# Snake Charmer: Physically Enabling Virtual Objects

**Bruno Araujo, Ricardo Jota, Varun Perumal, Jia Xian Yao, Karan Singh, Daniel Wigdor**
Dept. of Computer Science, University of Toronto
Toronto, ON, Canada
{brar, jotacosta, varun, jiaxianyao, karan, daniel}@dgp.toronto.edu

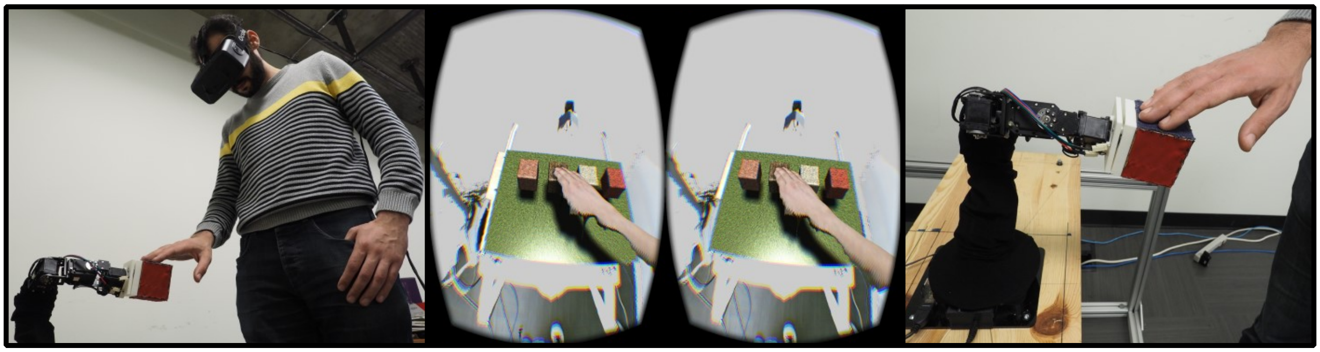

**Figure 1: A user wears a head-mounted display to render a virtual scene. When the user attempts to touch a virtual object, the robotic arm spatially aligns with that object's virtual representation and provides a physical surface, matching one or more of the object's shape, texture, and temperature, for the user to touch and feel.**

## ABSTRACT

Augmented and virtual reality have the potential of being indistinguishable from the real world. Holographic displays, including head mounted units, support this vision by creating rich stereoscopic scenes, with objects that appear to float in thin air - often within arm's reach. However, one has but to reach out and grasp nothing but air to destroy the suspension of disbelief. Snake-charmer is an attempt to provide physical form to virtual objects by revisiting the concept of *Robotic Graphics* or *Encountered-type Haptic* interfaces with current commodity hardware. By means of a robotic arm, Snake-charmer brings physicality to a virtual scene and explores what it means to truly interact with an object. We go beyond texture and position simulation and explore what it means to have a physical presence inside a virtual scene. We demonstrate how to render surface characteristics beyond texture and position, including temperature; how to physically move objects; and how objects can physically interact with the user's hand. We analyze our implementation, present the performance characteristics, and provide guidance for the construction of future physical renderers.

DOI: http://dx.doi.org/10.1145/2839462.2839484

## INTRODUCTION

*Virtual reality* has long been sought in both academic and corporate labs. In recent years, advances in pixel densities, display latency, tracking, and other innovations have brought forward a new generation of head-mounted displays. These displays have brought increasingly realistic virtual reality to consumer-level devices, and demonstrate the power of head-mounted displays for creating a sense of realism.

These efforts have largely focused on visual and auditory rendering; largely absent from these recent commercial offerings is rendering for the sense of touch. Of course, significant past efforts have been expended to attempt to further augment virtual reality with physicality, but each approach seen so far has distinct limitations. As an example, the popular *Phantom* device [21] is highly effective at producing the feeling of poking at an object with a handheld probe – which must remain in the user's hand throughout the experience. While past efforts have attempted to provide hands-free physical feedback, their reliance on air or electrical signals limits their range of sensation [15, 20, 27].

Another approach, known as *Robotic Graphics* or as *Encountered-type Haptic* interfaces [22, 32], utilizes robotic arms as actual physical objects to provide haptic feedback. They provide solid physical feedback, without requiring anything to be held in the hand. In this paper, we present *Snake Charmer*, shown in Figure 1, which builds on this past work of encountered-type haptic interfaces. Snake Charmer combines a head-mounted display (HMD) and, similar to past work, a robotic arm to dynamically simulate the physical presence of virtual content. We improve on earlier work through 3D positioning: as the user reaches into the simulation, the robotic arm tracks their movement, and takes-on the position of the object to be touched. As the user's hand, visible in their HMD, makes contact with the virtual object, they can actually *feel* it.

We make the additional contribution of simulating multiple sensations in an immersive environment. The robotic arm can pick-up and position *endpoints*: real, physical objects and textures that serve as physical proxy of virtual content. The user presence is increased by combining the physically enhanced virtual objects with a 3D live reconstruction of user arms and hands, seamlessly blending what is real and what is virtual. Thanks to the endpoints, we are able to match not only the physical location of an object's surface, but also one or more of its texture, temperature, weight, and overall shape. Moreover, by measuring user's applied force, or by detecting grip, we can signal the robot into a passive mode, allowing the user to move the object within the scene, as if picking-up real, physical objects.

In addition to simulating static objects, Snake Charmer is also able to simulate moving content, for example, an object being thrown across the scene, or a snake reaching out and tapping the user's hand. While this requires careful calibration of forces applied to the user, it can be quite effective in maintaining the suspension of disbelief, and enhance to the immersion of the virtual scene.

In this paper, we first review related work that has attempted to provide immersive virtual reality through enhanced physical feedback. We then describe how we achieve *surface synthesis* to generate physical output. Finally, we describe the overall system, present performance metrics, discuss our findings and describe future work.

