NormalTouch and TextureTouch: High-fidelity 3D Haptic Shape Rendering on Handheld Virtual Reality Controllers

Hrvoje Benko, Christian Holz, Mike Sinclair, Eyal Ofek
Microsoft Research, Redmond, WA, USA
{benko, cholz, sinclair, eyalofek}@microsoft.com

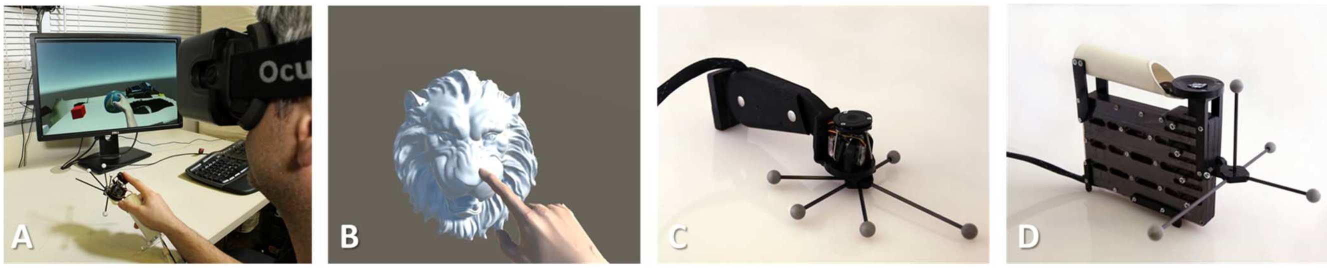

Figure 1: (a) Our 3D haptic shape controllers allow the Virtual Reality user to touch and feel what they would otherwise only see. (b) Our controllers enable users to explore virtual 3D objects with their finger. (c) *NormalTouch* renders the surface height and orientation using a tiltable and height-adjustable platform. (d) *TextureTouch* renders the detailed surface texture of virtual objects using a 4×4 pin array, which users experience on their finger pad.

ABSTRACT
We present an investigation of mechanically-actuated handheld controllers that render the shape of virtual objects through physical shape displacement, enabling users to *feel* 3D surfaces, textures, and forces that match the visual rendering. We demonstrate two such controllers, *NormalTouch* and *TextureTouch*. Both controllers are tracked with 6 DOF and produce spatially-registered haptic feedback to a user's finger. NormalTouch haptically renders object *surfaces* and provides force feedback using a tiltable and extrudable platform. TextureTouch renders the shape of virtual objects including detailed surface *structure* through a 4×4 matrix of actuated pins. By moving our controllers around in space while keeping their finger on the actuated platform, users obtain the impression of a much larger 3D shape by cognitively integrating output sensations over time. Our evaluation compares the effectiveness of our controllers with the two de-facto standards in Virtual Reality controllers: device vibration and visual feedback only. We find that haptic feedback significantly increases the accuracy of VR interaction, most effectively by rendering high-fidelity shape output as in the case of our controllers. Participants also generally found NormalTouch and TextureTouch realistic in conveying the sense of touch for a variety of 3D objects.

UIST '16, October 16 - 19, 2016, Tokyo, Japan

ACM 978-1-4503-4189-9/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2984511.2984526

Author Keywords
Haptics; Controller Design; Tactile Display; Virtual Reality.

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.

INTRODUCTION
The capabilities of current devices to render meaningful haptics lag far behind their abilities to render highly realistic visual or audio content. In fact, the de-facto standard of haptic output on commodity devices is vibrotactile feedback (e.g., built into mobile devices and game controllers). While ubiquitous and small, these vibrotactile actuators produce haptic sensations by varying the duration and intensity of vibrations. This makes them well suited for user-interface notifications, but fairly limited in conveying a sense of shape, force, or surface structure.

In Virtual Reality (VR), higher fidelity haptic rendering beyond vibrotactile feedback has been extensively explored through actuated gloves [11], exoskeletons [6, 10], or stationary robotic arms [13, 25, 26, 34]. While these solutions offer richer haptic rendering, they limit the convenience of use because they either restrict the user to a small working area or they require users to put on and wear additional gear.

As a result, *handheld* controllers—not gloves or exoskeletons—have emerged as the dominant interaction interface for current VR devices and applications (e.g., Oculus Rift, HTC Vive, and Sony PlayStation VR). The haptic feedback these VR controllers provide, however, is vibrotactile—much like on mobile phones and regular game controllers.

In this paper, we explore haptic *3D shape output* on handheld controllers that enables users to feel shapes, surfaces, forces, and surface textures. We present two novel devices, *NormalTouch* and *TextureTouch*, each using a different actuation method to render haptic 3D shape output. As shown in Figure 1c, NormalTouch renders objects' 3D surfaces and provides force feedback to touch input using an active tiltable and extrudable platform, on which the user rests their finger. TextureTouch (Figure 1d) houses a 4×4 matrix of actuated pins underneath the user's fingertip that individually render the 3D shape of virtual objects, including the coarse structure of the surface texture. While we chose VR as an immersive environment for integrating our controllers, they would be equally suitable for haptic output in other scenarios, such as video games, 3D modeling applications, teleoperation, or Augmented Reality.

In contrast to previous approaches, our controllers *integrate* shape output inside a lightweight tracked handheld form factor as shown in Figure 1a. The low weight of the controllers and the ability to track them in 3D throughout a larger environment enables users to obtain the sensation of much larger shapes by freely moving the controllers around and mentally integrating output sensations over time. We argue that this combination of haptic 3D shape output, movability, and 3D tracking produces a much higher-fidelity immersion in virtual scenes than other methods. Using our controllers, users explore 3D scenes tactually in addition to visually, by feeling the virtual objects and surfaces around them with their finger (Figure 1b). Users can hold the controllers in either hand and comfortably explore haptic output using either their index finger or thumb.

Below, we describe the implementation and design of our haptic 3D shape output controllers, their haptic rendering capabilities, as well as their implications for interaction in virtual environments. Finally, we report the results of our three-part evaluation that compared our controllers against vibrotactile feedback and a visual-only baseline condition. The results we found indicate that users get a significantly more accurate sense of virtual objects using haptic feedback, augmenting their perception of shape in virtual 3D scenes.

Contributions

Our paper makes the following four specific contributions:

1. *NormalTouch,* a handheld controller that renders haptics through an active tiltable and extrudable platform and senses force input from the user upon touch.
2. *TextureTouch*, a handheld controller that renders the 3D surface structures via a 4×4 array of actuated pins.
3. The integration of shape controllers in a VR system as well as a series of solutions to interaction challenges, such as object penetration and dynamic object behavior.
4. A user study comparing our two controllers with a visual-only and a vibrotactile feedback baseline, showing gains in accuracy and fidelity of haptic feedback.

RELATED WORK

There is a wide spectrum of haptic solutions, each with unique benefits and limitations. We focus our review of the related work on haptic devices that provide feedback to the user's hand in VR, wearable and mobile haptics, and tactile array displays.

Hand Haptics in Virtual Reality

Our research shares the same primary goal as many other hand haptics VR devices: to effectively render collisions, shapes and forces between the user's hand and the virtual scene.

As discussed in the introduction, the most widely used form of haptic feedback to the hand is vibrotactile actuation. Most common, vibrotactile actuators (including voice coils, eccentric weight motors, and solenoids) are frequently integrated into handheld controllers (e.g., HTC Vive, PlayStation Move and Nintendo Wii controllers), styluses [22, 23], or gloves [11]. For example, commercially available CyberTouch glove [11] uses 6 vibrating motors placed at each fingertip as well as on the palm of the hand, to render simulated tactile sensations to the hand. While most commonly used for simple touch notification, vibrotactile actuators have been used to render an illusion of one-dimensional force [30] and have also been found effective in rendering varying surface stiffness in virtual environments [37]. In our experiments, we used vibrotactile actuation as a baseline technique to compare against our shape rendering haptic actuators.

