertip. The Touch Thimble is another example of combining attachments with a robotic arm for haptic feedback [12]. In this work, a spring loaded thimble keeps the touch surface suspended from the fingertip until a virtual surface is contacted. Though our Haptic Revolver device does not provide the kinesthetic feedback that these robotic arms can offer, we do provide the sensation of making and breaking contact with real objects.

Azmandian et al. showed how retargeting can be used with passive proxies [2] to reuse the same proxy for multiple virtual objects. While Haptic Revolver also uses physical proxies on the wheel, our actuation removes the need for retargeting.

Other desktop devices render fingertip sensations during stationary use [28, 25, 27, 11] or in a limited range [5]. For example, the Plank is a desktop haptic device that uses a spinning wheel to render friction and various terrain shapes [25]. Other haptic devices use the tilt of a platform under the finger to convey surface information [27, 11]. Unlike these devices, Haptic Revolver is a handheld device that renders multiple haptic sensations at the fingertip.

HAPTIC REVOLVER IMPLEMENTATION

To render contact and motion on the fingertip in a compact form factor, we chose to use a wheel that raises and lowers and rotates in response to its position in the virtual environment. In the following sections, we describe the hardware and software of our system as well as the design of interchangeable haptic wheels to deliver custom haptic sensations.

Mechanical Design

We arrived at the design of Haptic Revolver through an iterative process. Each design, shown in Figure 3, improved the functionality and ergonomics of the device. Our final design, shown in Figure 2 has two degrees of freedom, each of which are actuated by a motor. A servo motor (Hitec HS-5070MH) raises and lowers the wheel assembly along an axis positioned along the grip of the controller. The wheel assembly is positioned along the axis of the index finger and consists of a 12 V DC motor (Faulhaber 1524_SR) housed in a 3D printed mount. The motor includes a 19:1 gearhead and a 4096 count 2-channel magnetic encoder. A wheel mount on the end of the wheel assembly allows custom wheels to be easily attached. With this gear ratio, the motor can spin at 180 rpm, which corresponds to a linear motion underneath the finger of 565 mm/s, assuming a 60 mm wheel diameter.

The controller is designed so that the index finger rests in a groove along the wheel axis. This lets the finger naturally rest on the surface of the wheel while preventing horizontal motion as the wheel spins. For improved ergonomics, the axis of the finger wheel assembly is offset from the grip handle by 110°.

A tactile button on the side of the device is used for selection and navigation and is in easy reach of the thumb during normal operation. If desired, there is room on the side of the controller for additional input elements, such as buttons, a joystick, or a touchpad. For positional tracking we attach the HTC VIVE Tracker to the end of the device, as shown in Figure 1 (left). This device integrates easily with the HTC VIVE head mounted display and Lighthouse tracking system and reports a 6-DOF position and orientation.

Figure 3. Examples of previous iterations of our device. This iterative design process led to important design features of our current device.

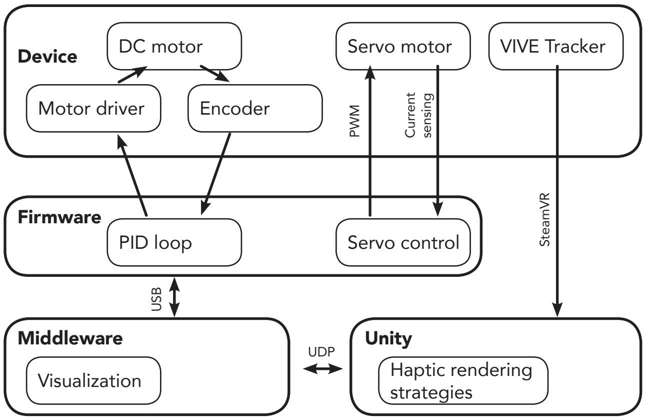

Figure 4. Architecture of the Haptic Revolver device and software stack. The device is powered by a PSoC, which sends commands to the two motor drivers and communicates with the PC via a USB serial connection.

Although we envision this device eventually being wireless, for simplicity, we use the device with a power and data tether. In our current implementation, both the power supply and electronics are external to the device. We note that there is enough room within the device grip for eventual placement of electronics and a battery. The device, including the VIVE tracker, weighs 237 g, which is comparable to a VIVE Controller (205 g). Table 1 shows additional device specifications.

Weight	237 g
Max wheel speed	180 rpm
Wheel diameter	60 mm
Max motion under finger ($\alpha = 1$)	565 mm/s
Max force against finger	3.35 N
Typical power consumption	1.25 W
Peak power consumption	2.5 W

Table 1. Mechanical and electrical specifications of our device

Software Architecture

The software architecture is summarized in Figure 4. The device is controlled by firmware running on a Cypress Programmable System on Chip (PSoC) 5LP. A PID loop on the PSoC turns the wheel to a specified rotation from a known starting angle using an external motor driver (DRV8871). The PSoC also drives the servo through pulse-width modulation (PWM). The two motors are each powered by separate step-up power regulators (Pololu U3V50F*). In order to measure the force exerted by the finger against the wheel, the PSoC also monitors the voltage across a 1 Ω shunt resistor in series with the servo motor. The PSoC interfaces with a PC using a USB serial connection running at 115 200 baud.

A Python middleware layer on the PC handles communication with the device, visualization and logging, and communication with the VR application. The application layer is built with the Unity 3D game engine. Our Haptic Revolver rendering engine in Unity determines the ideal wheel configuration and streams the desired settings to the Python middleware using a socket connection.

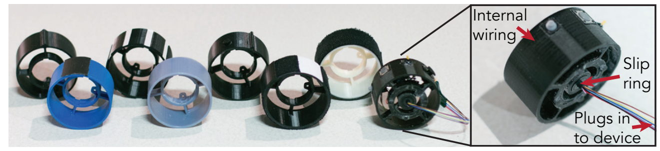

Figure 5. Interchangeable haptic wheels allow applications to customize the haptic experience with various shapes and textures. (right) Wheels with active components use a slip ring in the wheel mount to wire the electronics in the wheel back to the device.

