HapTwist: Creating Interactive Haptic Proxies in Virtual Reality Using Low-cost Twistable Artefacts

Kening Zhu
School of Creative Media
City University of Hong Kong
Hong Kong, China
keninzhu@cityu.edu.hk

Taizhou Chen
School of Creative Media
City University of Hong Kong
Hong Kong, China
taizhou.chen@my.cityu.edu.hk

Feng Han
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong, China
fhanab@connect.ust.hk

Yi-Shiun Wu
Department of Micro Engineering
EPFL
Lausanne, Switzerland
alan830908@gmail.com

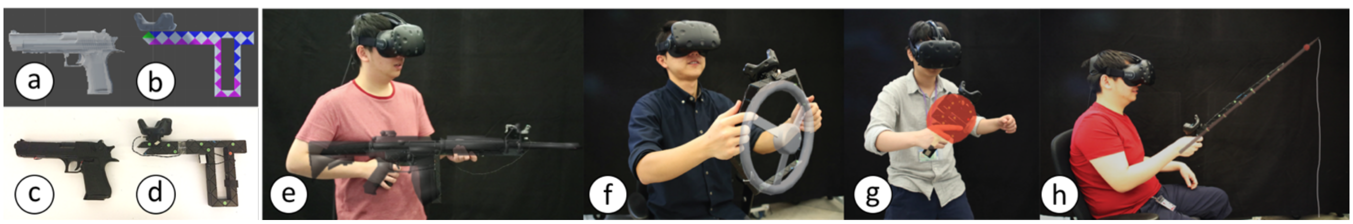

Figure 1: (a) Virtual pistol model, (b) generated Rubik's Twist shape, (c) real object, and (d) the physical haptic proxy, (e - h) Examples of HapTwist-generated interactive VR haptic proxies, left to right: machine gun, steering wheel, ping-pong paddle, and fishing rod.

Abstract

In this paper, we present a series of studies on using Rubik's Twist, a type of low-cost twistable artefact, to create haptic proxies for various hand-graspable VR objects. Our pilot studies validated the feasibility and effectiveness of Rubik's-Twist-based haptic proxies. The pilot results also revealed user challenges in the physical shape creation, motivating the development of the HapTwist toolkit. The toolkit consists of the shape-generation algorithm, the software interface for shape-construction guidance and interaction authoring, and the hardware modules for constructing interactive haptic proxies. The user studies showed that HapTwist was easy to learn and use, and it significantly improved user performance in creating interactive haptic proxies with Rubik's Twist. Furthermore, HapTwist-generated haptic proxies achieved similar VR experience as the real objects.

CCS Concepts

• **Human-centered computing** → **Virtual reality**; **Haptic devices**; **User interface toolkits**.

Keywords

Virtual Reality; Hand Grasp; Haptics; Rubik's Twist; Toolkit

ACM Reference Format:
Kening Zhu, Taizhou Chen, Feng Han, and Yi-Shiun Wu. 2019. HapTwist: Creating Interactive Haptic Proxies in Virtual Reality Using Low-cost Twistable Artefacts. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3290605.3300923

1 Introduction

In recent years, virtual reality (VR) has been gaining an increasing amount of attention in the consumer market. However, it remains challenging to use current vibration-based

ACM ISBN 978-1-4503-5970-2/19/05...$15.00
https://doi.org/10.1145/3290605.3300923

VR controllers to simulate realistic hand-grasping interaction, which often involves different combinations of thumbs, fingers, and palms [13]. This limitation prevents the effective representation of stereognostic information (i.e. the shape, size, and weight) of virtual objects. Researchers have spent a significant amount of time and effort in designing various types of haptic instruments for VR, such as handheld deformable controllers [50], exoskeletons [5, 8–10, 36], wearable electric stimulation [49], and grounded/ungrounded shape-changing surfaces [1, 4]. These haptic devices are usually considered as active haptics (AH), with high responsive speed and controllability. However, AH were often bulky, heavy, time- and money-consuming to set up, and are limited in expressiveness for providing the details of shapes. In addition, most existing AH handheld controllers (except [22]) only support single-hand/finger stimulation with the restriction of hand postures. On the other hand, researchers have investigated using physical objects as haptic proxies, which is also known as passive haptics (PH). Compared to AH, low-fidelity PH are relatively cheaper to produce, either by manual prototyping or digital fabrication (e.g., 3D printing). Moreover, a physical proxy can provide natural and real kinesthetic feedback for free-hand grasping. However, PH were limited by scalability and reusability [42]. It is usually slow to assemble or fabricate the 3D models for each VR experience. It can also be costly and tedious for users to prepare a large set of physical objects for each virtual experience. More recently, shape-changing interfaces were adopted as a type of passive haptic proxy [33]. However, the need of users' in-VR shape manipulation may require careful design of the VR scenario and storyline.

In this paper, we present HapTwist, a toolkit that facilitates the creation of interactive haptic proxies for hand-graspable VR objects. The toolkit leverages Rubik's Twist (RT, also known as Rubik's Snake [14]), a type of low-cost twistable artefact. With HapTwist, we aim to address the complexity and the cost of AH, and the low reusability and the limited interactivity of PH. Prior to the development of HapTwist, we investigated the feasibility of using RTs for passive VR haptics. The results showed that RTs could be shaped to form haptic proxies covering a wide range of hand-grasp postures. The RT-based haptic proxies offered significantly better VR experience than existing commercial VR controllers. On the other hand, users found it challenging to construct the RT-based proxies from scratch by themselves, and it was lack of active interactivity in the constructed proxies, motivating us to develop HapTwist.

The HapTwist software can generate RT-based shapes based on the 3D models, and provide guidance for physical construction. Different from existing passive haptic proxies, HapTwist software allows users to annotate the interactive input/output (I/O) modules in the generated shape, and export the scripting scaffolds for the annotated interactive modules. We developed a variety of I/O hardware modules, including push buttons, triggers, vibrators, fans, and thermal-electric heater/coolers. These modules support easy assembly and disassembly on the physical RT-based shapes. Based on our empirical tests, the smallest object that can be made by the current toolkit using one single RT (12-piece) is a lighter (H2.5xW2xD0.8"), and the largest object that can be made with single RT (108-piece) is the 2D replica of a fire extinguisher (H20xW10.6xD5.2"). Larger objects can be made by combining more RTs. While our current system could mainly generate 2D shapes, our studies showed it was sufficient to use 2D construction to represent a wide range of hand-graspable objects in VR. Our user studies also showed that users perceived the HapTwist toolkit as easy to learn and use. Further, it significantly improved user performance in creating haptic proxies, and the HapTwist-generated interactive proxies can achieve comparable haptic experience in VR to the real objects.