Larger forces and collisions have traditionally been rendered in VR by actuated articulated arms (e.g., PHANToM [25], Haptic Master [34], Virtuose 6D [13], Falcon [26], Snake Charmer [2]). More recently, a robotic arm actuator has been combined with a touch display in TouchMover 2.0 [32], which is capable of rendering both large forces in one dimension as well as haptic texture feedback via two voice-coils mounted on the display. While such devices render forces and collisions with relatively high fidelity, they sacrifice mobility and offer very restricted operating space. In particular, haptic arms with a physical earth reference such as the PHANToM device are well suitable for teleoperation of robots or tele-surgery where the user is stationary [5].

Large forces can also be rendered to the hand via the use of glove-based exoskeletons [6, 10]. CyberGrasp glove [10] uses five tendon actuators routed to the fingertips via the exoskeleton to add resistive force to each finger and prevent the user's fingers from penetrating a virtual object. The Rutgers Master II-ND [6] is a haptic glove that uses pneumatic actuators between the palm and the fingers to render large grasping forces to the fingertip.

Our NormalTouch device is closest to the Marionette [21], which uses tilt-platforms to convey the surface normal underneath four fingertips while the user is moving the device like a mouse. In contrast to the Marionette, our work focuses on handheld 3D interactions in VR scenarios. We were inspired by the effectiveness of tilt platforms in conveying the surface normal underneath the fingertip [1, 38].

In contrast to the actively actuated haptic devices, *passive haptics* can also be used to provide highly realistic haptic sensations in VR. With passive haptics, a stand-in physical proxy object may provide appropriate haptics for a rendered virtual object. For example, Azmandian et al. [3] recently demonstrated how a user can be redirected to reuse the same passive physical proxy for multiple virtual objects.

Wearable and Mobile Haptics

It is well understood in the haptic literature that the stimuli experienced by the hand when holding or exploring the shape of an object has both kinesthetic and cutaneous components [14]. Kinesthetic feedback requires larger actuation forces and provides the user with information about the relative position of the parts of their body (e.g., joints). Cutaneous information is felt by the pressure receptors in the skin and is a direct measure of the direction and intensity of contact forces as well as texture.

Other than the previously mentioned exoskeleton gloves, most of the wearable haptic actuators in the literature focus on rendering cutaneous stimuli. Prattichizzo et al. [28] offer a wearable haptic device that uses a three-string actuated platform capable of rendering cutaneous forces at the fingertip. Choi et al. [9] have developed a haptic actuator as a wearable Braille display based on dielectric elastomer which can be manufactured on a flexible substrate and wrapped around the fingertip. Similarly, Brewster and Brown [7] produced small wearable "Tactons", capable of rendering non-visual messages through Braille-like miniature pins. Velasquez et al. provide a comprehensive survey of similar haptic technologies targeted at the blind population [35]. In contrast, our devices are not wearable, but held in the hand, tracked with 6 DOF, and able to provide both kinesthetic and cutaneous feedback.

Cutaneous feedback has also been explored in mobile device form factors. Luk et al. [24] present a handheld haptic display platform based on the concept of lateral skin stretch. Hemmert et al. [15] applied shape changing and weight shifting in conceptual mobile devices to convey the sense of direction during interaction with the device.

UltraHaptics [8] demonstrated another related technology for providing ultrasonically created haptics to the hand in mid-air without the need for the user to hold a device. While promising technology, the sensations created are subtle and the working area is very restricted.

Tactile Array Displays

Our TextureTouch device builds upon the rich history of Tactile Array displays which use an array of electro-mechanically actuated pins/rods to render a dense tactile surface [12, 18, 27, 29, 36]. For example, the Exeter touch array [33] used piezo actuators to move 100 small pins in a 1.5cm square area underneath the fingertip. Lumen device used a coarser 13x13 array of illuminated rods to explore on-demand UI elements [29]. inFORM [12] explored the affordances of shape and object actuation when using a larger tactile array display with 30x30 actuated "pixels" that cover the area of approximately 15 sq. inches. Recently, Jang et al [19] used a single dimensional actuated tactile array integrated along the edge of a conceptual smartphone to convey haptic notifications.

Closest to our TextureTouch are solutions that mounted a tactile array display on a stylus. For example, UbiPen [22] had a tactile array on a stylus that added texture in addition to vibrotactile feedback when using the pen on the tablet. Kim at al. attached a similar stylus to a PHANToM device [25] for a palpation simulation application [20]. In contrast to that work, with TextureTouch we explored the interaction capabilities in a much more free-movement VR scenarios, where a tactile array is integrated in a highly movable, 3D tracked handheld controller.

HANDHELD CONTROLLERS FOR 3D SHAPE OUTPUT

The design goal of our haptic controllers is seamless use in a virtual environment. As such, they satisfy three requirements: 1) deliver 3D shape output in a handheld form factor, 2) a compact and lightweight form factor to facilitate unencumbered mid-air operation, and 3) provide a human-scale force in rendering 3D shapes for both cutaneous (i.e., haptic sensations on the finger surface) and kinesthetic feedback (sensation of actuating and displacing the finger).

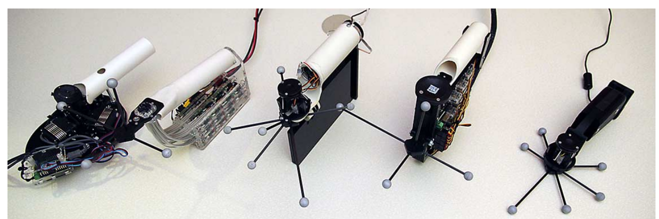

Figure 2: Five device prototypes (three NormalTouch versions and two TextureTouch versions) that we produced in our iterative process. Each design built on the learnings from the previous version, improving weight, robustness and mobility.

	NormalTouch	TextureTouch
Basic principle	Tilt platform	Tactile array
Height dynamic range	2.6 cm	1.4 cm
Rendering resolution	1 moving platform	4×4 pin array
Actuated area	3.3 cm diameter disk	1.3×1.3 cm^2
Maximum rendered angle	45 degrees	90 degrees
Weight	150 g	600 g
Dimensions (excl. markers)	7×20×5 cm	17×18×5 cm
Finger force sensing	Yes (FSR)	No

Table 1: Comparison of haptic shape controller properties.

We identified two promising technologies that meet these goals: tilt platforms [1, 21, 38] and tactile arrays [12, 18, 27, 29, 36]. While tilt platforms better render surface normals and are simpler to implement, tactile arrays render features smaller than the user's finger using individual pins. Through an iterative design process of the prototypes shown in Figure 2, we implemented two fully functioning prototype controllers, each one built around one of these two core technologies. Table 1 summarizes the main properties of each of our two haptic shape controllers: NormalTouch and TextureTouch. For illustration purposes in the descriptions below, we assume that the user moves their finger and the controller in a virtual scene with a variety of 3D objects. In the virtual environment, the user's hand is represented by a 3D hand model with matching 3D positions and 3D orientations.

NormalTouch: A 3D Tiltable and Extrudable Platform

As shown in Figure 3, the core of NormalTouch is an acetal (Delrin) platform that is actuated by three servo motors. A force sensor inside the disk detects touch input at a range of forces. The handle of the controller encloses all electronics, including the motor controller. The small retroreflective spheres mounted around the motors serve as markers to track NormalTouch in 3D with surrounding cameras.

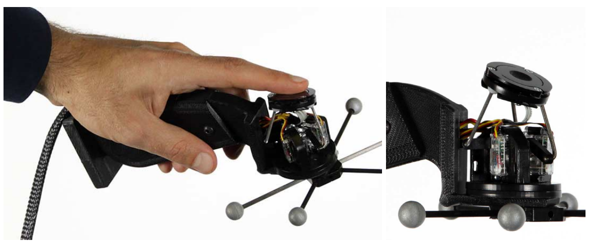

Figure 3: (left) NormalTouch during interaction. (right) Close-up of the tiltable and extrudable platform.