Interchangeable Haptic Wheels

While a simple plastic wheel can simulate touch contact and motion of the finger, there are many applications that would benefit from custom textures or shapes placed on the wheel to match elements in the virtual environment. Haptic wheels, such as those shown in Figure 5, are designed to slide onto the wheel mount and can be 3D printed or manufactured from other materials. The ability to customize wheels allows us to render certain objects with much higher fidelity. For example, a simple plastic wheel can be easily augmented by affixing materials with unique textures, such as cloth, rubber, or paper, which correspond to particular objects in the virtual scene. Textures such as bumps and grooves can also be printed directly into the wheel itself to render various surface textures. Coarser shapes printed on the wheel can simulate larger features in the scene. For example, a wheel with a raised region, such as the one shown in Figure 6 can be used to render edges. With such a simple wheel, a user can feel the boundaries of a physical button or feel when the finger slides off the edge of a surface. Other custom shapes can be designed to match specific objects in the scene. In a sculpting application, for instance, appropriate shapes on the wheel can allow the user to feel a tool beneath their finger during use.

Although many sensations can be rendered using passive wheel elements alone, additional functionality can be achieved with active wheels containing electronic components. For example, an active wheel can include input elements, such as buttons and switches that directly map to virtual widgets and add interactivity. Components such as Peltier elements can be added to the wheel to create additional haptic sensations that can be controlled dynamically.

To deliver electrical contacts onto a spinning wheel, we designed a wheel containing a slip ring, as shown in Figure 5 (right). Up to 12 wires attach to elements on the wheel, pass through the slip ring and out the front of the device, and plug in to a port on the bottom of the device. We created one such wheel with input elements that include buttons, a switch, and a joystick to enable interaction with different virtual widgets. We envision future designs of our device to include a through-bore slip ring in the wheel mount itself. In this design, electrical contacts to the wheel would be made through a physical connection along the outer face of the mount. This would move all the electrical components to the interior of the device and enable custom active wheels without the overhead of additional wiring.

HAPTIC REVOLVER RENDERING ENGINE

Because each wheel has a limited surface area, haptic elements on the wheel will likely not precisely match the size and position of elements in the virtual environment. We developed a rendering engine to analyze the scene, hand trajectory, and wheel configuration and determine how to control the device. As there may be competing goals within the scene, the engine operates by constructing and resolving a set of constraints to take into account dragging motion and the desired orientation of the wheel. At each frame (roughly 90 Hz), we raycast beneath the finger to determine the nearest collision with a haptic surface. To minimize jerky movements of the controller, we smoothly raise the wheel as the finger approaches a surface. We use similar penetration compensation techniques as described by Benko et al [3]. As the hand penetrates the virtual surface, we raise the wheel even further to provide pressure feedback to the user. Visually, the hand remains at the same height.

From the predicted penetration point, we scan left and right to determine which other haptic elements are nearby. If the user is making contact with a surface, we add shear constraints to move the wheel along with the user's motion. If no contact is made, we allow the wheel to spin quickly to ensure the constraints are met. If other haptic elements are nearby, we add positional constraints to ensure that the features on the wheel align with the virtual elements. In the constraint resolution step, we resolve any shear constraints with positional constraints to arrive at a desired wheel orientation.

To illustrate this process, consider the simple virtual scene shown in Figure 6 used with a wheel containing a small raised region, shown in black. As the user hovers over the blue surface (left), we ensure the correct texture (shown in blue on the wheel) is placed beneath the finger. We also make note of the nearby raised surface (shown in black) and add that as a constraint. As the user moves closer to the raised surface (center), the edge constraint is given higher priority and the feature on the wheel is brought close to the finger. If the user were making contact with the surface, they would feel the edge in the correct location. Finally, as the user moves onto the raised region (right), we impose two constraints, one for the edge in each direction. This effectively scales the gain of the wheel rotation so that the edges are placed in the correct spot, regardless of the size of the raised region.

When rendering shear forces during contact with a surface with no other constraints, a natural option is to spin the wheel such that the linear movement under the finger matches the linear movement in virtual space. In practice, this leads to quickly running out of room on the wheel before another feature arrives. To balance the realism of the dragging motion with practical constraints, we choose to reduce the wheel gain, α, to 0.6. For every 1 cm of finger motion, we spin the wheel such that it travels 0.6 cm beneath the finger. This value was chosen based on the results of a perceptual study described later. It represents the smallest gain before significant reductions in realism are observed.

Because our device uses a wheel, it inherently renders motion in only one dimension (horizontally). While this is appropriate in many scenarios, it would be ideal to support motion in two dimensions (horizontally and vertically). Because we noticed that users were insensitive to the direction of motion under

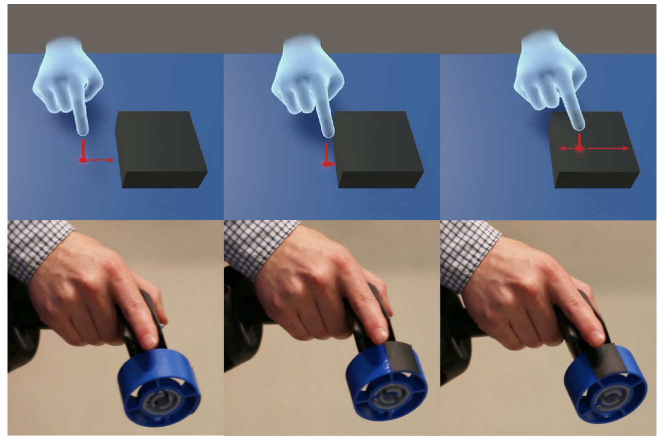

Figure 6. (left) When a user hovers over the blue surface, the rendering engine places the appropriate wheel surface under the finger and begins to track the nearby edge of the black surface. (center) As the user approaches the edge, the rendering engine positions the wheel so that the edge approaches the finger. (right) While hovering over the smaller black surface, the rendering engine adjusts the gain of the wheel so that the two edges are rendered correctly.

the finger when the surface is smooth, we simulate vertical motion by simply spinning the wheel horizontally. Although this is orthogonal to the actual direction of motion, prior work supports the feasibility of this illusion [30]. In this mode, we choose the direction based on the horizontal component of velocity and allow the wheel to switch directions only when there is a sudden change in hand direction or when the hand comes to a stop. We evaluate the efficacy of this rendering technique for different types of tracing behavior in Study 2.