This paper makes the following contributions:

- Empirical study with insights into the feasibility of using RT as haptic proxy for VR.
- The HapTwist toolkit that provides intuitive software interface and hardware components to facilitate the creation of interactive haptic proxies for VR.
- User study of the HapTwist toolkit validating its overall effectiveness and support on haptic proxy creation.

2 Related Works & Background

HapTwist is highly inspired by existing research on VR haptics. In this section, we will discuss existing active and passive haptic devices for hand-grasp interaction in VR. We will also review the existing research in applying deformable user interfaces for VR haptics.

Active & Passive Haptics for VR

Existing VR haptic interfaces can be categorized as two types: active (AH) or passive (PH). AH usually involve fine-controlled mechanical and electrical systems. They are usually heavy and bulky to set up, and place constraints on user movements. Recent research efforts have also attempted to design new types of VR handheld controllers with built-in haptic actuators. Provancher [37] introduced Reactive Grip, a VR controller that simulates in-palm friction and torsional forces using sliding contactors in the device's handle. Zenner and Kruger [50] developed Shifty, an ungrounded weight-shifting VR controller that adjusts the internal weight distribution with a linear stepper actuator. Benko et al. [4] designed NormalTouch and TextureTouch, two types of trackable handheld controllers with tiltable planes and actuated pin matrixes. These controllers can haptically render object surfaces and textures on a user's fingertips. More recently, Lo et al. [31] developed RollingStone, using a single slip tactile actuator on VR controllers to produce sensations of finger sliding and textures. Teng et al. [45] developed PuPop, a light-weight pneumatic shape-proxy interface worn on the palm to simulate the predefined primitive shapes for grasping in VR. Whitmire et al. [47] developed Haptic Resolver, a handheld controller with a rotary wheel that raises and lowers below the user's finger. The rotary wheel can be attached with different surface textures and input components (e.g., buttons) to simulate interaction with various VR surfaces. Holz et al. [22] designed Haptic Links, a set of electro-mechanical connections for rendering the stiffness between two HTC Vive handheld controllers to simulate bimanual object interaction in VR. Heo et al. [18] designed a handheld force feedback device called Thor's Hammer, which generates 3-DOF VR force feedback using propeller propulsion. With a similar form factor to commodity VR handheld controllers, these new controllers often restricted a user's hand posture during use. Thus, they placed constraint on free-hand grasping interaction with the virtual object, only providing localized feedback on a single finger or single hand. While Haptic Links supports bimanual interaction, it is still heavy and bulky, resulting in lower comfort than commercial controllers.

In contrast to AH, PH aim at using existing physical objects to provide haptic feedback [20]. Low-fidelity proxy objects are typically cheap to produce, by either manual prototyping or using digital fabrication (e.g., 3D printing and laser cutting). Passive haptics have been used for medical volumetric data visualization [27], 3D modeling [41], interacting with user interface elements [30], and, at a larger scale, representing entire rooms or spaces [24]. Through visual warping, a single physical proxy can be used to represent multiple virtual objects of the same type [3]. Recently, researchers have designed a machine-learning-based approach for the design and fabrication of haptic proxies to stimulate the illusion of shape and weight in VR [16]. Some researchers have leveraged other users to reposition and assemble large-scale haptic proxies [7]. Hettiarachchi and Wigdor developed Annexing Reality [19], a design platform to select the best haptic matches from the user's current environment for virtual objects. While physical 3D props demonstrate benefits in VR by reducing the cost and the complexity of the haptic devices, research also points the drawback of requiring specific physical objects for visually different virtual objects [42]. It could be challenging to find a physical prop from any given space that is identical/similar to a virtual object. Further, due to low reusability, it is costly and slow to fabricate (e.g., 3D print) 3D models for each virtual experience. It is also unrealistic to require a user to purchase a large set of props for each virtual experience.

The HapTwist toolkit leverages the low-cost advantage of PH and enhances the expressiveness and the reusability of haptic proxies by using a daily twistable artefact, Rubik's Twist, which can be shaped and assembled to represent different objects. A shaped Rubik's Twist can be disassembled and reused for new virtual scenarios. In addition, the generated haptic proxies support active I/O without any constraint on a user's hand postures.

Deformable Interface for VR Haptics

Many researches [38] have shown the benefits of shape-changing/deformable interfaces through the addition of tactile feedback to digital contents. A linear-actuator matrix form factor has been widely adopted in tabletop devices [15], wrist-worn devices [23], and handheld devices [28], for haptically representing digital information. However, these actuated deformable interfaces could suffer from similar weaknesses to those of active haptics (e.g., being bulky and not hand-graspable). Incorporating the advantages of robotic network, Zhao et al. [51] developed a set of block-type swarm robots to assemble physical handheld proxies for VR. While the self-assembly robot network could assemble and be reused for various low-fidelity haptic proxies, it might suffer from low accuracy and speed.

Aguerreche [2] designed general toolkits with stick-shaped modular props allowing assembly and disassembly of haptic representations for VR objects. Mueller et al. [35] developed faBrickation to generated construction instructions for replicating virtual models using Lego. However, it could be challenging and time-consuming to piece-by-piece build the shapes for complicated 3D models [43]. Recently, McClelland et al. [33] explored foldable paper-like devices as passive haptic proxies in VR. Their device with embedded folding sensors detected the shape of the device and summoned the 3D object with similar shape. Cheng et al. [6] developed iTurk, leveraging a large foldable object to represent cabinets, seats, and suitcases, etc. While these systems allowed and guided users to manually change the shapes of the proxies in VR, this feature potentially required careful design of in-VR interaction flows and storylines to reasonably guide user actions.

To facilitate the construction of shape-changing-interface-based haptic proxies, our HapTwist toolkit generates shaping and assembly instructions based on 3D geometric analysis. User studies showed that the generated instructions and scaffolds support the haptic representation of a comprehensive set of hand-grasp VR objects, with significantly improved efficiency of creating interactive VR haptic proxies.