When moved around in the virtual scene and making contact with virtual objects in the scene, NormalTouch replicates the surface normal of these objects. NormalTouch's default state is a fully retracted platform. As soon as the user makes contact with a virtual object, NormalTouch tilts its platform to the relative 3D orientation of the object's surface and extrudes the platform according to the user's movement of the controller in the physical space. This causes the user's finger to remain in the same 3D position—outside the virtual object's boundary, which is registered in the physical space as shown in Figure 4.

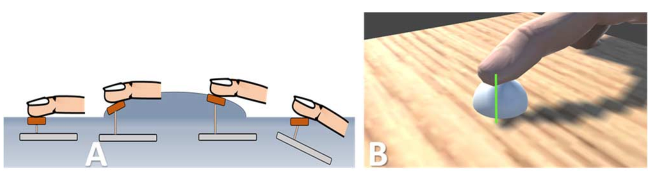

Figure 4. (A) Illustration of NormalTouch operation while rendering the surface of a 3D virtual object (gray). (B) Device height depicted in our 3D scene. Device's height and angle change to faithfully render the surface at the point of touch.

The core components of NormalTouch are the three servo motors that impart the mechanical three-dimensional freedom of the platform. We used three Hitec HS-5035HD nano servos arranged in a 3-DOF Stewart Platform as shown in Figure 3b. The servos are connected from the servos' control arms with revolute joints, through small rigid linkages to ball-and-socket spherical joints under the platform. The rigid linkages are restricted in movement to be always perpendicular to the servo's axis. This allows the three degrees of freedom imparted by the three servos to be mechanically transformed to the finger pad's yaw and pitch angles plus linear movement along the roll axis (towards and away from the user). All components are designed in CAD and mostly laser cut in Delrin plastic. An advantage of our configuration is that the overall 3D mechanism occupies a minimum volume compared to other implementations.

To control the servos, we integrated an off-the-shelf multi-servo USB controller (Pololu.com Mini Maestro-12) into a 3D printed controller handle. NormalTouch draws 375mA in average use (620 mA peak current). When our device is outfitted with a 3000mAh LiPo rechargeable battery for wireless operation, it yields ~8 hr battery life. The controller also senses analog voltages, in our case to detect force.

Force Sensing

NormalTouch senses force input using an off-the-shelf force transducer (Interlink Electronics FSR-402) in an end effector. We chose this implementation rather than sensing motor current because the latter can cause compromises from gear and bearing friction. The force sensor is a 13 mm disk using force sensing resistor material with electrodes, detecting forces between 0.2–20 N, which is adequate for our use.

The sensor is configured such that with applied force levels of less than 0.2 N, one of the two electrodes in the sensor is not in contact with the FSR material and results in infinite resistance and no voltage to the ADC, allowing us to reliably detect moments during which no touch is present. A small force applied to the sensor (~0.2 N) results in electrode contact and a reliable force reading.

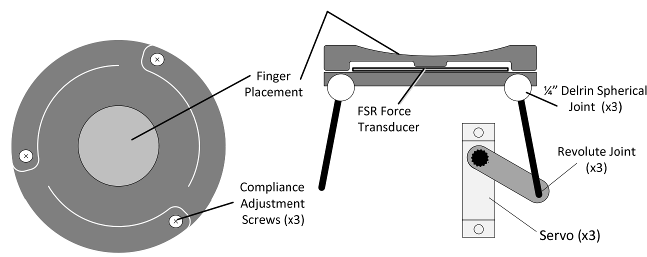

Figure 5: Design of NormalTouch's force-sensing platform.

To overcome this initial non-linearity and increase the low force sensitivity, the platform is composed of two separate Delrin-cut layers (Figure 5). In the top layer, the finger touches a smooth depression that we added for finger placement. The border of the platform disk is cut into a partial three-legged spiral spring to allow for an adjustable and compliant preload on the force sensor housed in the bottom disk.

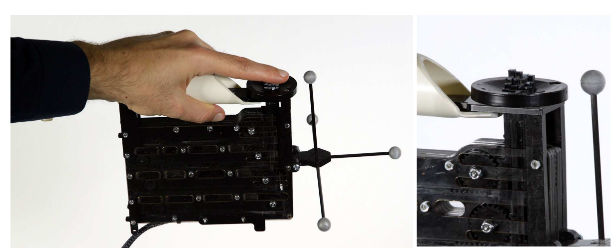

Figure 6: (left) TextureTouch showing the 16 servo motors and gear assembly. (right) Close up of the 4×4 array of pins.

TextureTouch: 3D Pixel Shape Output

As shown in Figure 6, TextureTouch packs a 4x4 actuated pin array as the primary method for haptically rendering 3D shapes and structures onto the user's finger. All electronics are attached to the side of the device. Similar to NormalTouch, small retroreflective spheres mounted to the base serve as markers for tracking the controller in 3D space.

TextureTouch behaves similarly to NormalTouch during use, except that this time 16 virtual probe lines detect contact with the surfaces of virtual objects in the scene as shown in Figure 7. This individual probing enables TextureTouch to detect fine-grained surface structure and relay that to the extrusion of the individual pins for the user to feel on their finger. Similar to NormalTouch, when the user's finger is outside all virtual objects, all pins in TextureTouch's array are fully retracted and the finger rests flat on the platform.

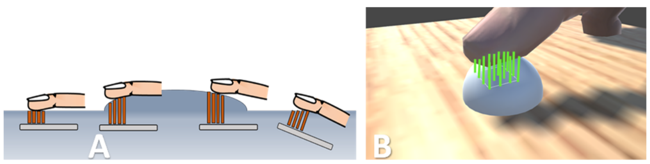

Figure 7: (A) Illustration of TextureTouch operation while rendering the surface of a 3D virtual object (gray). (B) Device pin heights depicted in our 3D scene.

TextureTouch comprises 16 linearly actuated adjacent pins in a 4×4 configuration. Each pin is individually driven by a small servo motor (HiTec HS-5035HD). We used rack and pinion mechanisms to convert the servos' rotary output to linear travel. An additional rack and pinion pair turns the motion at right angles for an optimized configuration and minimum volume as shown in Figure 8. A Pololu Mini Maestro-24 servo controller relays the extrusion levels determined by the virtual reality system from the PC to each servo motor. TextureTouch draws 800mA in average use (1.5A peak).

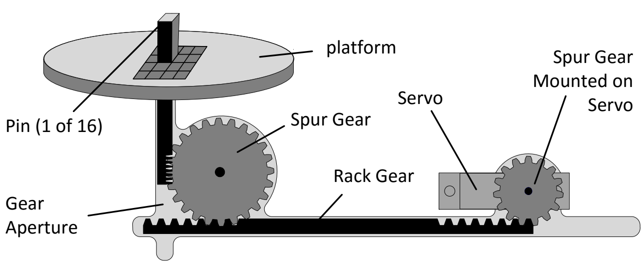

Figure 8: TextureTouch actuation mechanism for a single pin.

INTEGRATION OF HAPTIC CONTROLLERS INTO VR

The basic concept of haptic shape rendering in simulated 3D environments is a well understood topic [31]. In principle, one determines the collisions of the haptic proxy object (in our case a 3D fingertip) with 3D virtual objects in the scene, computes the resulting forces, and renders an equal and opposite force on the haptic device. This works well if the haptic device is stationary and capable of exerting enough force back to the user to prevent the user from moving further (otherwise the device simply gives up).

However, haptic shape rendering on a *handheld* device is more complicated, because both the platform and the actuated point are moving and held by the same hand. Therefore, any force our controllers render at the user's fingertip will inevitably be felt against the rest of the palm which holds the handle of the controller. In practice, however, the fingertip's sensitivity to kinesthetic and cutaneous forces is much higher than the sensitivity of the rest of the hand [28], rendering the actuation experience convincing. See our last experiment below for an evaluation of rendering fidelities.