Rendering with Custom Wheels

To allow new wheel designs to be easily added without modifying the scene, we created a simple wheel description specification, implemented as a JSON file. This file contains a list of features on the wheel and where they are located. The rendering engine uses this wheel description file to determine how to control the device. As the finger approaches a haptic element in the scene, the virtual object reports its desired haptic feature, such as a soft texture. The rendering engine finds the appropriate element on the wheel and turns it according to the constraint resolution steps. If a desired haptic element, such as a soft texture, is not present on the wheel, the engine will fall back on a suitable replacement feature, such as a smooth

```
{
  "name": "CasinoWheel",
  "features":[
    { "start": 0, "stop": 47.7, "height": 1,
      "texture": "hard", "name": "poker"},
    { "start": 47.7, "stop": 90, "height": 0,
      "texture": "soft", "name": "felt_small"},
    { "start": 90, "stop": 135, "height": 0,
      "texture": "paper", "name": "card"},
    { "start": 135, "stop": 360, "height": 0,
      "texture": "soft", "name": "felt_large"}
  ]
}
```

Figure 7. An example wheel description file used by the rendering engine. This file describes the wheel shown in Figure 12.

texture. Figure 7 shows an example wheel description file for the wheel shown in Figure 12.

As additional haptic features are added to the wheel, the amount of wheel space available for any one feature is reduced. This can make it more difficult to render the sensation of dragging along a surface, as the finger will quickly collide with an additional haptic element. To address this challenge, we developed several strategies for hiding undesired features on the wheels. These strategies are illustrated in Figure 8.

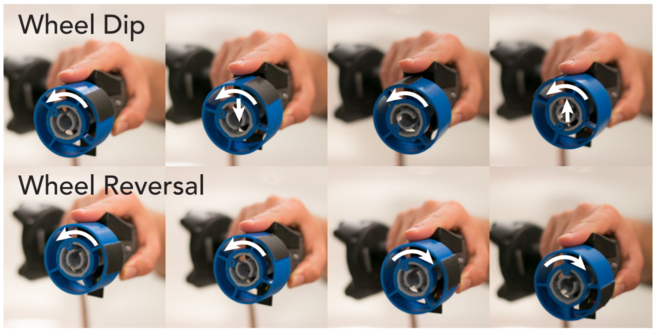

Figure 8. (top) As the finger approaches an obstacle (indicated here by the black region on the wheel), the dip strategy causes the wheel to lose contact with the finger while an undesired feature remains under the finger. (bottom) In the reversal strategy, the wheel begins to rotate in the opposite direction when an undesired feature is encountered.

Wheel Dip: Our first strategy simply lowers the wheel just before a feature would approach the finger. We ease the wheel position in and out to create a smooth transition and prevent jerky behavior. This strategy has the effect of causing the finger to lose touch with the wheel for a short period of time. We can actually shorten the time without contact by accelerating the wheel over the undesired feature once it has lost contact with the finger. This can reduce the amount of disturbance caused to the user.

Wheel Reversal: As an alternative strategy, we simply reverse the wheel direction before a collision occurs. While dragging along a surface using a wheel with other elements, the wheel will rotate back and forth to render the appropriate shear motion while keeping the finger on the correct region of the wheel, effectively hiding the other haptic features. This behavior is supported by our findings from Study 1, which revealed that rendering motion in the opposite direction has little impact on perceived realism. Although the overall direction of motion may not be noticeable, the act of switching directions does cause a noticeable shear against the fingertip. While this is not entirely unavoidable, we can reduce the frequency of such reversals by reducing the wheel gain, α. This method is also less noticeable if the user grips the controller more tightly, which reduces horizontal finger wobble during a direction change.

Because these two methods each have their advantages, we use them both in our rendering engine, depending on the circumstances. When a user is sliding over a surface that has a large physical size on the wheel, we use the reversal technique. We observe that this technique causes less disturbance overall because it never loses contact with the finger. If the physical region is small or the wheel needs to change orientations to accommodate other constraints, we use the wheel dip technique to skip ahead to the desired wheel region.

EVALUATION

To evaluate the fundamental haptic capabilities of Haptic Revolver, we conducted two studies to understand how wheel parameters impact the realism of the haptic rendering. In the first study, we measure the impact of the wheel speed gain and direction. In the second study, we explore simulating motion in two dimensions using a single wheel. These studies helped inform the design of our haptic rendering techniques. We recruited 12 right-handed participants (10 male, 2 female), age 18 to 48, to participate in both studies. Participants were instructed about the nature of the study and given a short overview of the Haptic Revolver device. Participants then put on an HTC VIVE head-mounted display and held our Haptic Revolver device in their right hand and a standard HTC VIVE controller in their left hand. All studies were conducted while the participants were standing. Participants briefly explored a demonstration scene where they were able to touch and swipe on a virtual object before the studies began. Each study took approximately 20 minutes and participants were compensated with an $8 meal coupon at a nearby cafeteria for their time.

Study 1: Rotation gain

In this study, we sought to understand how the wheel speed gain and direction impact the perceived realism of the haptic rendering. When the user's hand moves a distance of x to the right, the wheel spins such that a distance of αx has passed underneath the finger to the left. To most closely match reality, we would set the gain, α, to 1. However, in some cases, it is useful to modify the gain to move the wheel to a desired orientation more quickly or more slowly. We also wanted to explore how important it is to spin the wheel in the correct direction. We hypothesized (**H1**) that a one-to-one mapping from virtual motion to wheel motion (a gain of 1.0) would be most realistic. We further hypothesized (**H2**) that users would prefer that the wheel spins in the natural direction ($\alpha > 0$), but that they would prefer a reverse spin ($\alpha < 0$) to no spin ($\alpha = 0$).