3 Daily Deformable Objects/Materials: Why Rubik's Twist?

To understand the capabilities and limitations of daily deformable artefacts, we conducted a thorough survey, including literatures on common deformable objects and materials, and HCI-related studies on flexible/deformable interfaces

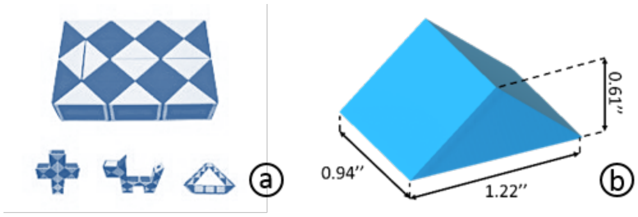

Figure 2: (a) Rubik's Twist (RT) and sample shapes, (b) The dimension of the prism element in RT.

[21, 25, 26, 29, 38] and mechanical assemblies [40, 46]. As the result, we divided deformable artefacts/materials into three main categories: flexible materials, modular assemblies, and twistable joints. The category of flexible materials (e.g., paper, clay, cloth, etc.) represents raw materials that can be malleable and/or ductile in a normal room condition (i.e. 24°C). Modular assemblies include the sets of unified modular parts that can be assembled into large structures, such as LEGO, slot-together pieces, etc. Lastly, the category of twistable joints covers artefacts that consist of multiple unified parts that are interlocked into a whole with twistable/bendable joints. One example is the Rubik's Twist (RT, as shown Fig. 2a), in which multiple prisms (Fig. 2b) are connected, so they can be twisted to form complex shapes but not separated.

According to existing taxonomies on shape-changing artefacts [29], we compared these categories in terms of the ease of shaping (i.e. granularity), the strength of holding the shapes (i.e. stiffness), and the reusability. While flexible materials could be easily shaped (e.g., folded/molded), the process of creating complex shapes with these materials could be challenging, with the requirement of advanced crafting skills, such as origami. Rubber and cloth are also flexible materials, but it is difficult to maintain their shape without external force support. Using modular assemblies, 3D shapes can be "voxelized" into small unified pieces, and the locking structures can hold the assembled shapes. However, the assembly and disassembly processes could be tedious for a large number of modular pieces. Lastly, the manipulation of a twistable-joint structure can usually be formatted, which supports algorithmic design. Their plastic stiffness allows better learning and keeping of shape, and the large number of twistable joints supports expressiveness [29]. In addition, it takes less steps/time than the modular assemblies to form 3D shapes and reset. Therefore, we chose RT as the primitive component for the creation of VR haptic proxies.

4 Pilot Studies: Rubik's Twists as VR Haptic Proxies

Prior to the development of the HapTwist toolkit, we conducted two pilot studies to validate the effectiveness of using RT for VR haptic proxies. Specifically, we focused on the following questions:

- Can users employ RT to create physical shapes that represents virtual hand-graspable objects?
- Can the RT-based physical shapes be effectively used as haptic proxies for hand-graspable objects in VR?

The large variety of available hand-graspable VR objects placed challenges in studying how expressive and capable RT is for creating haptic proxies in VR. Therefore, we first classified and selected the representative virtual objects for our study, based on existing hand-grasp taxonomies.

Selection of Virtual Hand-Graspable Objects

The interaction with real-world objects is highly related to the hand-grasping postures. Existing taxonomies for real-world grasping postures [13, 32] usually categorized a grasp by its need for precision or power, with various combinations of palm, thumb, and other fingers. The grasp types also indicate the sizes and the weights of the objects. Precision grasp is often applied to small objects/handles and usually involves only thumb and other fingers, while lager objects often require power grasp involving the palm. As the size of the graspable object increases, the palm changes from a closed state to an opened state. While existing hand-grasp taxonomies have mostly focused on single-hand grasping postures, there are many situations in real life and VR that involve bimanual grasping [22]. Therefore, we included the number of used hands into the consideration, as shown in Fig. 3, besides the need for precision or power, and the usage of palm, thumb, and other fingers.

To cover the common hand-grasp postures in VR as much as possible, we selected four objects: a mug - single-handed precision grasp; a pistol - single-handed power grasp with closed palm; a mobile phone - single-handed power grasp with opened palm; and a machine gun - bimanual power grasp. Fig. 4 shows the 3D models of these four objects and their real-life dimensions.

Pilot Study 1: Using RT to Physicalize Virtual Objects

In the first pilot study, we investigated the feasibility of using RT to physically replicate the virtual objects. We also aimed to understand any challenges users may encounter and the user-perceived enjoyment during this process.

Participants There were twelve participants (six female and six male) recruited from diverse backgrounds (six computer science students, three VR designers, and three interface designers). The average age was 24.6 years old (SD = 2.16). In the pre-study questionnaire, two participants rated their 3D-modeling skills as expert level, four as intermediate, five as beginners, and one had never tried 3D-modeling before. In addition, all participants had experienced HMD-based VR before. For skill with RT, three participants rated themselves as intermediate level, four as beginners, and five had never played with RT before.

Apparatus and Instruments Fig. 5 shows the setup of the study environment. In each session, the participant was provided

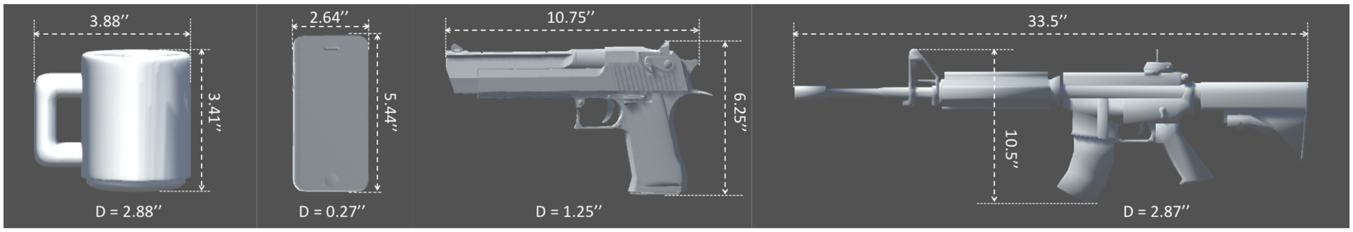

Figure 4: Selected virtual objects for the experiments and their dimensions, left to right: mug, mobile phone, pistol, and machine gun.

Number of Hands	Single			Bimanual
Grasp Type	Precision	Power		Power
		Closed Palm	Opened Palm	Closed Palm
Use of Palm	No	Yes	Yes	Yes
Use of Thumb	Yes	Yes	Yes	Yes
Use of Fingers	Yes	Yes	Yes	Yes
Examples				

Figure 3: Taxonomy for hand-grasp postures/objects in VR.