## RELATED WORK

In general, we build on the rich research area of haptic output [1, 2]. While an exhaustive review of this space is out of scope, three areas are of particular relevance: virtual reality, tactile feedback to input, and hands-free haptic output.

### Haptics in Virtual Reality

Henderson and Feiner have augmented existing real objects to create opportunistic controls [11]. This approach requires interesting real-world features, and only works for augmented reality. In virtual reality, users are unable to see the real world. However, previous work suggests that, when presented with approximate physical haptics, users can withstand some variability in the haptic feedback, for virtual reality selection tasks [16].

We are inspired by early related work in what McNeely terms *Robotic Graphics*; he argues that haptic output is of extreme value and proposes the use of robotic arms (Robotic Shape Displays) as means to provide positional physical feedback [22]. Using a head-mounted display and a robotic turret holding several switches, Gruenbaum is able to simulate a 2D control panel by having the robot dynamically move the right switch into position just in time for human touch [8]. It is on this work that we build directly in the present project, by expanding beyond this simple 2D simulation of a single type of surface. In particular, to this early inspirational work, we add all of the following:

- *Simulation of 3D objects, with realtime recalculation of collision points and alignment of a commodity robot.*
- *More accurate synthesis of virtual objects via dynamic selection of available textures.*
- *Modulation of temperature to match virtual objects.*
- *Simulation of weight by allowing the user to hold a virtual object in their hand and applying a downward force.*

Additional work has explored affixing technology to the user. Kawasaki and Mouri developed HIROII, a five-fingered haptic interface. The haptic device is placed opposite to the human hand, attached to each fingertip, and tracks hand position [14]. By constraining hand movements the system effectively allows users to grasp objects in a virtual world. Tachi [29] proposes a shape display. Using a shape approximation device, the system allows a user to explore the shape of a virtual object. This system is able to simulate surfaces and edges, but it does not support texture information. Finally, Yokokohji proposes a system in which users hold a physical cube to interact with a virtual table. [32]. While each of these systems provides a degree of surface synthesis, the requirement that the user attach something to themselves or hold a fiducial is limiting. Further, Snake Charmer's ability to dynamically change the surface which makes contact with the user allows us to better simulate a wider variety of textures, and even temperature. While temperature have been explored previously in virtual reality [33], this was only achieved for a single shape: a drink cup that could be configured as hot or cold. We integrate temperature sensory stimulation with an encountered-type haptic interface that is able to simulate multiple shapes.

### Input Devices and Tactile Feedback

A number of different approaches have been used to provide haptic feedback at different scales to users while giving input via a specially-equipped input device. The *PHANToM* [21] is a six degree of freedom force feedback device that has been heavily adopted by the virtual reality community. For example, it is utilized in the Ikit Visual Haptic Workbench [12] to support the visualization of volumetric data on a stereoscopic display, and it is used to simulate realistic fluids interactively as demonstrated by the *Virtual Crepe Factory* [3]. This device-based approach is also proposed by Wusheng [31], in the form of a glove enhanced with vibrators mounted in each finger. The glove provides both external forces of contact and internal forces of grasping.

Like those employed for virtual reality, these approaches require the user to hold an input device, rather than giving a feeling of directly touching a virtual object. Some efforts have been focused on using alternatives to real surfaces to produce physical sensations - research efforts have sought to enable hands-free haptic feedback by means of air pressure, acoustic techniques, or electrical signals [15, 20, 27]. While hands-free, because these use non-solids for output, they are limited to the degree they can simulate solid objects.

### Additional Encountered-type Haptics

Our approach relies on hands-free tactile feedback using an active device to render shape attributes. In addition to the projects discussed above in the area of robotic graphics, our current project builds on additional work in encounter-type haptics. Pangaro [23] and Patten [24] propose the magnetic actuation of objects, while Fukushima [6] controls object friction by applying vibration to a surface. Kim et al. apply the same friction principle to tactilely render features such as bumps or textures [15]. While these approaches provide hands-free haptic, they are constrained to the surface of a display, and thus do not simulate 3D shape as does our Snake Charmer robot.

Active shape displays can introduce a third dimension to a table surface [5, 13, 25]. Rods under the tabletop surface, actuated by servo motors, imbed a shape to an otherwise flat surface. Follmer *et al*. overlay digital information on a shape display to provide hands-free haptic feedback [5]. This setup is akin to ours but provides inadequate resolution, due to the limited number of rods. While shape displays are no longer constrained to the contact surface, the haptic spatial reach they provide is still restricted by the physical size of the rods, making larger interaction volumes infeasible.

Sinclair mounted a touchscreen on a linear actuator [26]. When touch is detected, the linear actuation is able to push back the finger to reflect the physical properties of virtual objects on the screen. In ZeroN, Lee et al. magnetically levitate a sphere. The sphere is free to move in a 3D volume and can serve as a proxy for virtual objects. Users are able to manipulate virtual objects by reaching into the volume and physically moving the sphere [18]. Both solutions provide a true 3D volume where tactile feedback can be sensed. However, each has significant limitations in the range of physical objects they are able to simulate.

In contrast, Snake Charmer is able to dynamically synthesize the size, shape, temperature, texture, and weight of arbitrary virtual objects. We dub this synthesis the "physical enablement of virtual objects".

## PHYSICALLY ENABLING VIRTUAL OBJECTS

We have paired a tracked head-mounted display with a depth camera for free-hand tracking, and a robotic arm to provide the physical feedback. The robotic arm allows us add a physical presence to virtual objects that are within user's reach. Users can reach inside the interactive volume to feel, push, heft, grab, and even be touched by virtual objects.