To test this, we asked users to swipe their finger along the length of a 50 cm wide virtual surface under 17 different gain settings from $\alpha = -1.6$ to 1.6 in increments of 0.2. Participants began the trial by positioning the tip of their finger within a small sphere on the surface. After exploring the surface for several seconds, participants ended the trial by moving their finger within the bounds of a second sphere, positioned just above the surface. We then asked the user to rate the haptic

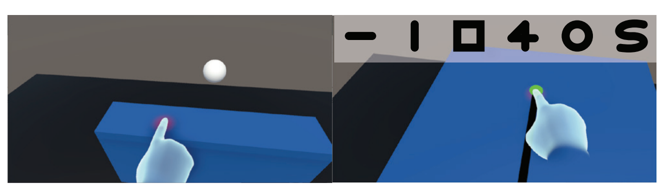

Figure 9. (left) In the first study, users slid their finger horizontally across a surface. (right) In the second study, users traced a path on a surface. (right, inset) The six paths used in the second study

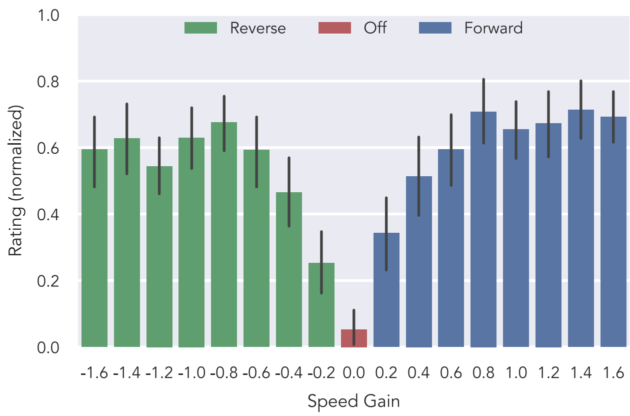

Figure 10. Results of the first user study showing mean realism ratings across participants as a function of the wheel speed gain. The error bars show a 95% confidence interval. A negative gain indicates the wheel was spun in the opposite direction.

realism by responding to an on-screen prompt. The prompt asked users "*How closely did the haptic rendering match your visual impression of the scene?*". All responses were collected on a 5-point Likert scale (1-not at all realistic, 5-highly realistic) by pointing at the desired response on screen and clicking with the VIVE controller in the left hand. Each block consisted of a single repetition of each gain setting presented in a random order. Participants completed three blocks each.

Results

Figure 10 shows the average rating across participants for each condition. In this plot, we normalize responses such that the highest and lowest responses for each participant become '5' and '1', respectively. Participants reported the lowest realism score for $\alpha = 0$, or when the wheel never moved. Realism scores increased as the wheel gain increased, but leveled off around $\alpha = 0.6$. Wheel direction had little impact on realism as shown by the symmetric nature of the graph. When asked after the study, only two users even noticed that the wheel was spinning in the reverse direction some of the time. This is consistent with prior work, which found that the direction of skin deformation had little impact on realism [30]

These results support several aspects of our rendering techniques. Most importantly, it suggests that as long as the wheel speed gain is at least 0.6, the gain does not matter much. This allows for some flexibility to spin the wheel faster or slower and accommodate other constraints. Second, these results validate our approach to avoiding features by reversing the wheel direction. Finally, these results do not conflict with our decision to render vertical motion with horizontal motion under the finger.

Study 2: Vertical movement

Though Haptic Revolver only spins in one dimension, it is important to explore whether we could effectively render motion in two dimensions. Since the results of Study 1 suggest that spinning in the opposite direction had little impact on perceived realism, we hypothesized that spinning in an orthogonal direction would also have little impact.

To explicitly test this, we displayed a path on a flat surface and asked users to trace the path in the forward and reverse directions. Each experimental block consisted of six paths and five wheel behaviors for a total of 30 trials, which were presented in a random order. Participants completed each block twice. To explore the effect of path shape, we chose paths (Figure 9, right) that include a combination of horizontal and vertical motion as well as a mixture of sharp edges and curves. Paths were scaled to fit within a 25 cm by 25 cm square.

In addition to modes that render wheel motion in the horizontal direction (Motion 1D) and in both directions (Motion 2D), we introduce three other baseline conditions. Our five wheel conditions are:

- **Motion 1D**: As the finger moves horizontally, the wheel spins with a gain of $\alpha = 1$. As in the previous study, moving vertically causes no change to the wheel.
- **Motion 2D**: Similar to Motion 1D, except the wheel also spins when the finger moves vertically. After a sudden change in direction or when the finger comes to a near stop, we reevaluate the spin direction according to the horizontal component of velocity.
- **Off**: A control condition in which the wheel does not spin at all. This is equivalent to $\alpha = 0$ in the previous study.
- **Shear 1D**: As the finger moves horizontally, the wheel turns slightly, causing skin deformation proportional to the horizontal velocity. Moving vertically causes no change to the wheel.
- **Shear 2D**: Similar to Shear 1D, except it also applies skin deformation when the finger moves vertically. After a sudden change in direction or when the finger comes to a near stop, the deformation direction is reset according to the horizontal component of velocity.

As in the first study, we then asked users to rate "*How closely did the haptic rendering match your visual impression of the scene?*" on a 5-point Likert scale (1-not at all realistic, 5-highly realistic). We hypothesized that users would find the Motion conditions more realistic than the Shear or Off conditions (**H3**) and the Motion 2D condition would be most realistic (**H4**).

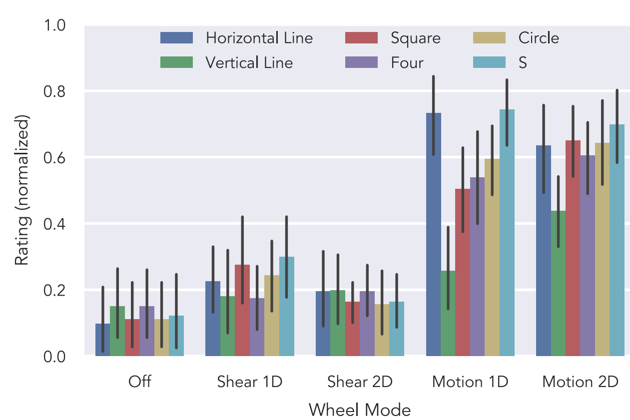

Figure 11. The results from the second study showing mean realism ratings across participants as a function of the wheel rendering mode and path drawn. The error bars indicate a 95% confidence interval.