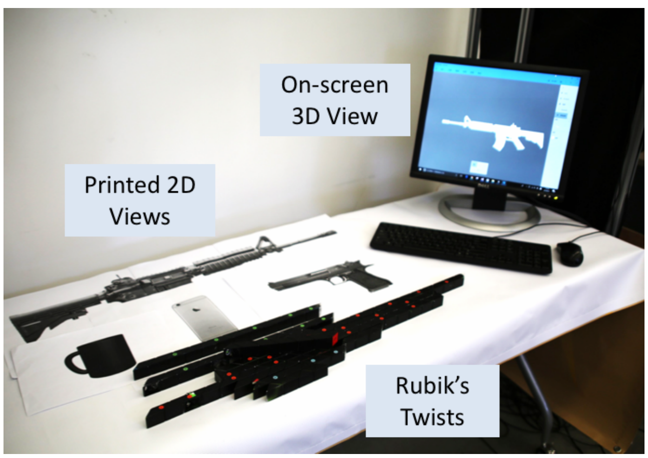

Figure 5: Setup of Pilot Study 1.

with a set of standard RTs with different numbers of triangular prisms (one 108-piece, four 24-piece, and four 12-piece). The RTs were painted in black to avoid any color bias.

The 3D models were shown in a 20-inch monitor, and the participant could freely change the viewing angle with the mouse. Furthermore, we provided the paper-printed 2D side views of the models in their real-world scales. The post questionnaire contained items related the ease and the enjoyment of shape creation, and the perceived workload based on the NASA TLX questionnaire [17], on a 7-point Likert scale.

Task and Procedure Each session included one participant and one moderator. After finishing the pre-questionnaire, the participant was introduced to the RTs and played with

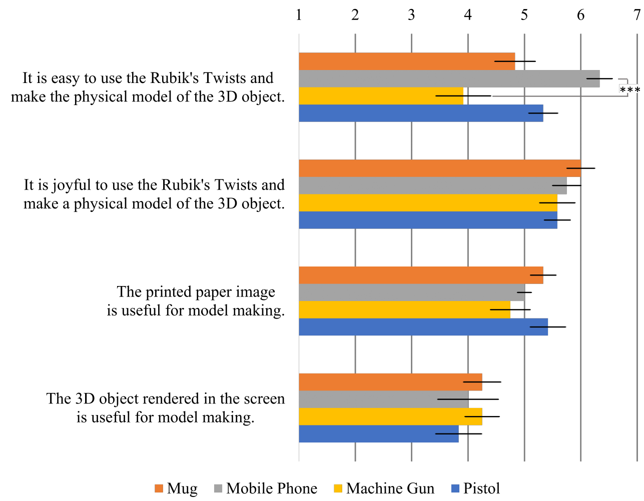

Figure 6: Results of post questionnaire in Pilot Study 1. ***: p < 0.0005

them freely for five minutes as warm-up. The moderator then showed one 3D model on the screen and the corresponding paper print-out. The participant was asked to create three physical replications of the shown model using the provided RTs. After shape creation, participants needed to choose the physical creation that they thought was best match with the 3D model, finish a post questionnaire, and then proceeded to the next 3D model. The four 3D models (mug, mobile phone, pistol, and machine gun) were shown in a Latin-square counterbalanced order. We interviewed the participants about their experience and challenges encountered after making all the models. The time spent on each shape creation was logged, and the whole process was video recorded with the agreement of each participant.

Results Averagely, it took 11.8 minutes (SD = 1.2) for the participant to create a RT-based physical shape. Participants could successfully use single RT to create the physical shapes of the mug, the mobile phone, and the pistol, but not the machine gun due to its large size. We statistically compared the time spent for shape creation and the ratings for post questionnaire for the four 3D models. One-way ANOVA showed that the type of the 3D models significantly affected the time spent on the shape creation ($F(3, 44) = 7.165$, p <

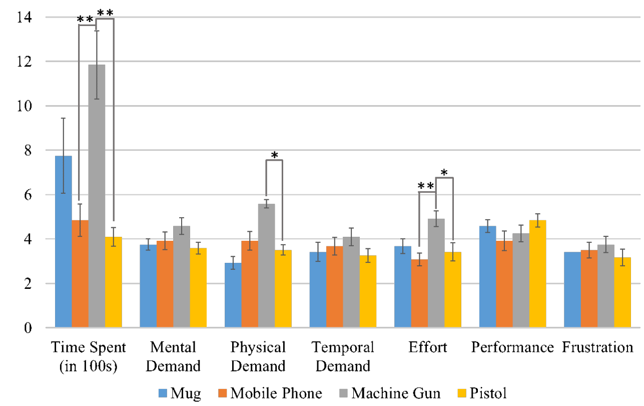

Figure 7: Results of average time spent (in 100s) and participant-perceived workload indicated by NASA TLX in Pilot Study 1. *: p < 0.05, **: p < 0.005

0.005, $\eta^2 = 0.328$). For the questionnaire results, Friedman Test showed the type of the 3D model significantly affected the perceived ease ($\chi^2(3) = 12.78$, $p < 0.005$), physical demand ($\chi^2(3) = 17.83$, $p < 0.005$), and effort ($\chi^2(3) = 12.1$, $p < 0.05$) of the physical shape creation. There was no significant difference on the enjoyment rating among the four models. Fig. 6 illustrates the descriptive results on the post-questionnaire ratings related to the user experience on creating RT-based shapes, and Fig. 7 shows the average time spent and the perceived workload on shape creation. (The results of post-hoc pairwise comparison are indicated by *).

Generally, users found it joyful to create physical shapes of the selected 3D models using RTs (5.7/7, SD = 0.89). Fig. 8 shows the examples of the favored physical shapes created by the participants according to the virtual models. Most of the time, the participants created 2D physical shapes to represent the virtual models, with only 7 out of 144 created shapes being 3D. On the other hand, the post-session interview revealed that using RT could be time-consuming (from 6 to 20 minutes) and challenging for non-experienced users to create the physical shapes. Many participants commented that the most challenging part was to figure out where and how to start, and they spent most of the time on fitting the virtual models by “trial and error".