Our approach contributes to the field of robotic graphics, adding additional physical simulation to the virtual objects. We explore three novel physical embodiments: surface synthesis, interaction, and physical behavior.

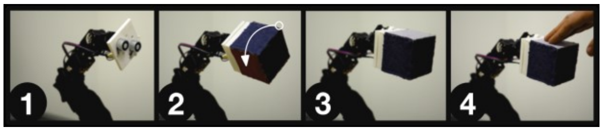

**Figure 3: Surfaces are synthesized by (1) the robotic arm; (2) with an attached endpoint. By rotating the endpoint to present the desired texture (3), the surface texture presented to the user (4) aligns to the virtual object.**

### Surface Synthesis

We feel that object position alone, without additional physical cues, is not sufficient to suspend disbelief when touching a virtual object [7, 9]. Cues such as orientation, texture, and temperature are essential in order to understand any object. Leveraging a robotic arm, we can provide many of these cues and, indeed, further immerse the user in the belief that the object is physically present.

Our robotic arm is equipped with "endpoints" which provide a palette of textures which can be programmatically selected. When a finger is sufficiently close to a virtual surface, the robotic arm moves to spatially align with the location and orientation of that portion of the object. The robotic arm grip rotates to provide the right texture (see Figure 3). When the finger approaches another surface with a different texture, the arm moves to the object's position and selects another texture, thus dynamically rendering distinct tactile feedback.

Physical characteristics are supported by attaching different endpoints. This ability gives our system the opportunity to simulate a significant range of physical renderings. By taking advantage of the agility of our robotic arm, one haptic endpoint can be built to support multiple variations of the

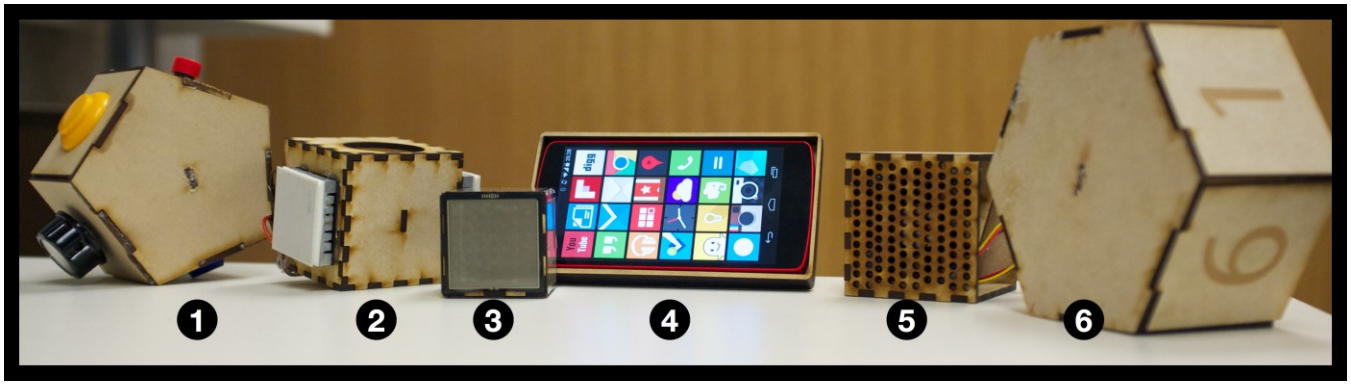

**Figure 2: Examples of different endpoints that can be attached to simulate surface characteristics. (1) Physical controls, (2) Peltier Pad for heating/cooling, (3) whole object interaction via pressure sensing, (4) touchscreen, (5) air flow, (6) six sided texture base.**

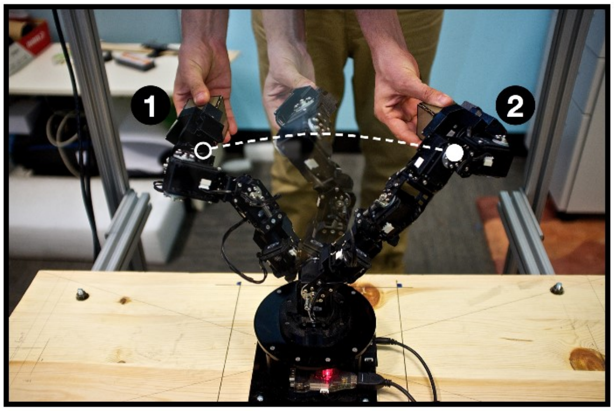

**Figure 4: When a user grabs an object (1) the robotic arm releases control, allowing the user to physically move the virtual object. (2) On release, the robotic arm assumes control, maintaining the new position.**

same physical characteristic (e.g. multiple textures of a sneaker) or distinct characteristics (e.g. texture, pressure, and temperature). To increase the number of available tactile surfaces, the robotic arm can exchange the endpoint in realtime. We contribute a wide variety of endpoints to enable a varierty of surface simulations (see Figure 2 and the video figure).

Some characteristics can be sensed before a surface is touched. For example, when presented with a virtual animal, breath can be felt coming from its mouth; when the user is in a position to feel the breeze, the robotic arm moves into position and a fan endpoint (Figure 2.5) is activated.

### Physical Interaction

In the real world, an object is touched with an intention, for example to move it to another position or activate a function. In the same vein, a virtual button should be able to be physically clicked, and object should be moved by grasping or pushing. Physical interactions are supported on both local interactive elements (physical widgets - Figure 2.1) and physical actions such as grasping, pushing, or holding an object to feel its weight. We demonstrate two physical interactions: users can push the object with one finger by applying pressure to the corresponding side, as one would 'push' an object lying on a table, with objects of different mass requiring more pressure. Users can also grasp the object to move it: we simply turn off the servos on the robotic arm and let the user reposition the object freely. When the user releases the object, we reactivate the robotic arm servos and maintain the object's position, as depicted in Figure 4.