Results

The mean ratings from all participants are summarized in Figure 11. Mann-Whitney U tests show that participants perceived the Motion conditions ($n = 288$, median $= 0.67$) as more realistic than the Shear conditions ($n = 288$, median $= 0.0$, $U = 67382$, $p < 0.001$) and the Off condition ($n = 144$, median $= 0.0$, $U = 35494$, $p < 0.001$). Participants also perceived the Shear conditions as more realistic than the Off conditions ($U = 25107$, $p < 0.001$). This confirms **H3** and highlights the importance of rendering more than just skin deformation under the finger.

Ultimately, we did not find strong support for **H4** as no significant difference was observed between the Motion 1D ($n = 144$, median $= 0.67$) and Motion 2D ($n = 144$, median $= 0.67$) conditions in aggregate. However, by breaking down the analysis by path, we find some trends that suggest our 2D rendering technique is still effective. Participants perceived Motion 2D as more realistic than Motion 1D when tracing a vertical line (medians $= 0.33, 0.25$, $n = 24$, $U = 178.5$, $p = 0.021$) and a square path (medians $= 0.66, 0.5$, $n = 24$, $U = 208$, $p = 0.096$), paths which both have vertical components. No significant differences were observed with the other paths. While we expect no differences in the horizontal line, the other paths contain significant diagonal or curved components. In these cases, the difference between our Motion 1D and 2D rendering largely comes down to a difference in speed. For example, in the diagonal portion of the circular path, the 1D mode would render the motion at half the speed of the 2D mode, since only half of the motion lies in the horizontal direction. Since we are not highly sensitive to the magnitude of motion (as confirmed by Study 1), it is unsurprising that differences were not observed on these paths.

EXAMPLE APPLICATIONS

To explore applications for Haptic Revolver, we built several scenes and corresponding haptic wheels that highlight different capabilities of the device. We invited 11 users to try out three of these demos in order to elicit qualitative feedback on our device and rendering techniques. We also refer the reader to the Video Figure accompanying this paper for a demonstration of each of these applications.

Card Table (Texture Rendering): The first application highlights the ability of Haptic Revolver to render different textures under the fingertip. In this scene, several playing cards and poker chips lie on a card table within a virtual casino. A user can touch and drag along the felt table surface, the playing cards, and the plastic poker chips and feel an appropriate texture beneath the finger. For this application, we designed a wheel containing *felt*, *plastic*, and *paper*. As shown in Figure 12 (left), two felt regions are used in order to render the transition from paper or plastic to felt in either direction. If a user presses lightly on one of the virtual objects, the finger will slide over it and feel the surface moving beneath it. If a user presses harder, the object will be dragged along with the finger and the device will render a shear force due to friction.

Painting and Sculpting (Force Sensing): Haptic Revolver can also turn passive props on the wheel into interactive objects by sensing the force on the wheel. In this scene, a user can paint and sculpt a 3D model by choosing between a *spray-paint tool*, a *finger painting tool*, and a *sculpting tool*. As shown in Figure 12 (center), the wheel consists of a raised plastic cylinder to simulate the top of a can of spray-paint and a narrow ridge to simulate the back of a knife. The tool and color can be selected by pointing at the desired element and clicking with the thumb button. When a tool is selected, the appropriate haptic element is positioned under the finger and left there until a new tool is selected. To use the tool, a user simply presses down on the haptic element beneath the fingertip. The device detects this added force on the wheel and activates the tool. When finger-painting, a smooth surface of the wheel is positioned under the finger and it spins back and forth during use to render shear forces and motion.

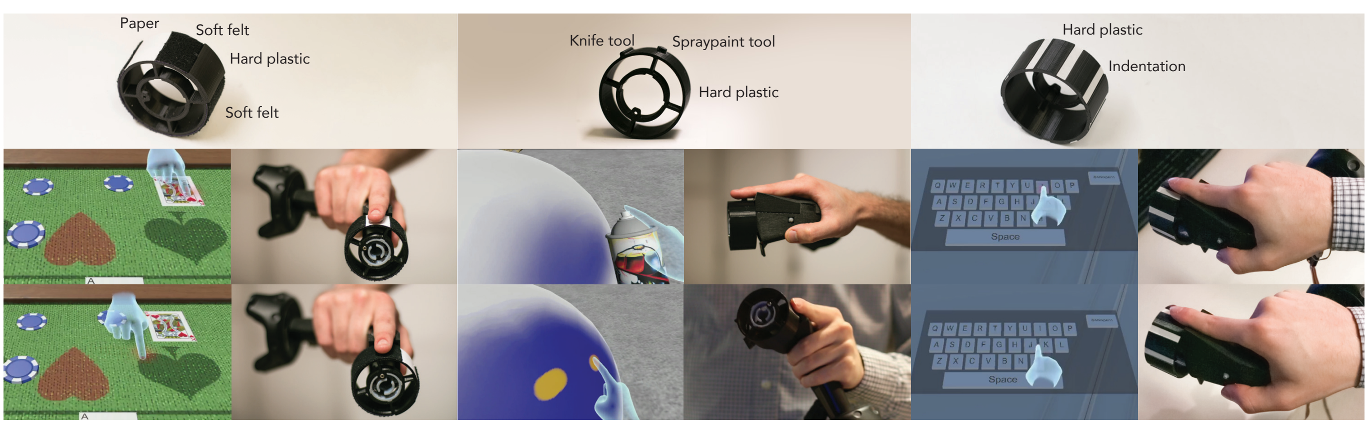

Figure 12. (left) A card table demo that highlights our ability to render different textures. The wheel used in this demo consists of two regions of soft felt, a hard plastic ridge, and a small section of paper. When the user touches an object in the scene, the appropriate texture is placed underneath the fingertip. (center) A painting and sculpting demo that highlights the ability to render shapes and sense the force applied to the wheel. The wheel used in this demo consists of a raised nub and a ridge to simulate holding tools. The user presses on the wheel to activate the tool. The model can be explored by touch. (right) A keyboard demo that highlights our ability to render edges and shapes. The wheel used for this demo consists of nine raised plastic regions with grooves in between. When a user approaches the edge of a key, the edge of a groove is placed under the finger.