Pilot Study 2: Using the Rubik’s-Twist-based Physical Shapes as VR Haptic Proxies

In the second pilot study, we investigated the effectiveness of the user-created RT-based physical shapes as the haptic proxies in VR. We compared the user-perceived sense of presence in VR using three types of proxies: real objects, RT-based physical shapes, and existing VR handheld controllers. The same group of participants was invited to the second pilot study two weeks after the first pilot study. As psychological research has shown that the persistence of human memory significantly declines after two weeks [11], we can assume that the participants could not well remember the RT-based shapes they created in the first pilot study.

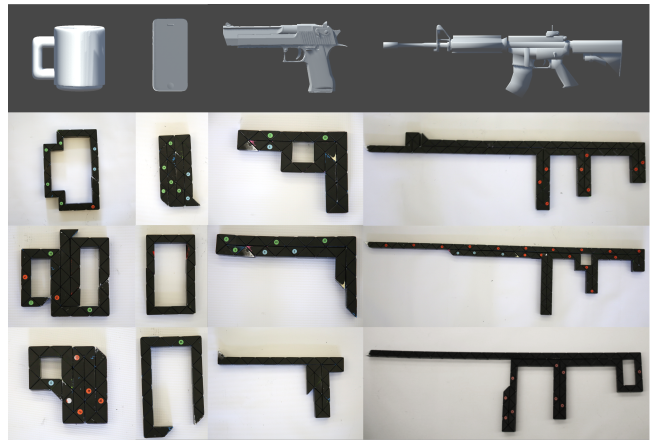

Figure 8: Examples of Rubik’s-Twist-based shapes created by the participants in pilot study 1.

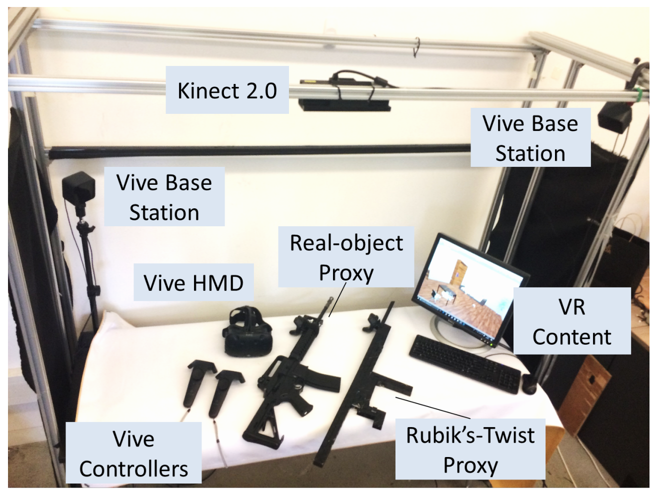

Figure 9: Setup of Pilot Study 2.

Apparatus Fig. 9 shows the setup of the study environment. We developed a virtual-world prototype using Unity3D 2017. The prototype used HTC Vive HMD, a pair of Vive handheld controllers, and two Vive trackers, in combination with an overhead Microsoft Kinect v2.0. The Kinect camera was used to capture the geometry and the appearance of the user’s body. The system displayed realistic 3D views of the user’s own body using the depth and color camera images. The Vive trackers were attached to the real objects and the RT-based shapes, communicating the tracking results to the virtual world for graphical rendering.

There were three modes of object manipulation: 1) using the real objects, 2) using the RT-based shapes, and 3) using the Vive controllers. For the RT-based shapes, the participants were given the shapes that they created and chose as the best match with the VR models in the first pilot study.

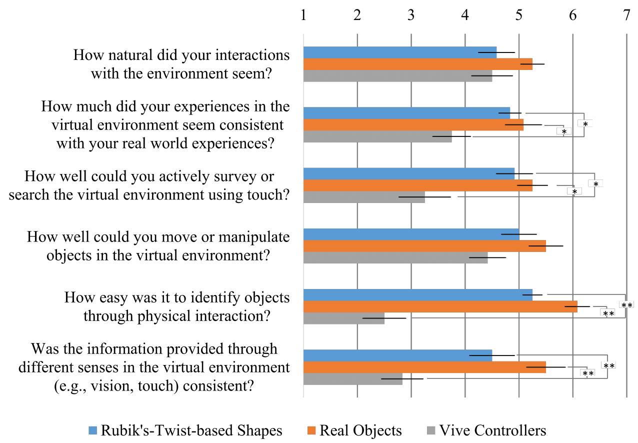

Figure 10: Participants' questionnaire ratings on the haptic experience in VR in Pilot Study 2. *: p < 0.05, **: p < 0.005.

Task and Procedure Similar to the first pilot study, each session included one participant and one moderator. The moderator first taught the participant how to use HTC Vive controllers. The participant then went through three sub-sessions of VR interaction representing the three modes of object manipulation. In each sub-session, he/she could freely interact with the virtual objects by picking up, touching, moving, and rotating the objects. The virtual objects were shown one by one in a random order. The participant could spend as much time as they wanted with each object, and informed the moderator when he/she wanted to change. After interacting with all four objects, the participants responded to the haptic-related questions from the presence questionnaire [48], and moved to the next mode. The visual and auditory feedback was the same across the three modes, and the three modes was presented in a Latin-square-based counterbalanced order.

Results Friedman Test showed the type of the control mode significantly affected the perceived consistency of VR and real world ($\chi^2(2) = 9.8$, $p < 0.05$), capabilities of touch-based interaction ($\chi^2(2) = 6.82$, $p < 0.05$), ease of object identification through touch ($\chi^2(2) = 19.6$, $p < 0.005$), and consistency of the multisensory information in VR ($\chi^2(2) = 9.95$, $p < 0.05$). Fig. 10 shows the descriptive results, along with the results of post-hoc pairwise comparison, of user ratings on sense of presence related to haptics in VR.

5 Discussion on Pilot Studies

When being used as VR haptic proxies, the RT-based physical shapes provided significantly better sense of presence for various VR hand-graspable objects over commercial VR controllers. This suggested the feasibility of using RTs for the creation of VR haptic proxies. On the other hand, the first pilot study revealed the difficulties users may encounter during the creation of RT-based shapes. The participants also commented that while the physical shapes provided a realistic haptic sensation, they lacked interactivity (e.g., triggering), especially for active VR objects, such as guns. These results of the two pilot studies suggested a need for guidance and scaffolding for physical shape creation and interactivity annotation for RT-based VR haptic proxies.