In their most basic operation, both NormalTouch and TextureTouch operate on the same principle. When the tracked controller penetrates the surface of a virtual object, the controller's articulation point(s) extends to compensate for the penetration, thus rendering the surface in contact (Figure 4 and Figure 7). NormalTouch has a single extension platform and we additionally orient the platform to relay the surface normal at the collision point. TextureTouch individually performs this calculation for every one of its 16 pins.

To further support the haptic sensations with visual feedback, we animate the joints of the virtual finger when touching virtual objects to signal to the user that a collision has occurred (Figure 9).

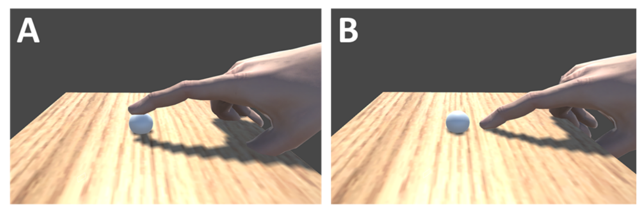

Figure 9: The joints of the virtual finger are animated to reflect contact with an object and match the haptic actuation.

Handling Surface Penetration

Since our devices do not have a physical earth reference, it is impossible to prevent the user from penetrating virtual objects in the scene with their controller hand. How the device behaves when the dynamic range of its actuator is exhausted and the fingertip penetrates the surface can have significant impact on the quality of the experience.

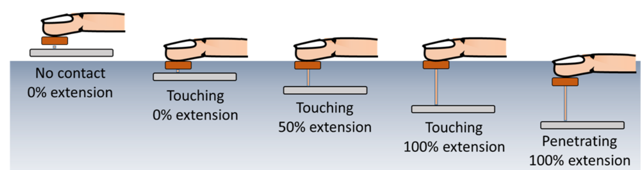

Figure 10: Illustration of basic height rendering for NormalTouch. When in contact with the surface, only the controller base keeps moving and the finger remains at the same location within the dynamic range of the device. Note: the same behavior applies to each pin in TextureTouch.

We deliberately chose to keep the platform fully extended upon complete penetration (Figure 10). While this does not give the user a clear signal that they have penetrated the surface, it renders a more consistent behavior. Alternatively, retracting the platform upon penetration (i.e., rendering 0N force) frequently results in undesirable behavior. The explanation for this is that, most of the time, penetration was not intended, and the user will correct their behavior and retract their hand back to the surface. This produces strong oscillations between platform's full extension and no extension which is highly confusing and undesirable.

Penetration Compensation

In our pilot evaluations, we noticed that penetration with virtual objects was almost never a desirable goal. In most cases, users wanted to touch the object and penetrating it "broke" the experience. To prevent it, we implemented a *penetration compensation* technique which effectively decouples the real position of the controller (reported by the OptiTrack tracker) and the virtual position of the hand in the scene (Figure 11).

Penetration compensation dynamically offsets the fingertip (and the hand) along the vertical axis of the device, which effectively keeps the finger on the surface of the virtual object regardless of how deep it penetrates. In practice, we limit the penetration compensation to 20 cm distance and temporally smooth the depth. This makes it relatively easy to move the hand around and explore the object surface. Rather than waiting for full extension, we apply penetration compensation once the platform reaches 75% of its dynamic range to ensure that there is some dynamic range left to adequately render surface variations in height and normal (Figure 11).

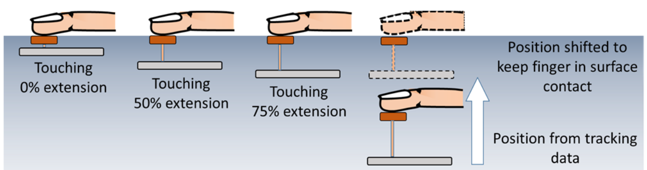

Figure 11: Our *penetration compensation* technique stops the 3D virtual hand at the surface of the virtual object to facilitate easier exploration. Dashed lines indicate the shifted hand position (offset from the tracked position of the real hand).

VR Application Scenarios

All our haptic and visual rendering was performed in Unity game engine (version 5.3.2). We used Oculus Rift DK2 head mounted display which was tracked by its own Oculus camera. Each handheld device was configured with a unique cluster of retro-reflective markers (Figure 2) and was tracked via OptiTrack V120:Trio tracking system (optitrack.com). Our OptiTrack system was calibrated to report in the same coordinate system as the Oculus Rift and all components were rigidly mounted to eliminate the need for recalibration.

OptiTrack system was configured to report the pose of the center of the platform of each device (which for NormalTouch includes a lowered area for the finger to rest). OptiTrack reports a mean tracking error of <1mm for our controllers (which are ~1m from the cameras during use). We instructed participants to rest their finger pad on the platform center and the 3D VR hand was rendered accordingly (when the platform moves, the virtual fingertip moves as well). While currently not implemented, the location of the fingertip on the platform could be tracked for extra precision (e.g., using a simple 3x3 capacitive touch array).

We implemented several VR scenarios to test the effectiveness of our devices. We explored rendering a variety of rigid and deformable 3D objects, such as simple shapes, as well as 3D models of cars, animals, etc. (Figure 1d and Figure 12a). We also experimented with rigid body physics simulations (Figure 12b). In this scenario, the user could use force sensing and feedback on the NormalTouch device to flick a ball across the table.

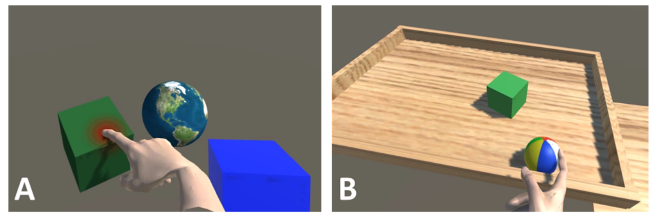

Figure 12: VR scenarios we explored: (A) playing with rigid and deformable objects, (B) interacting with a rigid body physics simulation.

EVALUATION

We conducted a three-part user evaluation to determine the extent to which our 3D shape-output prototypes increase the level of fidelity for users in virtual-reality environments. We compared the performance of our two prototypes TextureTouch and NormalTouch to two baseline interfaces: a controller with vibration-only output and a visual-only condition with no haptic feedback.

To assess the fidelity of haptic shape rendering using each controller, participants completed two types of targeting tasks: a pointing and a tracing task [39]. Both tasks assess how accurately participants match the visual stimuli with haptic sensations, and thus how much haptic 3D shape output aids them in interacting with virtual 3D objects. In a third task, participants explored the shape of virtual objects using each of the controllers to rate the level of fidelity of haptic shape rendering each of the interfaces provides.

Interfaces

Two devices shown in Figure 13 were used to complete the tasks and we tested four interfaces: *NormalTouch*, *TextureTouch*, *VibroTactile* and *VisualOnly*. To ensure the comparability of all four interfaces, we modified a NormalTouch device with five additional layers of acrylic to exactly match the weight, balance and shape of the TextureTouch device (see Figure 13). To enable *VibroTactile* interface, we further modified the NormalTouch device and incorporated a vibration motor extracted from an Xbox controller. In summary, this NormalTouch variant served as the device for three interfaces: *NormalTouch, VibroTactile* and *VisualOnly*.

Both, *TextureTouch* and *NormalTouch* rendered the 3D haptic shapes of virtual objects according to the position of the user's finger in the virtual environment. The *VibroTactile* interface activated the vibration motor whenever the bottom of a participant's finger was within +/- 2mm of the surface, an experience similar to haptic vibration feedback in game controllers (e.g., Forza racing game with Xbox One controller). This simple vibration scheme (vibrate when in contact with the surface) was chosen as a good baseline since it is a widely-deployed behavior of current consumer VR controllers (e.g., HTC Vive).

Figure 13: Devices used in the evaluation. (left) *TextureTouch* and (right) a modified *NormalTouch* controller, used for the NormalTouch interface, the *VibroTactile* interface (using an Xbox vibration motor), and the *VisualOnly* interface that tracked the user's finger without haptic feedback.