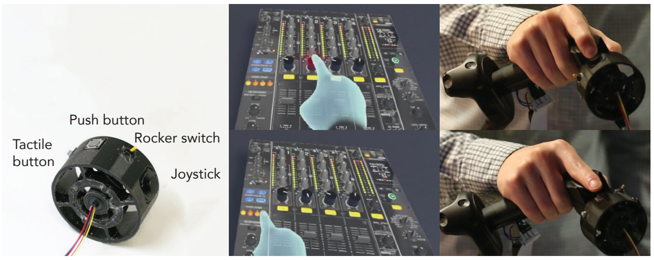

Figure 13. A demo with a DJ mixer board that highlights our ability to put interactive elements on the wheel. The wheel in this demo consists of several physical UI elements wired up to the device. When a user touches a virtual UI element, not only do they feel the shape of a similar physical element, but they can physical interact with the widget.

Keyboard (Shape Rendering): In this application, we highlight the ability of Haptic Revolver to render custom shapes under the fingertip and improve the experience of using virtual buttons with a purely passive wheel. In this scene, an on-screen keyboard allows a user to enter text by pressing on the virtual key with their finger. As shown in Figure 12 (right) the wheel for this scene consists of nine raised plastic ridges, each approximately the size of a standard key on a keyboard. Each raised "key" is separated by a small indentation to simulate the gap between keys. When the user presses on a virtual key, the rendering engine ensures the edges for the physical ridge align with the edges of the virtual key (see Figure 12, right). The gap can be felt by touching between two virtual keys (see Figure 12, bottom right). In fact, a user can lightly brush along an entire row of the keyboard and feel a rapid succession of bumps, much like one would feel on a real keyboard.

DJ Mixer (Active Wheels): While the previous applications have highlighted the capabilities of Haptic Revolver using only passive wheels, additional functionality can be added through the use of active electronic elements. In this DJ mixer application, we haptically render a virtual DJ mixer using a wheel with *active buttons*, a *rocker switch*, and a *low profile two-axis joystick*. Each widget on the mixer board is linked to a physical widget on the device. Buttons and switches have a direct mapping on the device. A joystick on the wheel metaphorically maps to virtual knobs and dials. Pushing the joystick in a particular direction rotates the knob to the same direction. Sliders are rendered using a switch on the wheel, though a simple passive object would suffice as well. When a user touches the thumb of the slider, the tactile switch is positioned under the finger. Much like the dragging behavior in the card table demo, moving the slider causes the haptic element under the finger to tug against the skin, rendering frictional shear forces. When the slider reaches its extreme point, the haptic element begins to slide off the finger.

User Feedback

To better understand the performance of our device, we invited an additional 11 users (10 male, 1 female) from our institution who had not tried the device before to provide feedback on its use. We sought to understand how our Haptic Revolver device compares to standard vibrotactile notification. During the study, participants tried three of our example applications: the card table scene, the keyboard scene, and the painting and sculpting scene. For simplicity and timing, we omitted the fourth DJ mixer scene. Upon arriving, users were given an introduction to the device and head-mounted display, and allowed to become accustomed to our device in a simple tutorial scene. Over the next thirty minutes, participants tried the three scenes using both our device, with the appropriate wheel for each scene, and a standard HTC VIVE controller. The VIVE controller vibrated upon contact with a virtual surface.

To elicit reactions to our device, participants explored a scene through a guided walkthrough and then provided feedback through a semi-structured interview about their experience. Questions focused on the haptic realism of various aspects of the scene, preferences related to the rendering of both devices, and usability aspects of our device. To enable a more quantitative comparison between devices, we also asked users to rate the haptic rendering ("How well did the haptic rendering match your visual impression of the scene") of each device on a 5-point Likert scale after each scene. Participants spent 3-5 minutes exploring each combination of scene and controller. Participants experienced both controllers within a scene before moving on to the next scene. We randomized the presentation order of both the scenes and the devices. Participants were compensated with an $8 meal coupon for their time.

Results

Participants were generally excited about our device and appreciated that it could render more than just vibrations. Many participants remarked that while using the Haptic Revolver, they felt like they were actually touching the surface. In the card table scene, P10 remarked, "It actually felt like I was moving my finger along a felt table". P3 noted that when touching a surface, our device responds based on how they move their hand, which felt much better than the vibrotactile controller. While using the vibrotactile notification, many users remarked that it was not the sensation they were expecting. Some users noted that while it did not feel realistic, they still appreciated the vibration feedback as an indication of touch. Users also rated our device more realistic than vibrotactile notification with consistent median ratings of 4 for our device and 3 for vibrotactile ($n = 11$) in each scene. Wilcoxon signed-rank tests between the realism responses showed that these differences were significant for the card table ($T = 0$, $p = 0.003$),

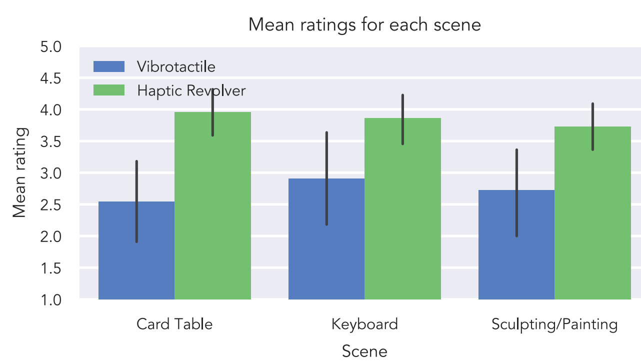

Figure 14. Quantitative results of our feedback elicitation study showing mean realism ratings across participants for our Haptic Revolver device and vibrotactile notification. The error bars indicate a 95% confidence interval. All differences are significant at the 0.05 significance level

keyboard ($T = 6$, $p = 0.016$), and sculpting scenes ($T = 0$, $p = 0.003$).