6 HapTwist Toolkit

To overcome some of the limitations mentioned above and facilitate the creation of RT-based VR haptic proxies, we developed HapTwist. The toolkit runs on Unity3D 2017.1.1f1 and HTC Vive. It consists of an algorithm (Fig. 11f) that generates RT-based structures, and a set of software scaffoldings (Fig. 11a - c) and hardware modules (Fig. 11d, e, g - j) to support the interactivity of haptic proxies.

Software

We designed an algorithm for generating the RT shapes based on the geometric structures of the 3D models. As the preprocessing step, we assumed that the 3D model was designed with the real-world dimensions, and the bounding box of the 3D model was obtained. As shown in Fig. 10f, the algorithm first voxelized the 3D model in the form of 6-connected voxel collections with $128 \times 128 \times 128$ resolution using a standard ray-triangle intersection calculation [34]. The model's 3D skeleton was attained based on the voxel collection according the shape of the central cross section of the 3D model. The voxelized structure was then segmented based on the skeleton tree [39]. With the skeleton tree and the segmented voxel collection, the algorithm started the repetitive calculation of proper placements of the 3D prisms, which are the basic elements of the Rubik's Twist. We defined the dimension of the 3D prism as W1.22*H0.61*D0.94 inches (Fig. 2b). The placement process started from the end node in the skeleton farthest from the current camera view. The first prism was placed at the position of this node, and oriented towards the first neighboring node of the starting node in the skeleton.

With the results of the preprocessing step, we parameterized the to-generate prism's coverage of the voxels, and adopted a recursive algorithm, as elaborated in Fig. 13, to place the rest of the triangular prisms in the RT-based shape. For the placement of the next prism, the RT rule allows for four types of rotations and translations of the prism, denoted as $P_{possible} = \{p_0, p_{45}, p_{90}, p_{135}\}$, as shown in Fig. 12. The numbers indicate the rotation angle of the to-generate prism relative to the current prism. The placement of the next prism was determined by the placement score for each possible placement. As shown in Fig. 13, the placement score was calculated by the hyperbolic-paraboloid combination of the number of covered voxels and "ray-cast" hit voxels, to made the difference among different placement more observable, compared to the linear combination. The current algorithm stops when all voxels are covered. The algorithm will first

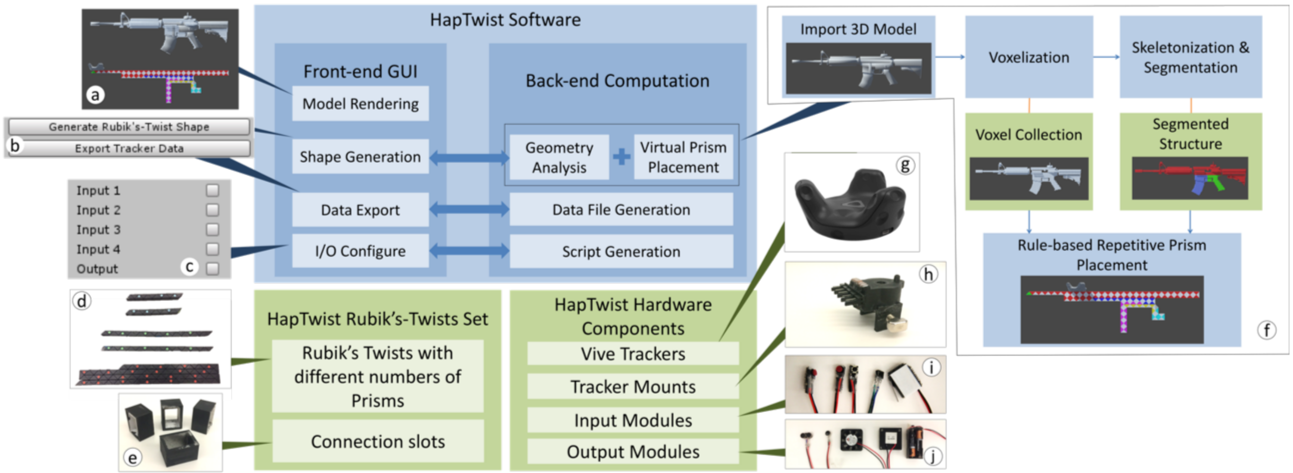

Figure 11: System Diagram of HapTwist: (a) 3D model rendering in Unity3D, (b) Graphical user interface (GUI) for shape generation and data export, (c) GUI for I/O settings on individual prism, (d) Rubik's Twists (one 108-piece, two 24-piece, and two 12-piece), (e) Connection slots for Rubik's-Twist Assembly, (f) Algorithmic pipeline for shape generation, (g) Vive tracker, (h) tracker mount with POGO pin connections, (i) Input modules: push buttons and trigger, (j) Output modules, left to right: dual vibrators, single vibrator, fan, thermo-electric (TEC) module, power supply for fan and TEC module.

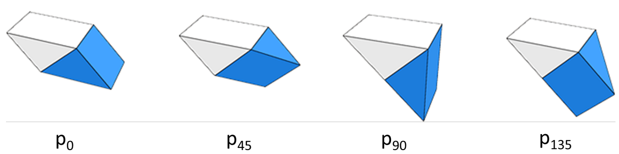

Figure 12: Possible arrangement of position and rotation between two Rubik's-Twist prisms.

use the shortest RT (i.e. standard 12-piece), then the longer ones, in order to minimize the number of the extra pieces.

We implemented the algorithm as a Unity plug-in with the feature of importing 3D models. Fig. 10 b &c show the key graphical-user-interface elements of the plug-in, supporting the generation of RT-based shape, the data/script export, and the configuration of interactivity. As the generated shape may involve the assembly of multiple RTs, the plug-in computes the number and the types of RTs needed and highlights each RT in a different color. The plug-in also labels the prisms with number IDs to assist the identification of the twisted positions in the real objects. Fig. 14 shows the examples of RT-based shapes generated by the HapTwist software for various 3D models.

The plug-in also allows the user to indicate the placement of the Vive tracker and the hardware I/O components on the generated shape. Furthermore, the plug-in generates a C#-based scripting template for integrating the haptic proxy with the user's VR project. Please refer to the supplementary video for a demonstration of the software interface.

Hardware

HapTwist hardware modules include a set of RTs Twists with different numbers of triangular prisms (Fig. 11d), Vive trackers (Fig. 11g) with 3D-printed mounting structures (Fig.