### Physical Behavior

In some scenarios, it is desirable to proactively reach-out to the user. Instead of waiting for a touch to a virtual object, the arm preemptively intersects the hand positions to provide physical feedback. This useful in training for proper use of machinery, or to simulate an animal reaching out and playing – see our video figure for the simulation of a snake reaching out to 'nudge' the user. We have found this extremely effective in improving immersion – the volume of yelps during demonstrations makes this abundantly clear.

## SYSTEM IMPLEMENTATION

Snake Charmer is based on a *DK2 Oculus Rift* head-mounted display for the visualization and a *Robai Cyton Gamma 300* robotic arm to physically enable the virtual objects. The virtual environment is blended with a live 3D reconstruction captured by an Xbox One Kinect depth camera (Figure 5). The input space of the depth camera, the Oculus positional tracker, and the robotic arm's servo feedback (mapped to a model of the arm) are calibrated into a single coordinate space.

The robotic arm is positioned at waist height. This allows the users to reach into the interactive volume with enough time for the system to detect hand position and predict where the user is going to touch, while still being comfortable to reach. The robot is detected within our setup using the RGBD information of the depth sensor. The arm's reach and position effectively define the interactive volume.

The depth camera is used to track the user's hand position. First, we remove the robotic arm mesh information from the 3D reconstruction using bounding boxes updated with the real-time state of the robotic arm. The remaining mesh information is rendered orthogonally from a top view to track hand position. This position is computed by analyzing the largest connected components filtered by an edge detector based on color discrepancies. Then, we calculate the closest virtual object to hand position using a k-d tree data structure. Finally, we use the *Robai API* to control the robotic arm and match both the position and local orientation of the virtual object to be ready to be touched. All tracking and updating is done continuously, and runs at approximately 30 Hz. A rendering of the user's hands is included in the imagery shown to the user, as can be seen in Figure 9.

Endpoints are typically cubes with 64mm-square faces. We choose this size based on the precision of the robotic arm and the detail of the hand visualization presented to the user. The endpoints are magnetically attached to the robotic arm with two Groove electro magnets controlled by an Arduino board. The Arduino board is also used to read status and control the different endpoints used in our explorations This allows the robotic arm to quickly change endpoints by turning off the magnets to discard an endpoint and turn them on to pick up a different endpoint. We pre-recorded a set of 3D positions for endpoints around the robot. During the setup, the handles were attached to the robot in charge to place them on these

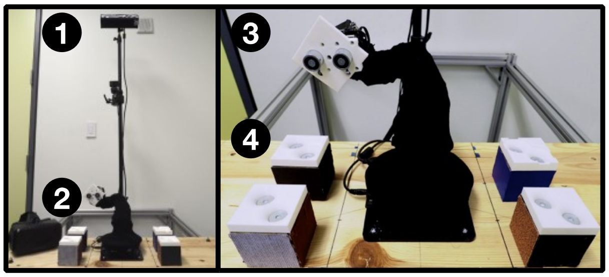

**Figure 5: The system is composed by a depth camera that tracks the hand (1), an Oculus Rift (2), and a Robai 300 robotic arm (3). The robotic arm is able to magnetically attach endpoints (4) accordingly to the application.**

positions. The repeatability of motion, achievable by the robot, was sufficient to grab them programmatically.

### Endpoint Types

We implemented three types of endpoints: passive, input, and active. These are described below and in the figures.

Passive Endpoints simulate textures or local features – see Figure 6. These are implemented based on geometrical shapes, where each side is a different characteristic. Larger shapes allow for more characteristic to be represented, but are constrained by weight. The last passive endpoint simulates a snake head and was 3D printed as a round surface designed for physical contact with the user's hand.

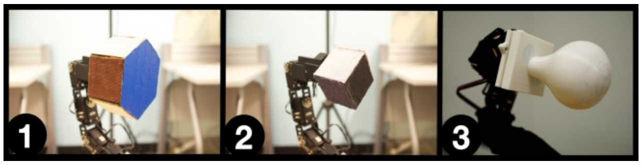

**Figure 6: Endpoints synthesize physical features: (1) various materials of a shoe (2) several textures for a clothing scenario (3) a snake head used to 'attack' the user.**

Input Endpoints provide fine-grained user input, such as the *grasp-force* sensor, *touch sensor*, and *tangible UI* controls, shown in Figure 7. Input is captured via an Arduino.

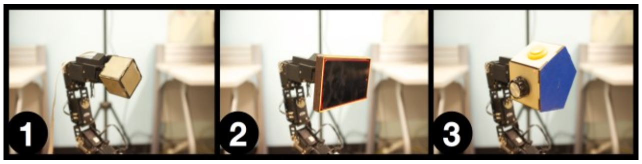

**Figure 7: Examples of endpoints used for input. (1) Force sensors detect grasping, (2) a phone used for its capacitive touch sensor (3) physical UI controls enable fine-grained input and physical feedback.**

Active Endpoints, simulate physical stimuli (Figure 8). *Airflow* output is enabled using a fan, with programmatic control of intensity and direction. The *temperature* endpoint enables exposure of the user to hot or cold surfaces: two opposing faces of the cube are covered by a Peltier Pad, that acts as the temperature element (similar to [17]). At 6 volts the default temperatures in indoor conditions are 20°c and 40°c respectively. An accurate temperature can be achieved by implementing a PID controller that regulates the temperature by using a feedback sensor and a control loop.