In *VisualOnly*, the system provided no haptic output or additional visual feedback of any kind, beyond seeing their finger penetrate the surface of the object. Lastly, in the *VibroTactile* and *VisualOnly*, participants used a NormalTouch controller with a fully retracted platform.

For all interfaces, participants wore an Oculus Rift DK2 headset to see virtual objects and a representation of their hand in the virtual space. Participants were standing throughout all trials, holding the controllers in the space in front of them next to a table as shown Figure 14, and kept their head in the same position throughout.

Figure 14: A participant during the experiment. The Oculus system tracked the participant's head. Both haptic shape controllers were tracked by an Optitrack tracking system.

Tasks

Targeting Accuracy Task: During this task, participants were repeatedly presented with one of the three target types shown in Figure 15. For each target type, crosshairs on top of the object highlighted the target. The three target types encompass features in virtual objects that are (a) smaller than a user's finger, (b) within the dimension of a finger, and (c) substantially larger than a finger. To start a trial, participants grabbed the controller in their dominant hand, held it into a virtual 'off space' and pressed a button on a presentation clicker with their non-dominant hand. This triggered the virtual object and crosshairs target to appear. Participants then positioned the controller and thus their finger in VR as accurately as possible to acquire the target and pressed the clicker to confirm the location, which completed the trial.

We logged the participant's 3D finger positions throughout each trial along with their timestamps as well as the orientation of the controller and their head position and orientation. We also logged the duration of each trial from clicking in the off space until target acquisition.

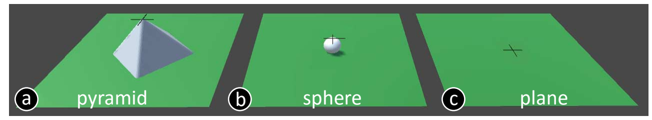

Figure 15: Haptic targets for accuracy tasks: haptic features that are (a) smaller than, (b) within the dimensions of, and (c) substantially larger than a human finger.

Tracing Accuracy Task: During this task, participants followed a 3D path with their fingertip as accurately as possible. Tracing paths included two straight lines and two curved lines, once appearing flat in front of the participant and once rotated at a 35° angle (Figure 16). Participants started each trial by holding a controller in their dominant hand, moving it to the green cone, which indicated the beginning of a path, and pressing the clicker. Participants then traced the path to the red sphere and pressed the clicker to complete the task.

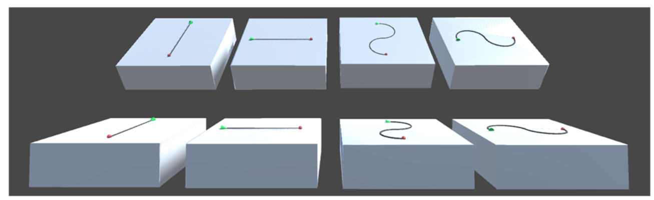

Figure 16: Tracing paths. We compared straight and curved paths in two directions from two perspectives. A green cone marked the beginning of a path, a red sphere indicated its end.

Similar to the first task, we logged all timestamped 3D finger positions throughout each trial as well as the orientation of the controller and participants' head positions and orientations. We also logged the duration of each trial between pressing the clicker on the start and end of the path.

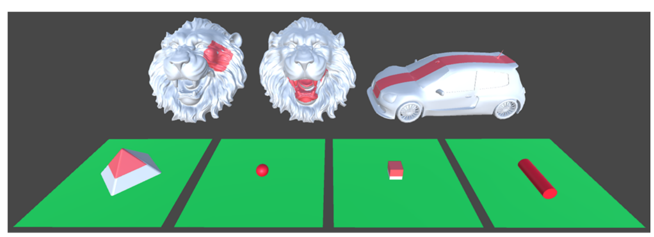

Figure 17: Participants assessed the fidelity of haptic rendering of this 7 objects using each of the three haptic interfaces.

Fidelity Assessment Task: During this task, participants pressed the clicker, saw each of the high-quality 3D models shown in Figure 17, and explored their shape and surfaces through touch by moving around the controller in the areas marked in red. We limited each trial to 20 seconds to compare participants' impressions of rendering quality and fidelity in each of the interfaces. After completing each targeting and tracing task, participants verbally rated how well the haptic rendering matched their visual impressions of the virtual object on a 5-point Likert scale, ranging from 1 (mismatch) to 5 (accurate match). We also recorded each participant's verbal comments. For this task, participants did not use the *VisualOnly* interface, since it obviously did not present any haptic rendering.

Procedure

Before the study, the experimenter explained the purpose of our high-fidelity haptic output controllers to each participant and demonstrated each of the four interfaces in a static virtual reality environment. Participants then put on the Oculus Rift headset and experienced a static scene using each haptic interface to familiarize themselves with our controllers as well as the baseline interfaces. Participants then performed a series of targeting and tracing tasks for training purposes. On average, training took 15 minutes per participant.

Each participant completed all three targeting tasks with ten repetitions using each of the four interfaces ($3 \times 10 \times 4 = 120$ trials) and all eight tracing tasks with two repetitions using each interface ($8 \times 2 \times 4 = 64$ trials). Trials were randomized across participants to account for sequence effects. On average, participants completed the experiment in 50 minutes.

Participants

We recruited 12 right-handed participants (4 female), ages 24–56 from our institution. 6 participants had never tried any VR system, 5 participants had tried on a VR headset once before, and 1 participant used HTC Vive on a weekly basis. Participants received a small gratuity for their time.

Hypotheses

We hypothesized that haptic feedback would increase the level of fidelity of perceiving virtual objects, resulting in:

H1. Haptic feedback leads to more accurate targeting and tracing compared to *VisualOnly* feedback.

H2. *NormalTouch* and *TextureTouch* allow targeting with higher accuracy than *VibroTactile*, because they render 3D shapes with higher fidelity, facilitating precise touch.

H3. *TextureTouch* produces the lowest error overall, because it renders structure on the participant's finger as opposed to just the surface normal.

H4. Participants complete trials fastest in the *VisualOnly* condition, because no cues other than visual need cognitive attention and time to process.

Results

Targeting Accuracy Task: We ran a one-way repeated measures ANOVA on mean error distance from the crosshairs target in the completed trials with participant as the random variable. We found a significant main effect on error distance ($F_{3,9} = 11.284$, $p < .002$) for $\alpha = .05$. With post-hoc t-tests using Bonferroni-adjusted confidence intervals, we found three significant differences: Participants produced a lower mean error distance using a haptic controller (i.e., *NormalTouch*, *TextureTouch*, *VibroTactile*) than when receiving *VisualOnly* feedback (all $p < .02$) as shown in Figure 18a.

We now break down the error distance into the error in the plane of the target (*x/y* error) and the error from the plane (*z* error). While we could not find a significant main effect of interface on average *x/y* error (Figure 18b), we found a significant effect on average *z* error ($F_{3,9} = 24.596$, $p < .001$). Post-hoc t-tests using Bonferroni correction showed significant differences between each haptic interface and *VisualOnly*, respectively, as well as between *NormalTouch* and *VibroTactile* (all $p < 0.015$). As illustrated in Figure 18c, *VibroTactile* feedback reduced the *z* error by 37% compared to *VisualOnly* feedback, whereas *TextureTouch* reduced it by 58% and *NormalTouch* by 65% to 1.4 mm compared to 4 mm for *VisualOnly* feedback.

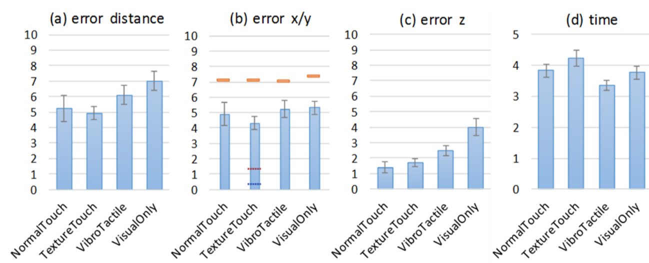

Figure 18: 3D targeting error (in millimeters), error in the plane of the target (x/y), error from the plane of the target, and average completion time (in seconds). Red indicators show minimum radiuses for reliable touch targets [16].