Algorithm: prismPlacement

prismPlacement (*P*, *V*, *T*)
Input: A list *P* of currently generated prism
A list *V* of voxels generated based on the 3D model
A skeleton tree *T* in the form of list
Output: An updated list P_{update} with a newly generated prism

if (all skeleton nodes in *T* has been covered **or** all voxels in *V* has been covered)
return *P* as P_{update}

get the last prism p_{last} in *P*.
create the list $P_{possible}$ of possible placement for the new prism, denoted as $P_{possible} = \{p_0, p_{45}, p_{90}, p_{135}\}$, according to the relative position and orientation to p_{last} as shown in Fig. 12.

for each p_k in $P_{possible}$, $k \in \{0, 45, 90, 135\}$
compute the placement score
$ps_k = (1 - col_k) * cov_k * r_k$
where
col_k denotes whether p_k collides with existing prisms in *P* (1 – collided, 0 – no collided);
cov_k denotes the number of voxels covered by p_k;
r_k denotes the number of hit voxels by ray-casting the shape of p_k towards the next uncovered node in *T*.

find p_{max} in $P_{possible}$ for the maximum value ps_k.

instantiate the first prism p_{new} at p_{max}*.position()* with the orientation of p_{max}*.orientation()*.
label the voxels covered by p_{new} as *beCovered.*
if (p_{new} covers an uncovered skeleton node *N* in *T*)
label *N* as *beCovered*

P_{update} = *P.insert*(p_{new})
prismPlacement (P_{update}, *V*, *T*)

Figure 13: Pseudo code of RT-based shape generation.

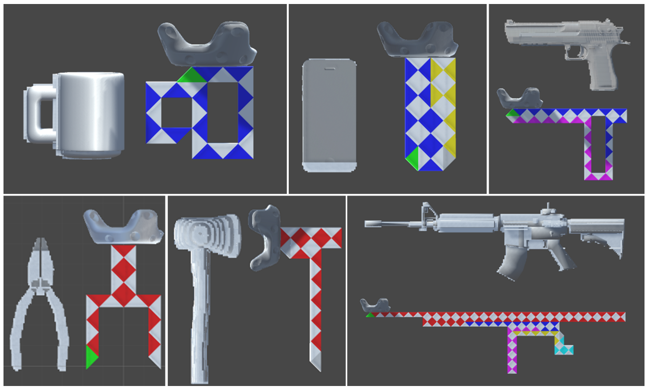

Figure 14: Examples of algorithm results.

11h), and a list of input components (Fig. 11i, e.g., push buttons, switches, and triggers) and output components (Fig. 11j, e.g., flat vibrators, fans, and thermo-electric elements).

The strength of the RT-based construction may reduce as the length increases, because the lever distance increases. In the current toolkit, we designed a set of connection slots (Fig. 11e) , allowing users to firmly concatenate multiple RTs. We also provided Velcro tapes for the participants to fasten the RT parts when needed.

Upon finishing the physical shape construction, the user can fix the tracker on the shape by fastening the screw in the mounting structure. The tracker-mounting structures provide female-style output connection and male-style input connection for the POGO pins in the Vive trackers, allowing easy connection to the I/O components. Fig. 15 shows the examples of HapTwist-generated haptic proxies.

7 Toolkit Evaluation

We evaluated the effectiveness of using the HapTwist-generated shapes as VR haptic proxies and the usability of the toolkit for creating interactive VR haptic proxies. We were interested in the following hypotheses:

H1: While being used as VR haptic proxies, the HapTwist-generated shapes can provide the sense of presence on haptics (i.e., user's perceived consistency of VR and real world, capability of touch-based interaction, ease of object identification via touching, and consistency of the multisensory information in VR) with no significant difference from the real objects.

H2: HapTwist can significantly improve user performance (e.g., time spent) on creating RT-based VR haptic proxies, compared to the situation without HapTwist.

H3: HapTwist can significantly improve user experience (e.g., perceived workload, ease, and enjoyment) on creating RT-based VR haptic proxies, compared to the situation without HapTwist.

Participants

Twelve participants, with an average age of 24.4 years old (SD = 2.93), were invited. To avoid potential bias in users' subjective ratings, we adopted the following strategies as suggested in [12]: 1) all these participants didn't attend the pilot studies, but with similar experience on RT and VR as the pilot-study participants; 2) the study facilitator was from the same ethnic group as the participants; 3) the facilitator didn't know the participants in person beforehand, and didn't explicitly disclose any personal association with the toolkit.

Figure 15: HapTwist-generated Proxies and their real-object counter parts.

Seven participants had previous experience with HMD-based VR, while the rest had never tried but saw others playing VR and HMD before. Ten participants indicated that they had never used RT before, while all of them reported previous experience on 3D content creation (e.g. games and animations).

Apparatus

To study the effectiveness of the generated shapes as VR haptic proxies, we prepared a set of RT-based proxies generated by HapTwist and the real objects for the four virtual objects used in our pilot studies. Both types of physical proxies were equipped with Vive trackers and hardware triggers/buttons.

To investigate the usability of HapTwist, we provided each participant a complete set of the HapTwist toolkit: Unity plug-in, four RTs of different lengths, Vive trackers, hardware I/O modules, connection slots, and the HTC Vive HMD. The virtual-world prototype was used as the sample Unity project for integrating the HapTwist-generated proxies. In addition, the participants were given a printed tutorial on how to use HapTwist step by step. The tutorial used the 3D model of a pair of pliers as the example.

Task and Procedure

Each session included one participant and one facilitator. The participant needed to complete two activities: 1) experience the HapTwist-generated shapes and the real objects as

VR haptic proxies, and 2) use HapTwist to create interactive VR haptic proxies.

All the apparatuses were hidden from the participant's sight before the activities to avoid any bias. In the first activity, the participant went through two sub-sessions of VR interaction representing two modes of object manipulation using: 1) the HapTwist-generated proxies and 2) the real-object proxies. Similar to our pilot study, he/she could freely interact with the virtual objects. The virtual objects were shown one by one in a random order. After interacting with each object, the participants took off the HMD, rated the haptic-related questions from the presence questionnaire [48], and moved to the next object. The visual and auditory feedback was the same across the two modes, and the two modes of haptic proxies were presented in a Latin-square-based counterbalanced order.

In the second activity, the participant first followed the step-by-step tutorial to create the interactive haptic proxy for a pair of pliers. He/she then continued to use HapTwist to create the haptic proxies for the four selected virtual objects, and integrated the proxies into the sample VR project. This was assigned in counterbalanced order based on Latin Square. After the second activity, the participant rated their perceived ease and joy of haptic-proxy creation, and NASA-TLX-based workload, in a 7-likert scale.