This study also revealed opportunities to improve future iterations of the device. Several participants commented on the noise made by the motors. This can be improved in future iterations by using higher quality brushless motors. Some users found our reversal technique for avoiding sections of the wheel distracting, particularly when moving slowly. For example, P4 did not realize the reversal technique was intentional and commented "the motion didn't always match up with the motion of the finger". In our testing, a user's sensitivity to the reversal can be significantly reduced by ensuring the finger is pressed down against the groove on the controller, minimizing the horizontal sway during the reversal. Adding an elastic finger strap to the device may help ensure the finger rests in the ideal place. In the keyboard demo, many participants attempted to explore the sides of the keyboard, which were visually rendered as a smooth surface. Participants were surprised when they could feel edges while touching this surface. For generality, wheel designs should include a larger smooth region that can be utilized as a general purpose touch surface.

A few unexpected observations arose from this study. First, we observed differences between users with VR experience and novice or first-time VR users (seven participants had used a VR system for less than an hour). The attention of novice users was largely consumed by the novelty of the VR experience and visual aspects of the scene. With these users, it was more difficult to elicit feedback related to the haptic rendering.

Interestingly, we also observed that users started to comment on more nuanced haptic features while using our device. For instance, P4 noted that for the poker chip in the card table scene, they could feel the texture, but it was not exactly like a poker chip. As another example, P7 said "It doesn't feel exactly like spray-paint, but this is cool. I have a nice sense that I'm holding it." We suspect that because our device rendered textures and shapes in an attempt to match reality, users had higher expectations and noticed subtle imperfections, similar to the uncanny valley effect observed in 3D animation. While vibrotactile notification provided touch feedback, it was clearly not realistic and users did not expect it to be. This raises interesting questions about what level of detail is appropriate in haptic rendering for VR.

Finally, we observed that not all users preferred greater realism in the experience. For example, P8 initially felt shocked when they could feel our device rendering surface textures and remarked, "It was hard to get used to it touching my finger. It felt more real, but that was a bad thing. If I had more time I could've gotten used to that". This suggests that while it is important to explore methods of accurately rendering haptic feedback, realism may not be the only design goal.

DISCUSSION, LIMITATIONS, AND FUTURE WORK

Haptic Revolver goes beyond vibrotactile notification to explore what rendering multiple haptic sensations can add to the VR experience. While a simple haptic wheel can render touch contact and motion under the finger, the use of interchangeable haptic wheels allows applications to design custom haptic experiences. We envision some haptic wheels to be general purpose, while others may be tailored to provide highly realistic experiences for certain applications. When an end user purchases a new application that would benefit from a customized haptic experience, it could come with its own haptic wheel. Future work could explore actuating the wheel axis forward and backward to automatically switch between multiple wheels installed on the device.

Our ability to place electronic components on the wheel significantly broadens the design space of haptic wheels. We chose to focus on interactive elements such as buttons and switches because these are commonly found in virtual environments. Our choice of physical widgets supports a wide range of virtual elements such as those found in a airplane cockpit or a car dashboard. However, electronic elements placed on the wheel are not limited to these physical controls. For example, a Peltier element on the wheel could enable temperature feedback. Adjustable mechanical components could enable additional flexibility by dynamically changing the wheel in response to the virtual environment.

We add additional interactivity to haptic wheels by sensing the force the user applies. In one of our example applications, we use this as a binary indicator of pressure to activate the spray-paint. Further interactivity could be added by using the continuous pressure signal. For example, a user might touch with varying pressure to control the spread of the paint stream.

In our keyboard demo, we improve the realism of the scene by adding ridges so that the user can feel the edges and shape of the keys. Users enjoyed this and remarked on its realism, but it is currently unclear whether this can improve performance on the keyboard in any way. Exploring how physical haptic feedback impacts task performance is an interesting avenue for future work.

Lastly, though we made an effort to design our device for users with varying hand sizes, individual differences cause subtle changes in how the fingertip rests on the wheel. This results in a different resting height for the wheel which, for some users, was manifest as physical contact before virtual contact was made. Future versions could include a proximity sensor in the wheel to automatically calibrate the correct height.

CONCLUSION

Haptic Revolver is a general-purpose handheld VR controller that goes beyond vibrotactile stimulation to render touch contact with virtual surfaces, motion along a surface, textures, and shapes using interchangeable haptic wheels. By customizing wheels for the virtual environment, designers can use Haptic Revolver to render realistic haptic feedback on the fingertip. We demonstrated techniques to render motion along a surface in two dimensions and adapt a particular wheel for use in arbitrary scenes. We conducted two user studies to inform and validate the design of our haptic rendering techniques and a third study to elicit qualitative feedback from participants. We believe that Haptic Revolver offers high-fidelity haptic rendering with clear advantages over vibrotactile solutions and we hope others will build upon our design to continue enabling better haptic experiences for VR.