The red bars in Figure 18b illustrate the minimum target sizes required for reliable target acquisition modeling the *spread* of input using each of the four types of feedback [16]. While we could not find a significant main effect of interface on minimum target size, we see that the minimum target sizes for interfaces that provided haptic feedback tend to trend lower than the minimum size for *VisualOnly* feedback.

Given that *TextureTouch* features a spatial component for rendering features unlike the other three interfaces, a feature may have been rendered on the controller and been haptically noticed by participants even though their finger did not perfectly align with the virtual target. The dotted blue line in Figure 18b reflects the drop in input error below 1mm under this consideration. Similarly, the dotted red line represents the spread of minimum reliable targets of 1.3mm.

We also ran a one-way ANOVA on mean completion time and found a significant main effect ($F_{3,9} = 9.586$, $p < .004$). Post-hoc t-tests using Bonferroni correction showed a significant difference between *VibroTactile* and *VisualOnly* ($p < .006$) as shown in Figure 18d.

Tracing Accuracy Task: For all of participants' tracing trials, we computed for each point on their trace the closest point on the target trace for all data points between the green cone and red sphere. A one-way ANOVA on mean error distance found a significant main effect of interface ($F_{3,9} = 27.729$, *p*

< .001). Post-hoc t-tests using Bonferroni-adjusted confidence intervals showed significant distances between all haptic interfaces and *VisualOnly* as well as between each of our two controllers and *VibroTactile* (all $p < .02$). While *NormalTouch* and *TextureTouch* decreased the error distance by 15% and 18%, respectively, *VibroTactile* incurred a 2% error increase compared to *VisualOnly* feedback (Figure 19a).

Breaking this down to the error in the plane of the trace, a one-way ANOVA on error distance found a significant main effect ($F_{3,9} = 12.154$, $p < .002$). Post-hoc t-tests using Bonferroni correction showed a significant difference between *TextureTouch* and *VibroTactile* as well as *TextureTouch* and *VisualOnly* (both $p < .007$) as shown in Figure 19b.

Regarding the z error, i.e., the distance to the plane of the trace, a one-way ANOVA found a significant main effect ($F_{3,9} = 16.816$, $p < .001$). Post-hoc t-tests using Bonferroni correction showed significant differences between each of our two controllers (both $p < .002$) and *VibroTactile* as well as between *TextureTouch* and *VisualOnly* ($p < .03$). Compared to *VibroTactile* feedback, both our controllers reduced the average tracing z error by ~26% as shown in Figure 19c.

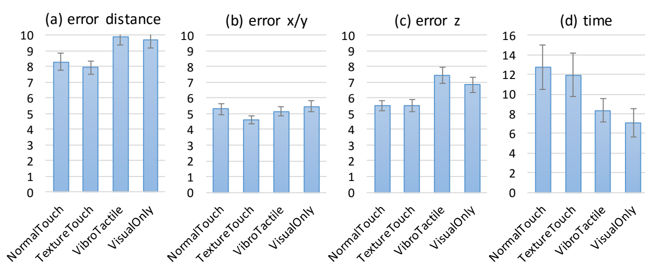

Figure 19: Mean 3D tracing error (in millimeters), mean tracing error in the plane of the path, mean error from the plane of the trace, and average completion time (in seconds).

We also ran a one-way ANOVA on tracing time and found a significant main effect on interface ($F_{3,9} = 12.603$, $p < .001$). Post-hoc t-tests using Bonferroni connection showed that completion time using our two controllers was significantly different to *VisualOnly* feedback. *NormalTouch* incurred a slowdown of 81%, while *TextureTouch* accounted for an 70% delay in completion (Figure 19d).

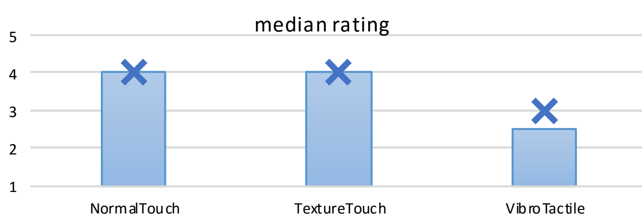

Figure 20: Median Likert ratings of controllers the fidelity rating task. Blue crosses indicate the mode of ratings.

Fidelity Assessment Task: We ran Wilcoxon rank-sum tests using Benjamin-Hochberg-adjusted confidence intervals to compare the results of participants' ratings of haptic fidelity. Participants' ratings of each of our controllers were significantly different from their rating of *VibroTactile* (both $p < .03$). As shown in Figure 20, the median rating for both *NormalTouch* and *TextureTouch* was 4 out of 5, whereas *VibroTactile* received a median rating of 2.5. Figure 21 shows histograms of the ratings for objects with large faces (cube, cylinder, sphere, car) and smaller features (lion's mane, teeth, and pyramid). To understand participants' ratings, we transcribed their comments during this part of the study.

Discussion
The experiment supported our first hypothesis in that all haptic controllers offered accuracy benefits in acquiring a virtual target. Unsurprisingly, the effect is largest for the distance to the surface; all haptic interfaces precisely indicate the moment the virtual finger makes contact with the surface. This effect was most pronounced in both our controllers compared to vibrotactile haptics, because they offer a dynamic range for virtual surfaces: participants not just felt the contact with the virtual object, but could in addition use the extrusion height of our controllers to precisely position their finger, which supports our second hypothesis.

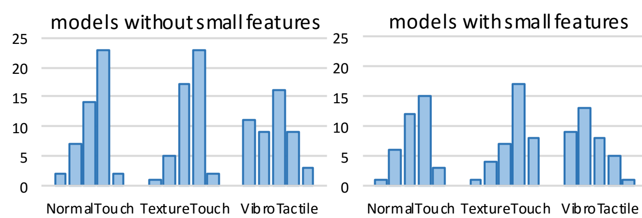

Figure 21: Histogram of median ratings for (left) flat surface objects (cube, cylinder, sphere, car) and (right) objects with small features (two lion parts, pyramid).

We found no significant difference in accuracy between NormalTouch and TextureTouch and thus no support for our third hypothesis; to align the center of the actuated platform with the virtual target, apparently, participants did not additionally benefit from TextureTouch *directly* rendering features on the finger surface whereas they had to move NormalTouch around to "feel" the top of the sphere, for example. We can see from Figure 18b that the finger positions in the plane of the target contribute most of the overall error.

Comparing the results of errors in the plane of the target to previous findings, we see that targeting errors are in the same range: The red lines in Figure 18b indicate the minimum sizes for 'reliable buttons' (targets that capture 95% of all input) at a magnitude of 7–8 mm, which compares to the 8.2mm diameter of minimum reliable targets in related touch studies (when controlled for the finger's yaw, pitch, and roll angle [16]), which the task in our study loosely included. This also validates the quality of our measurement apparatus because the average error we observed is comparable to that reported on traditional 2D touchpads. At the same time, this suggests that despite the use of haptic controllers, targeting remains a dominantly *visual* operation, which compares to research on (finger) touch accuracy for physical targets [17].

Considering that TextureTouch renders haptic features as a much higher resolution than the three other interfaces, one would expect to observe much lower errors on TextureTouch during targeting tasks, as the haptic rendering alleviates occlusion problems. Therefore, we reanalyzed the input data to account for situations in which TextureTouch rendered the haptic target onto the user's finger, but itself was not *centered* on the virtual target. In these situations, participants were able to feel the target on their finger and, as we had observed in pilot studies, often *shifted* their finger on the pin array. TextureTouch thus moved parts of the virtual targeting task into the physical world and, after an initial ballistic phase, turned the task into a physical targeting task. Assuming that participants moved their finger on the platform, something TextureTouch is currently not capable of tracking, the average x/y targeting error drops below 1mm as shown by the dotted blue line in Figure 18b. Along with it, the minimum button size for reliable virtual targets drops to 1.3mm. Further studies are required to determine the true accuracy of TextureTouch for targeting tasks by additionally tracking participants' finger locations on the pins.