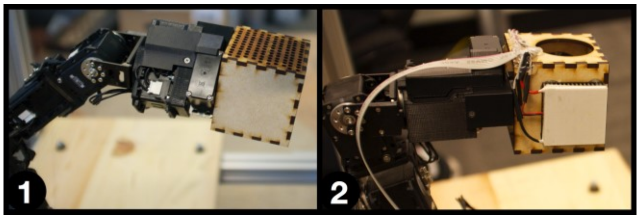

**Figure 8: Example of active endpoints. These two endpoints simulate physical stimuli: air flow (1), and temperature (2).**

## APPLICATION SCENARIOS

We explore each of the three types of endpoints in four applications described below. These serve to demonstrate the value of physical enablement of virtual content.

### Footwear Shop

The first exploration uses Snake Charmer's ability to simulate different textures using a multi-textured endpoint specially designed for virtual models of footwear (i.e. presenting rubber, plastic, leather, nubuck, and suede). In this application (Figure 9), the user is presented with a footwear collection, and explore each virtual model one at the time using a set of virtual buttons to customize the active shoe, or to navigate the collection. The robot synthesizes both the virtual model and the user interface, whenever the user's hand reaches for the virtual elements. The robotic arm is continuously placed at the predicted touch position and orients the endpoint to present a textured face that matches the footwear texture of the active virtual model, or physical to virtual buttons. Enabled by Snake Charmer, the footwear shop allows users to feel different textile variations for virtual footwear models that look visually identical.

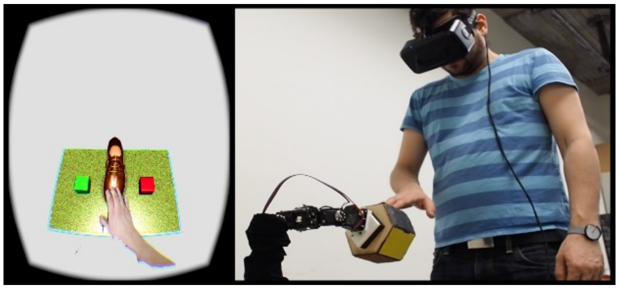

**Figure 9: The footwear virtual shop: the hexagonal endpoint syntheses textures and acts as tangible UI.**

### Building Blocks Playground

The second scenarios enables virtual elements as to act as elements of a tangible UI. We built a virtual building block playground, where users physically grab a pressure sensitive endpoint to move, stack, and drop virtual blocks as if they were real. While physic engines have been previously coupled to tabletops [30], Snake Charmer brings a real tangible representation to the physic simulation that can be picked up and repositioned, pushed, or dropped.

When the user picks up a block, the endpoint becomes a tangible representation of the block and can be move freely in space. This is achieved by disabling the robotic arm torque as soon as the user grasps the endpoint, and by re-enabling it on release. The physical simulation is continuously updated to the expected behavior on the virtual scene. Furthermore the robotic arm torque can be controlled to restrict movement along a given virtual motion, to simulate rigid contacts, or to constrain the motion to mimic the physical interference of other objects. This enables the user to feel physical behaviors of virtual objects while supporting tasks from stacking to throwing a building block to hit other blocks. A working version of this scenario is presented in the video figure.

### Control Panel Simulator

Here we revisit the initial use of robotic graphics [8], where the endpoint is used as a proxy to physical widgets such as buttons, switches, and sliders. In our version, the control panel can simulate any combination of physical controls (Figure 10). The robotic arm overlays each virtual widget with a physical equivalent. Triggering the physical element activates the relevant simulated widget. This application allows the customization of control layouts, and could for example enable rapid iteration of physical control panels.

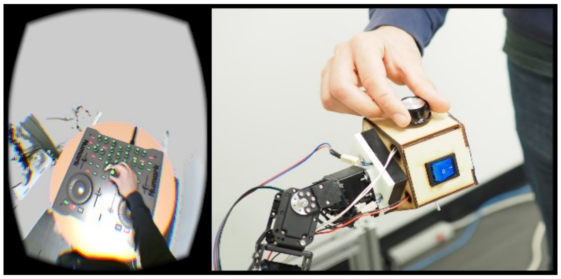

**Figure 10: Testing a virtual DJ mixer using an endpoint composed of physical controllers.**

### Fantasy World

Virtual worlds have long been a focus of VR developers, be it for gaming, storytelling, or simulation. Perhaps because of a lack of physical output, these worlds are often limited to elaborate visual and auditory scenery, out of the user's reach. Snake Charmer can be used to bring physicality to virtual worlds. Our toy environment demonstrates the physical enablement of four objects that can be place in the virtual environment as desired: a volcano, an iceberg, a windmill, and a snake. Each object has a very distinct sensory contribution and is represented using different endpoints: using the Peltier pad enables the volcano and the iceberg feel hot/cold, the fan generates wind that can be sensed in front of the windmill, similarly to [27], and the snake can attack the user using the physical contact endpoint – see Figure 12.

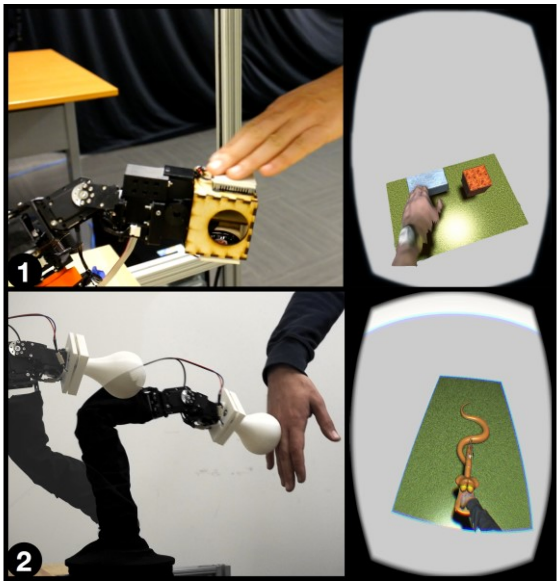

**Figure 12: Simulation of (1) the temperature of a block of ice or lava (2) A nudge from a virtual snake.**

## SYSTEM METRICS

We have found that the combination of tracking and robotic technologies we have employed is sufficient to enable a degree of physical augmentation of virtual content. Performance of our system is a measure of how quickly the arm can reposition itself to simulate a newly targeted virtual object. The components of this time are comprised by three things: sensing latency, arm speed, and the size of the area to which the arm must be moved.