REFERENCES

1. Bruno Araujo, Ricardo Jota, Varun Perumal, Jia Xian Yao, Karan Singh, and Daniel Wigdor. 2016. Snake Charmer: Physically Enabling Virtual Objects. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '16)*. ACM, New York, NY, USA, 218–226. DOI: http://dx.doi.org/10.1145/2839462.2839484
2. Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. 2016. Haptic Retargeting: Dynamic Repurposing of Passive Haptics for Enhanced Virtual Reality Experiences. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1968–1979. DOI: http://dx.doi.org/10.1145/2858036.2858226
3. Hrvoje Benko, Christian Holz, Mike Sinclair, and Eyal Ofek. 2016. NormalTouch and TextureTouch: High-fidelity 3D Haptic Shape Rendering on Handheld Virtual Reality Controllers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 717–728. DOI: http://dx.doi.org/10.1145/2984511.2984526
4. Mourad Bouzit, Grigore Burdea, George Popescu, and Rares Boian. 2002. The Rutgers Master II-new design force-feedback glove. *IEEE/ASME Transactions on mechatronics* 7, 2 (2002), 256–263.
5. Tom Carter, Sue Ann Seah, Benjamin Long, Bruce Drinkwater, and Sriram Subramanian. 2013. UltraHaptics: Multi-point Mid-air Haptic Feedback for Touch Surfaces. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 505–514. DOI: http://dx.doi.org/10.1145/2501988.2502018
6. Francesco Chinello, Monica Malvezzi, Claudio Pacchierotti, and Domenico Prattichizzo. 2015. Design and development of a 3RRS wearable fingertip cutaneous device. In *Advanced Intelligent Mechatronics (AIM), 2015 IEEE International Conference on*. IEEE, 293–298.
7. Inrak Choi, Elliot W Hawkes, David L Christensen, Christopher J Ploch, and Sean Follmer. 2016. Wolverine: A wearable haptic interface for grasping in virtual reality. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 986–993.
8. H. Culbertson, J. Unwin, B. E. Goodman, and K. J. Kuchenbecker. 2013. Generating haptic texture models from unconstrained tool-surface interactions. In *2013 World Haptics Conference (WHC)*. 295–300. DOI: http://dx.doi.org/10.1109/WHC.2013.6548424
9. exiii. 2016. EXOS Project. (2016). Retrieved September 3, 2017 from http://exiii.jp/exos/
10. Haption. 2013. Haption Virtuose 6D. (2013). Retrieved August 31, 2017 from https://www.haption.com/site/index.php/en/products-menu-en/hardware-menu-en/virtuose-6d-menu-en
11. Diana Krusteva, Deepak Sahoo, Asier Marzo, Sriram Subramanian, and David Coyle. 2015. Marionette: A Multi-Finger Tilt Feedback Device for Curvatures and Haptic Images Perception. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15)*. ACM, New York, NY, USA, 1229–1234. DOI: http://dx.doi.org/10.1145/2702613.2732729
12. Katherine J Kuchenbecker, David Ferguson, Michael Kutzer, Matthew Moses, and Allison M Okamura. 2008. The touch thimble: Providing fingertip contact feedback during point-force haptic interaction. In *Haptic interfaces for virtual environment and teleoperator systems, 2008. haptics 2008. symposium on*. IEEE, 239–246.
13. Thomas H Massie, J Kenneth Salisbury, and others. 1994. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, Vol. 55. Chicago, IL, 295–300.
14. Novint. 2011. Novint Falcon. (2011). Retrieved September 3, 2017 from http://www.novint.com/index.php/products/novintfalcon
15. Claudio Pacchierotti, Gionata Salvietti, Irfan Hussain, Leonardo Meli, and Domenico Prattichizzo. 2016. The hRing: A wearable haptic device to avoid occlusions in hand tracking. In *Haptics Symposium (HAPTICS), 2016 IEEE*. IEEE, 134–139.
16. Claudio Pacchierotti, Stephen Sinclair, Massimiliano Solazzi, Antonio Frisoli, Vincent Hayward, and Domenico Prattichizzo. 2017. Wearable haptic systems for the fingertip and the hand: taxonomy, review, and perspectives. *IEEE Transactions on Haptics* (2017).
17. Domenico Prattichizzo, Francesco Chinello, Claudio Pacchierotti, and Monica Malvezzi. 2013. Towards wearability in fingertip haptics: a 3-dof wearable device for cutaneous force feedback. *IEEE Transactions on Haptics* 6, 4 (2013), 506–516.
18. William R Provancher. 2014. Creating greater VR immersion by emulating force feedback with ungrounded tactile feedback. *IQT Quarterly* 6, 2 (2014), 18–21.
19. Dexta Robotics. 2017. Dexmo. (2017). Retrieved September 3, 2017 from http://www.dextarobotics.com/
20. Samuel B. Schorr and Allison M. Okamura. 2017. Fingertip Tactile Devices for Virtual Object Manipulation and Exploration. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3115–3119. DOI: http://dx.doi.org/10.1145/3025453.3025744
21. Massimiliano Solazzi, Antonio Frisoli, and Massimo Bergamasco. 2010. Design of a novel finger haptic interface for contact and orientation display. In *Haptics Symposium, 2010 IEEE*. IEEE, 129–132.

22. Paul Strohmeier and Kasper Hornbæk. 2017. Generating Haptic Textures with a Vibrotactile Actuator. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4994–5005. DOI: http://dx.doi.org/10.1145/3025453.3025812

23. CyberGlove Systems. 2017. CyberGrasp. (2017). Retrieved September 3, 2017 from http://www.cyberglovesystems.com/cybergrasp/

24. Richard Q Van der Linde, Piet Lammertse, Erwin Frederiksen, and B Ruiter. 2002. The HapticMaster, a new high-performance haptic interface. In *Proc. Eurohaptics*. 1–5.

25. Bill Verplank, Michael Gurevich, and Max Mathews. 2002. The Plank: Designing a Simple Haptic Controller. In *Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME '02)*. National University of Singapore, Singapore, Singapore, 1–4. http://dl.acm.org/citation.cfm?id=1085171.1085180

26. Go Touch VR. 2017. Touch the Virtual Reality with VR Touch. (2017). Retrieved September 6, 2017 from https://www.gotouchvr.com/

27. Maarten WA Wijntjes, Akihiro Sato, Vincent Hayward, and Astrid ML Kappers. 2009. Local surface orientation dominates haptic curvature discrimination. *IEEE transactions on haptics* 2, 2 (2009), 94–102.

28. Hiroaki Yano, Shoichiro Taniguchi, and Hiroo Iwata. 2015. Shape and friction recognition of 3D virtual objects by using 2-DOF indirect haptic interface. In *World Haptics Conference (WHC), 2015 IEEE*. IEEE, 202–207.

29. Vibol Yem and Hiroyuki Kajimoto. 2017. Wearable tactile device using mechanical and electrical stimulation for fingertip interaction with virtual world. In *Virtual Reality (VR), 2017 IEEE*. IEEE, 99–104.

30. Vibol Yem, Mai Shibahara, Katsunari Sato, and Hiroyuki Kajimoto. 2016. Expression of 2DOF Fingertip Traction with 1DOF Lateral Skin Stretch. In *International AsiaHaptics conference*. Springer, 21–25.

31. André Zenner and Antonio Krüger. 2017. Shifty: A Weight-Shifting Dynamic Passive Haptic Proxy to Enhance Object Perception in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 23, 4 (2017), 1285–1294.