Interestingly, the *VibroTactile* interface showed no speed-accuracy tradeoff in the study. Participants were significantly faster and significantly more accurate in targeting with the *VibroTactile* haptic feedback than with *VisualOnly* feedback, which does not support our fourth hypothesis. One reason could be that participants targeted visually, but confirmed each trial as soon as they felt the vibration and thus contact. We assume that while our two haptic controllers provided similar feedback, participants spent additional time trying to feel the feature in the plane of the target.

In part, the tracing task echoed the findings of the targeting task: Participants traced paths significantly more accurately using either *NormalTouch* or *TextureTouch*, showing that haptic shape feedback significantly increases the accuracy for this task and supporting H1. Somewhat surprisingly, *VibroTactile* feedback was counterproductive to accurate tracing, most likely because participants obtained a false sense of 'being in touch' with the object. Indeed, Figure 19c reveals that a large part of the *VibroTactile* error stems from the distance to the plane. Interestingly, participants were significantly more accurate using *TextureTouch* than in the visual condition, partially supporting H2 and H3. We attribute this to the size of the path ridge, which could be felt in the individual pixels, but produced more of a jittery behavior in *NormalTouch* as participants scrubbed across the ridge.

TextureTouch's added accuracy came at a cost of speed, as participants completed the trials significantly faster in the visual condition. Only *NormalTouch* was slower, presumably because participants tried to precisely locate the ridge. Participants' average speed using the *VisualOnly* interface supports H4. The fact that this speed came at no substantial loss in accuracy is explained by a participant's quote: "To be extra accurate, I'm making sure that the trace always passes through my fingernail at all times and hold it steady." This observation, too, reflects findings in the related work, where participants have been reported to use visual features on their finger to align them with targets for added accuracy [17]. However, this accuracy comes at a cost of breaking the physical realism, as the user's virtual 3D finger may occasionally penetrate the solid surface of the virtual object.

The fidelity task produced some interesting insights. As evident in the histograms, participants preferred *TextureTouch* for objects with fine-grained detail and *NormalTouch* for touching objects with large faces, which shows the complementary nature of both controllers. On the other hand, *TextureTouch* received lower ratings when participants explored smooth surfaces, as our controller produced seemingly noisy haptic signals. Interestingly, the noisy actuation was a result of participants' hand motions, rendering resolution and naturally occurring jitter, since algorithmically, *TextureTouch* renders virtual shapes 1:1 with no additional processing.

Participants frequently commented that even *NormalTouch* produced a sensation of surface features that matched the visual structure, such as the lion's mane. Interestingly, participants seemingly integrate shape over time by moving the controller around and thus obtain a sense of changes in shape or even the surface structure, despite the fact that the controller renders merely a *state* at any given point in time. Participants' responses to the lion's mane proved particularly insightful; even though the platform on *NormalTouch* was simply wiggling as participants moved their finger across it, they attributed high levels of rendering fidelity to the device even for minute structures. One explanation for this is that the visual channel may dominate the overall impression; users expect to feel high-frequencies in the texture and the controller produced them as they scrubbed across the surface.

The *VibroTactile* interface resulted in mixed ratings across participants. Some participants perceived the vibration as irritating. P3 said, "It shouldn't be vibrating. A plane doesn't vibrate" and P1 told us, "It was really distracting since I was trying to be accurate, but the device kept vibrating." Others found the sensation to match touch well in contrast. P4 said, "Once I was on the surface, it was really easy to follow it with the vibration" and P2 assessed, "I liked the vibration best of all of them. It just feels like it makes the most sense." P10 commented "the structure of [the lion] is almost too complex for the simplicity of the vibration and it just feels weird", recognizing the limited dynamic range of vibration to indicate shape, especially in comparison to the higher-fidelity rendering our other two controllers performed.

Our choice of vibration behavior (vibrate when in contact with the surface) likely had an impact on the users' experience. We acknowledge that there are many other sophisticated vibration schemes possible; however, we chose this simple vibration behavior for our baseline as it is a common across many VR controllers and allows the user to relatively accurately locate a surface in space. We empirically chose a +/- 2mm activation threshold for *VibroTactile* to balance being able to accurately find the surface and maintain contact with it in motion. We have also experimented with continuously vibrating when inside a virtual object, which resulted in larger errors and disturbing on-off vibration behavior when tracing the fingertip along the surface.

It should be noted that in our user evaluation, we deliberately added 450g to NormalTouch to balance it with TextureTouch and also matched their form factors, allowing us to compare only the effects of haptic feedback, while keeping the weight and form factor the same across conditions. If the form and weight differed, factors like user fatigue and device maneuverability would likely vary between devices and participants might express different preferences.

Finally, it became apparent that the fact that our controllers are fully tracked in 3D enabled participants to explore virtual objects more comprehensively, including their back or bottom surfaces. Participants generally appreciated that they could explore structures that they could not directly see.

LIMITATIONS AND FUTURE WORK

While the results of our experiment confirm that there are clear benefits to higher-fidelity haptics on handheld controllers, much work remains to be done. Our devices' limitations have significant effects on the user's haptic perception. NormalTouch's ability to render angles, forces and heights is physically limited. Similarly, TextureTouch is bulky, complex and with limited pin resolution and height. All our devices currently emit audible noise during operation. We will continue to optimize our designs to improve the experience.

Since we only render haptics underneath the finger pad, our haptic rendering can have surprising effects, particularly when touching the edges/corners or rotating the wrist while compensating for penetration. For example, the user might expect to feel something as the side of their fingertip brushes against a virtual object, but currently our devices render nothing. A future evaluation is needed to better understand how well our haptics match the user's expectation.

Furthermore, we do not yet know how much haptic fidelity is necessary for a convincing VR experience. In fact, we have anecdotal evidence that absolute haptic accuracy might not always be necessary. For example, on several occasions we observed people trying out our devices when they were not well calibrated (e.g., NormalTouch would render a surface normal in a drastically different direction than it was supposed to). To our surprise, people often claimed that the device accurately rendered the surface when in fact it was obviously incorrect. While anecdotal, this points to the need to further evaluate whether or not it is important to precisely match the haptic rendering in order for it to be considered realistic and high fidelity.

Though the force sensing and feedback did work with the NormalTouch device, it was not explored in the study as it was not available in TextureTouch. We plan to use force sensing feature in both devices to explore non-rigid objects as well as input objects like buttons and sliders with non-linear tactile behavior such as detents. Also, as VR controllers normally have input sensors such as buttons and a touchpad, we plan on adding these to both NormalTouch and TextureTouch with a capacitance sensor array on the finger pad. We also plan to experiment with multiple force sensors around the perimeter of the touch pads to allow a static finger placement with off-axis forces to impart a sensed XY vector for input similar to a joystick.

Lastly, there is an opportunity to experiment with different material covers for our devices' actuation areas. In particular, flexible membranes on TextureTouch could physically smooth some of the noise caused by the coarse pin resolution.

CONCLUSIONS

We presented an investigation of handheld haptic *shape* output devices that provide high-fidelity 3D haptic feedback to the finger for use in virtual environments. We demonstrated instances of two haptic shape output devices. NormalTouch renders the 3D surface normal of virtual objects using a tiltable and extrudable platform. TextureTouch uses 16 individual pins, arranged in a 4×4 grid, to render the fine-grained surface details to the user's fingertip.

As VR technologies become more mainstream, there is a clear need for haptic solutions that offer more than simple buzzing and rumbling to the hand. We believe that the haptic directions we explored have a chance to become a part of the standard VR interaction vocabulary in the near future.

ACKNOWLEDGMENTS

We thank Chris O'Dowd for help with building a wireless handheld controller.