### System Latency

We define the system latency as the time between detecting a user's hand and spatially aligning the robotic arm with the targeted virtual object (see Figure 11). All latency values were empirically measured over several dozen samples. The *Kinect* specification indicates a minimum latency of 20 ms and an average of 60 ms for the depth sensing. Once depth data is reported, our software requires approximately 30ms to determine hand position. At this point, we have all the information to update the system and (1) render the new state and (2) update the arm position. The remaining component is thus the speed with which the arm can reposition.

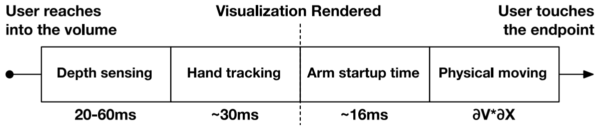

**Figure 11: System block diagram with estimated latency contributions of the individual components.**

#### *Arm Speed*

Three factors affect the time required to reposition the arm: startup time, movement speed, and distance of movement.

*Startup:* Once the signal is sent to the USB subsystem, the arm typically begins to move within 16ms.

*Loaded Speed:* We measured the speed with three loads: no endpoint, cube endpoint (70g), and hexagon endpoint (170g). Our results indicate that these loads have no effect on speed. , therefore enabling texture feedback without decreasing the maximum velocity. Figure 13 shows velocity profiles for the three conditions at four different sideways distances. Each scenario reported in both Figure 13 and Figure 14 was repeated 20 times using the +/- 1mm precision of repeatability achievable with the robotic arm.

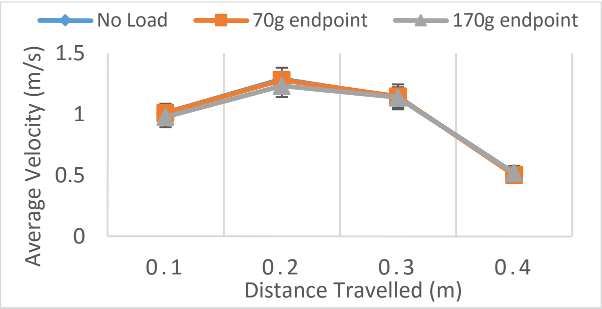

**Figure 13: Sideways velocity for no load (0g), 70g and 170g endpoints. When overextending (0.4m) the robotic arm slows down, thus short distances are preferred.**

*Speed/Distance:* The distance to be traveled is generally minimized by continuously tracking the user's hand, even when not in contact with an object. However, if a user exits and then quickly re-enters the interaction volume, the distance the arm must travel can be great.

Figure 14 shows the average velocity when the arm is moving towards the user's hand, from a resting known position – the typical movement when the user first reaches into the scene. We measured five distances between 0.1 and 0.3 meters. Figure 14 shows the results: we found the maximum velocity of the arm to be 0.9 m/s, for longer distances of 0.25 or 0.3 m, and somewhat slower for smaller distances. The mechanics of the arm make recoil slower than forward movement. This is not problematic because the user is not interacting with the system at this time.

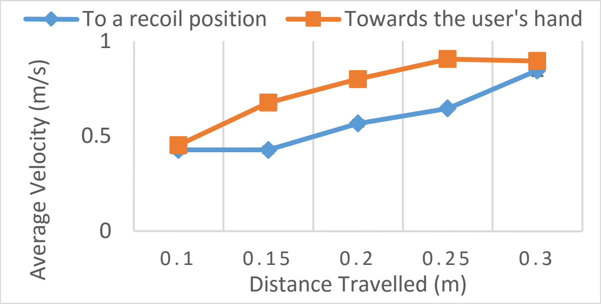

**Figure 14: Velocity profiles for when the robotic arm is extending to spatially align a surface (in orange) and for when it is recoiling in to a rest position.**

### *Interactive Volume*

We next compute the maximum distance to be moved. The reachable volume of the robotic arm is 187 liters, as measured from extremity to extremity, but the interactive volume is additionally constrained by Snake Charmer's need to present a particular endpoint at a particular orientation. While interacting with the virtual scenes proposed in our system, we collected robotic arm positions, and measured a maximal interactive volume of 122 liters (0.54x0.51x0.44m), which is contained within a larger sensing volume of the 3D camera (263 liters). Considering all measurements reported, we estimate the overall system latency.

### *Overall Latency & Mitigation*

The worst case for latency occurs when the robotic arm must spatially align with a surface that is the farthest possible from the resting position. Assuming an average velocity of 0.83 m/s, and a distance of 0.31m, the worst case in the implemented system is 421ms of latency. Of course, because we continuously track the user's hand position, typical movements of the arm are on the order of a 2-3 centimeters. The worst case in typical usage is when the user first reaches-in to the volume, because there is no previously known position. To address this, the user's hand is tracked in a larger volume (263L) than the interactive space (122L). Simple prediction algorithms are used to keep the position of the robotic arm as close as possible to the likely object that is being touched.

## EARLY OBSERVATIONS

Although we have not yet conducted formal user evaluations of our system, we have observed many visitors experiencing Snake Charmer over the last 1.5 years. These are early insights on how people reach into an interaction volume and open directions for future explorations.

When presented with the system, the initial reaction is one of curiosity. Although trusting, users are hesitant to reach in until they have seen some movement of the robotic arm. Cues that indicate that the robotic arm is moving (in particular the servos sound) see a typical reaction of hesitantly reaching towards the objects, almost expecting an object not to be there. Once the users make contact with the endpoint and realize the physicality of the virtual object, the novelty of the prototype sets in, and they start to explore the scene. The experience is highly delightful, with users routinely exclaiming things such as "it's really there!" and "this is amazing!". The clear hesitation demonstrated when interacting with the virtual snake, shown in Figure 12, users report feeling like there really is a living creature in the room – this despite the rudimentary graphical representation evident in the figure. From these informal examinations, it is clear that robotic graphics, which has only been initially explored, and expanded upon in our work, provide a highly compelling avenue for advancing virtual reality beyond the visual and auditory senses.

Interestingly, despite our best efforts, it is possible in the current prototype, when attempting to synthesize a surface, the robotic arm hits the user's finger, giving a bit of a 'surprise'. We were intrigued by the normally positive user response. Many users assumed that the robotic arm was right and the object was closer than expected. One user in particular believed that the robot was taking control, moving his hand to the right position. The power of robotic graphics as an output modality is highly apparent.

### Discussion

We envision a world where the endpoint is as narrow as a fingertip and precise enough to mimic even the most minute curvature. However, commodity robotic arms have limits on velocity and spatial resolution. At the moment they limit how fast we can follow a finger and the fidelity of the synthesized surface. Despite this, we have found the present configuration to be highly engaging.

The robotic arm is able to spatially align to a surface, but sometimes the arm is required to be horizontally extended to render the surface. This means that, if the user pushes hard against the endpoint, the robotic arm will bend under the applied force. It would be advisable to create a feedback loop where, when the user causes the arm to bend, the virtual scenery is updated to reflect it.

Taking inspiration in shape displays technology, our approach can be enriched by increasing the number of robotic arms. Merely increasing to two robotic arms would allow us to model corners and ladders, both currently impossible. Multiple robotic arms would allow us to model complex surfaces, and maintain, at all times, a surface that users could approach, but also explore without lifting their finger.

## CONCLUSION

We have presented Snake Charmer, an extension of robotic graphics, which utilizes virtual reality to physically enable virtual objects. By spatially aligning physical surfaces to virtual objects, we offer the suspension of disbelief that the object is physically there. We demonstrate our approach with a multitude of endpoints that allows us to dynamically render textures, temperature, and physical weight. Our early observations show that the approach has promise and that users can easily abstract and pretend that virtual objects are real, i.e. that can be seen and touched.

## ACKNOWLEDGEMENTS

We would like to acknowledge the fundamental help of Kate Down and Katie Barker during the writing of this paper.

## REFERENCES

1. Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. 2010. TeslaTouch: Electrovibration for Touch Surfaces. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 283–292. DOI: http://dx.doi.org/10.1145/1866029.1866074
2. Mohamed Benali-Khoudja, Moustapha Hafez, Jean-Marc Alexandre, and Kheddar Abderrahmane. 2004. Tactile interfaces: a state-of-the-art survey. In *Proceedings of the International Symposium on Robotics*, Vol. 31. 721–726.
3. Gabriel Cirio, Maud Marchal, Sebastien Hillaire, and Anatole Lécuyer. 2011. The Virtual Crepe Factory: 6DoF Haptic Interaction with Fluids. In *ACM SIGGRAPH 2011 Emerging Technologies (SIGGRAPH '11)*. ACM, New York, NY, USA, Article 17, 1 pages. DOI: http://dx.doi.org/10.1145/2048259.2048276
4. Lionel Dominjon, Jérôme Perret, and Anatole Lécuyer. 2007. Novel devices and interaction techniques for human-scale haptics. *The Visual Computer 23, 4 (2007)*, 257–266.
5. Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: Dynamic Physical Affordances and Constraints Through Shape and Object Actuation. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 417–426. DOI: http://dx.doi.org/10.1145/2501988.2502032
6. Shogo Fukushima, Yuki Hashimoto, and Hiroyuki Kajimoto. 2008. Tabletop Interface Using a Table's Circular Vibration and Controllable Friction. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems (CHI EA '08)*. ACM, New York, NY, USA, 3801–3806. DOI: http://dx.doi.org/10.1145/1358628.1358933
7. James J. Gibson. 1962. Observations on active touch. *Psychological Review* 69 (1962), 477–491. Issue 6. DOI: http://dx.doi.org/10.1037/h0046962
8. Peter E Gruenbaum and W McNeely. 1997. Implementation of dynamic robotic graphics for a virtual control panel. *Presence: Teleoperators and Virtual Environments* 6, 1 (1997), 118–126.
9. Kelly S. Hale and Kay M. Stanney. 2004. Deriving Haptic Design Guidelines from Human Physiological, Psychophysical, and Neurological Foundations. *IEEE Comput. Graph. Appl.* 24, 2 (March 2004), 33–39. DOI: http://dx.doi.org/10.1109/MCG.2004.1274059
10. M. Heilig. 1962. Sensorama Simulator. (1962). http://www.mortonheilig.com/SensoramaPatent.pdf
11. Steven J. Henderson and Steven Feiner. 2008. Opportunistic Controls: Leveraging Natural Affordances As Tangible User Interfaces for Augmented Reality. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology (VRST '08)*. ACM, New York, NY, USA, 211–218. DOI: http://dx.doi.org/10.1145/1450579.1450625
12. Milan Ikits and J. Dean Brederson. 2005. The visual haptic workbench. *The Visualization Handbook* (2005), 431–449.
13. Hiroo Iwata, Hiroaki Yano, Fumitaka Nakaizumi, and Ryo Kawamura. 2001. Project FEELEX: Adding Haptic Surface to Graphics. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. ACM, New York, NY, USA, 469–476. DOI: http://dx.doi.org/10.1145/383259.383314
14. H. Kawasaki and T. Mouri. 2007. Design and Control of Five-Fingered Haptic Interface Opposite to Human Hand. *Robotics, IEEE Transactions on* 23, 5 (Oct 2007), 909–918. DOI: http://dx.doi.org/10.1109/TRO.2007.906258
15. Seung-Chan Kim, Ali Israr, and Ivan Poupyrev. 2013. Tactile Rendering of 3D Features on Touch Surfaces. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 531–538. DOI: http://dx.doi.org/10.1145/2501988.2502020
16. L. Kohli, M.C. Whitton, and F.P. Brooks. 2012. Redirected touching: The effect of warping space on task performance. In *Proceedings of IEEE Symposium on 3D User Interfaces (3DUI), 2012*. 105–112. DOI: http://dx.doi.org/10.1109/3DUI.2012.6184193
17. A. Kron and G. Schmidt. 2003. Multi-fingered tactile feedback from virtual and remote environments. In *Proceedings of 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*