REFERENCES

1. Alexander, J., Lucero, A., and Subramanian, S. 2012. Tilt displays: designing display surfaces with multi-axis tilting and actuation. In *Proc. of ACM MobileHCI '12*.
2. Araujo, B., Jota, R., Perumal, V., Yao, J.X., Singh, K. and Wigdor, D. Snake Charmer: Physically Enabling Virtual Objects. In *Proc. of the TEI '16*, 218-226.
3. Azmandian, M, Hancock, M., Benko, H., Ofek, E., and Wilson, A.D. 2016. Haptic Retargeting: Dynamic Repurposing of Passive Haptics for Enhanced Virtual Reality Experiences. In *Proc. of ACM SIGCHI '16.*
4. Benali-Khoudja, M., Hafez, M., Alexandre, J.-M. and Kheddar, A. 2004. Tactile interfaces: a state-of-the-art survey. In *Int. Symposium on Robotics*, vol. 31.
5. Bowman, D., Kruiff, E., La Viola, J., and Poupyrev, I. 3D User Interfaces. Addison Wesley. 2004.
6. Bouzit, M., Popescu, G., Burdea, G., and Boian, R. 2002. The Rutgers Master II-ND force feedback glove. In *Proc. of HAPTICS '02*. 145-152.
7. Brewster, S. and Brown, L.M. 2004. Tactons: structured tactile messages for non-visual information display. In, *Australasian User Interface Conference '04,* 18-22.
8. Carter, T., Seah, S. A., Long, B., Drinkwater, B., Subramanian, S. 2013. Ultrahaptics: Multi-point mid-air haptic feedback for touch surfaces. In *Proc. ACM UIST '13.*

9. Choi, H.R., Kim, D., Chuc, N.H., Vuong, N.H.L., Koo, J., Nam, J., and Lee, Y. 2009. Development of integrated tactile display devices. In *Electroactive Polymer Actuators and Devices. Proc. of SPIE* Vol. 7287.

10. CyberGrasp Glove, CyberGlove Systems Inc. http://www.cyberglovesystems.com/cybergrasp/. Last accessed April 8, 2016.

11. CyberTouch Glove, CyberGlove Systems Inc. http://www.cyberglovesystems.com/cybertouch/. Last accessed April 8, 2016.

12. Follmer, S., Leithinger, D., Olwal, A., Hogge, A., and Ishii, H. 2013. inFORM: Dynamic physical affordances and constraints through shape and object actuation. In Proc. of ACM UIST '13. 417-426.

13. Haption Virtuose 6D. http://www.haption.com/site/index.php/en/products-menu-en/hardware-menu-en/virtuose-6d-menu-en. Last accessed July 29, 2016.

14. Hayward, V., Astley, O., Cruz-Hernandez, M., Grant, D., and Robles-De-La-Torre, G. 2004. Haptic interfaces and devices. *Sensor Review*, 24, 1. 16–29.

15. Hemmert, F., Hamann, S., Lwe, M., Wohlauf, A., Zeipelt, J., and Joost, G. 2010. Take me by the hand: Haptic compasses in mobile devices through shape change and weight shift. In *Proc. of ACM NordiCHI'10*.

16. Holz, C. and Baudisch, P. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. In *Proc. CHI '10*, 581–590.

17. Holz, C. and Baudisch, P. Understanding Touch. In *Proc. of ACM SIGCHI '11*, 2501–2510.

18. Iwata, H., Yano, H., Nakaizumi, F., and Kawamura, R. 2001. Project feelex: Adding haptic surface to graphics. In *Proc. of ACM SIGGRAPH '01*. 469-476.

19. Jang, S., Kim, L., Tanner, K., Ishii, H., and Follmer, S. 2016. Haptic Edge Display for Mobile Tactile Interaction. To appear in *Proc. of ACM SIGCHI '16*.

20. Kim, S.Y., Kyung, K.U., Park, J., and Kwon, D.S. 2007. Real-time area-based haptic rendering and the augmented tactile display device for a palpation simulator. *Advanced Robotics*. 21, 9. 961–981.

21. Krusteva, D., Sahoo, D., Marzo, A., Subramanian, S., and Coyle, D. 2015. Marionette: a multi-finger tilt feedback device for curvatures and haptic images perception. In *Extended Abstracts of ACM SIGCHI '15.*

22. Kyung, K.U., and Lee, J.Y. 2009. Ubi-Pen: A Haptic Interface with Texture and Vibrotactile Display. *IEEE Comput. Graph. Appl.* 29, 1. 56-64.

23. Lee, J., Dietz, P., Leigh, D., Yerazunis, W., and Hudson, S. Haptic Pen: A Tactile Feedback Stylus for Touch Displays. In *Proc. of ACM UIST '04.*

24. Luk, J., Pasquero, J., Little, S., MacLean, K., Levesque, V., and Hayward, V. 2006. A role for haptics in mobile interaction: initial design using a handheld tactile display prototype. In *Proc. of ACM SIGCHI '06.*

25. Massie, T. and Salisbury, J.K. 1994. The PHANToM Haptic Interface: A Device for Probing Virtual Objects. *Dynamics and Control 1994.* In *Proc. of the ASME Winter Annual Meeting '94.*

26. Novint Falcon, Novint Technologies Inc. http://www.novint.com/index.php/novintfalcon. Last accessed April 8, 2016.

27. Overholt, D. 2001. The MATRIX: A novel controller for musical expression. In *Proc. of New Interfaces for Musical Expression (NIME '01).* 1-4.

28. Prattichizzo, D., Chinello, F., Pacchierotti, C. and Malvezzi, M. 2013. Towards wearability in fingertip haptics: a 3-DoF wearable device for cutaneous force feedback. *IEEE Trans. Haptics*, vol. 6(4), 506 -516.

29. Poupyrev, I., Nashida, T., Maruyama, S., Rekimoto, J., and Yamaji, Y. Lumen: Interactive visual and shape display for calm computing. *SIGGRAPH ETech '04.*

30. Rekimoto, J. 2013. Traxion: a tactile interaction device with virtual force sensation. In *Proc. of ACM UIST '13*.

31. Salisbury, K., Conti F., and Barbagli, F. 2004. Haptic rendering: introductory concepts. *IEEE Computer Graphics and Applications*, 24, 2. 24-32.

32. Sinclair, M., Pahud, M. and Benko, H. 2014. TouchMover 2.0 - 3D Touchscreen with Haptic Feedback and Haptic Texture. In *Proc. of IEEE HAPTICS '14.*

33. Summers, I.R., Chanter, C.M., Southall, A.L., and Brady, A.C. 2001. Results from a tactile array on the fingertip. In *Proc. of Eurohaptics '01*, 26-28.

34. Van der Linde, R.Q., Lammertse, P., Frederiksen, E., and Ruiterm B. 2002. The Haptic Master, a new high-performance haptic interface. *Proc. of Eurohaptics '02.*

35. Velazquez. R. 2010. Wearable Assistive Devices for the Blind. Chapter 17 in A. Lay-Ekuakille & S.C. Mukhopadhyay (Eds.), *Wearable and Autonomous Biomedical Devices and Systems for Smart Environment: Issues and Characterization*, LNEE 75, Springer.

36. Wang, Q. and Hayward, V. 2010. Biomechanically optimized distributed transducer based on lateral skin deformation. *The International Journal of Robotics Research.* 29, 4. 323-335.

37. Wellman, P., and Howe, R.D. 1995. Towards realistic vibrotactile display in virtual environments. In *Proc. of ASME,* 57, 2. 713-718.

38. Wijntjes, M. W., Sato, A., Hayward, V. and Kappers, A. M. Local surface orientation dominates haptic curvature discrimination. *IEEE Haptics* 2(2), 94-102.

39. Yano, H., Taniguchi, S., and Iwata, H. 2015. Shape and Friction Recognition of 3D Virtual Objects by Using 2-DOF Indirect Haptic Interface. In *Proc. of IEEE World Haptics Conference (WHC '15).* 202-207.