*(HAPTICS 2003)*, 16–23. DOI: http://dx.doi.org/10.1109/HAPTIC.2003.1191219

18. Jinha Lee, Rehmi Post, and Hiroshi Ishii. 2011. ZeroN: Mid-air Tangible Interaction Enabled by Computer Controlled Magnetic Levitation. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 327–336. DOI: http://dx.doi.org/10.1145/2047196.2047239
19. Kevin A. Li, Patrick Baudisch, William G. Griswold, and James D. Hollan. 2008. Tapping and Rubbing: Exploring New Dimensions of Tactile Feedback with Voice Coil Motors. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, New York, NY, USA, 181–190. DOI: http://dx.doi.org/10.1145/1449715.1449744
20. Nicolai Marquardt, Miguel A. Nacenta, James E. Young, Sheelagh Carpendale, Saul Greenberg, and Ehud Sharlin. 2009. The Haptic Tabletop Puck: The Video. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*. ACM, New York, NY, USA, Article 2, 1 pages. DOI: http://dx.doi.org/10.1145/1731903.1731943
21. Thomas H. Massie and J. K. Salisbury. 1994. The PHANToM haptic interface: A device for probing virtual objects. In *Proceedings of the ASME Dynamic Systems and Control Division*. 295–301.
22. W.A. McNeely. 1993. Robotic graphics: a new approach to force feedback for virtual reality. In *Proceedings of the Annual IEEE International Symposium on Virtual Reality 1993*. IEEE, 336–341. DOI: http://dx.doi.org/10.1109/VRAIS.1993.380761
23. Gian Pangaro, Dan Maynes-Aminzade, and Hiroshi Ishii. 2002. The Actuated Workbench: Computer-controlled Actuation in Tabletop Tangible Interfaces. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST '02)*. ACM, New York, NY, USA, 181–190. DOI: http://dx.doi.org/10.1145/571985.572011
24. James Patten and Hiroshi Ishii. 2007. Mechanical Constraints As Computational Constraints in Tabletop Tangible Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 809–818. DOI: http://dx.doi.org/10.1145/1240624.1240746
25. Ivan Poupyrev, Tatsushi Nashida, Shigeaki Maruyama, Jun Rekimoto, and Yasufumi Yamaji. 2004. Lumen: Interactive Visual and Shape Display for Calm Computing. In *ACM SIGGRAPH 2004 Emerging Technologies (SIGGRAPH '04)*. ACM, New York, NY, USA, 17–. DOI: http://dx.doi.org/10.1145/1186155.1186173
26. Mike Sinclair, Michel Pahud, and Hrvoje Benko. 2013. TouchMover: Actuated 3D Touchscreen with Haptic Feedback. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces (ITS '13)*. ACM, New York, NY, USA, 287–296. DOI: http://dx.doi.org/10.1145/2512349.2512805
27. Rajinder Sodhi, Ivan Poupyrev, Matthew Glisson, and Ali Israr. 2013. AIREAL: Interactive Tactile Experiences in Free Air. *ACM Trans. Graph.* 32, 4, Article 134 (July 2013), 10 pages. DOI: http://dx.doi.org/10.1145/2461912.2462007
28. Robert J Stone. 2000. Haptic Feedback: A Potted History, From Telepresence to Virtual Reality. In *Proceedings of the First Int. Workshop on Haptic Human-Computer Interaction*. Springer.
29. Susuma Tachi, Taro Maeda, Ryokichi Hirata, and Hiroshi Hoshino. 1994. A construction method of virtual haptic space. In *Proceedings of the 4th International Conference on Artificial Reality and Tele-Existence (ICAT'94)*. 131–138.
30. Andrew D. Wilson, Shahram Izadi, Otmar Hilliges, Armando Garcia-Mendoza, and David Kirk. 2008. Bringing Physics to the Surface. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, New York, NY, USA, 67–76. DOI: http://dx.doi.org/10.1145/1449715.1449728
31. Chou Wusheng, Wang Tianmiao, and Hu Lei. 2003. Design of data glove and arm type haptic interface. In *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS 2003*. 422–427. DOI: http://dx.doi.org/10.1109/HAPTIC.2003.1191332
32. Y Yokokohji, R Hollis, and T Kanade. 1999. WYSIWYF Display: A Visual/Haptic Interface to Virtual Environment. *Presence* 8, 4 (Aug 1999), 412–434. DOI: http://dx.doi.org/10.1162/105474699566314
33. Mounia Ziat, Taylor Rolison, Andrew Shirtz, Daniel Wilbern, and Carrie Anne Balcer. 2014. Enhancing Virtual Immersion Through Tactile Feedback. In *Proceedings of the Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST'14 Adjunct)*. ACM, New York, NY, USA, 65–66. DOI: http://dx.doi.org/10.1145/2658779.2659116