Results

H1: HapTwist-Generated Shapes as VR Haptic Proxies Mann-Whitney U Test showed that for all four virtual objects, there was no significant difference between using HapTwist-generated shapes and the real objects in terms of the participants' responses to the haptic-related questions in the presence questionnaire. The effect sizes, *r*, for all haptic-related questions were smaller than 0.1. This indicated the difference between HapTwist-generated proxies and real-world objects was trivial in terms of users' responses to the haptic-related questions, further validating H1.

H2 & H3: Using HapTwist to Create Haptic Proxies The participants generally rated that it was easy to follow the provided tutorial and learn to use the toolkit (6.0/7, SD = 0.90), use the toolkit to generate the shape of RTs (5.9/7, SD = 0.91), follow the software outcomes to create the RT-based physical shapes (5.7/7, SD = 1.07), and integrate the physical RT-based shapes as the haptic proxies to the VR scenario (5.6/7, SD = 1.09). In addition, they found it joyful to use the toolkit (6.2/7, SD = 0.86), and the satisfaction on the toolkit outcome averagely scored 5.8/7 (SD = 1.01).

Fig. 16 illustrates the comparison between the conditions with and without HapTwist. To examine whether the HapTwist toolkit could facilitate the process of haptic-proxy creation, we compared the time the participants spent and the workload on shape creation for the four virtual objects

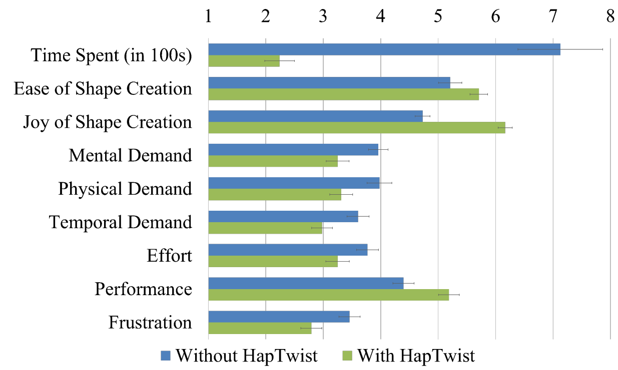

Figure 16: Comparison between the conditions without and with HapTwist.

with HapTwist and without HapTwist (previous study). The independent-samples T-Test showed that in overall, HapTwist significantly shortened the time spent ($t(94) = -6.25$, $p < 0.005$). Two-way ANOVA showed an interaction effect of the usage of the toolkit and the type of the virtual object on the time spent on the physical shape creation ($F(3, 88) = 3.39$, $p < 0.05$, $\eta^2 = 0.104$). Post-hoc pairwise comparison revealed that HapTwist significantly shortened the time spent for creating the machine-gun model ($p < 0.005$), the mug model ($p < 0.005$), and the phone model ($p < 0.005$), while the reduction in working time for the pistol model was marginal ($p = 0.052$).

Mann-Whitney U Test showed HapTwist significantly reduced the perceived mental demand ($U = 949$, $p < 0.05$), physical demand ($U = 950.5$, $p < 0.05$), temporal demand ($U = 961$, $p < 0.05$), and frustration ($U = 896.5$, $p < 0.05$) during the process of RT-based shape creation. HapTwist also significantly improved the perceived performance ($U = 881.5$, $p < 0.05$) and joy ($U = 520$, $p < 0.005$). In our pilot study on creating RT-based shapes without HapTwist, many participants commented that "*it was difficult to figure out where to start*", "*I feel a little bit rushing and short of ideas on constructing the model*", "*I feel hard to rotate as the size of model becomes bigger*". While using HapTwist, all the participants suggested that the software outcome was "*helpful*", and that "*you don't need to be an expert on Rubik's Twist to make the shapes*".

8 Discussion

Reusing HapTwist-Generated Physical Shapes

While the HapTwist-generated shapes offered a similar VR experience to the selected real objects, we were also interested in the possibility of reusing the generated shapes to haptically represent other virtual objects. In the questionnaire for after the usage of HapTwist, the participants listed possible alternative objects as follows:

- Mug: brass knuckles, purse, vase, and teapot.
- Pistol: driller, door handle, crutch, hammer, axe, crossbow, and hoe.

- Mobile Phone: mirror, remote controller, photo frame, and cigarette box.
- Machine Gun: any long gun, sword, and guitar.

The reusability of HapTwist-generated shapes reduces the need of creating multiple haptic proxies. One HapTwist-generated physical shape can be used for different virtual objects under the same or different VR scenarios. For instance, the pistol shape can represent a single-handed weapon inventory, while the machine-gun shape can represent the two-handed weapon inventory. The object switching can be achieved with either a hardware trigger or a scenario change.

Switching among Multiple Physical Shapes

When playing with multiple HapTwist-generated shapes in the same VR scenario, the user needs to put down the current held object before picking up other objects. Therefore, there is a need for a dedicated space, such as tables or walls, to place the physical proxies. In a room-scale VR setting, the proxies can be placed in different locations, and the user needs to physically move and pick up the objects. The user can also carry the proxies on his/her body, such as the pistol/machine guns in holsters and axe fastened on the belt. Although this may place extra space requirements on VR users, it leverages the kinesthetic sensation and movements of the users, potentially improving the VR experience [24].

Limitation

Firstly, while the Vive trackers ensured stable movement tracking and VR rendering, five participants commented that the size of the tracker affected the experience of touching the virtual objects. They indicated the experience of touching the trackers that were not rendered in VR was "*a strange feeling*", especially for the small objects (e.g., mobile phone and mug). These issues may be solved as the tracking technology advances with smaller devices. In addition, the weight of the tracker sometimes caused difficulties on balancing the physical objects while being held. To address this problem, we will improve our algorithm to optimize the tracker placement according to the center mass of the object.

Secondly, it was difficult to replicate the details of some virtual objects using RTs. The current algorithm picks the prism pose which covers and "foresees" (ray-cast hit) the most voxels. In the case of the machine gun, the horizontal prism covers and foresees more voxels than the vertical prism at the position of the scope, so this part of the model was omitted. As future work, users' annotation could be involved to support the complex shape representation. Another solution could be a neutral network for RT shape generation, taking user-created shapes as training sets, to simulate the shape-construction process by the users. However, it would require a specific training set for each virtual object, which may limit its scalability. In addition, the current algorithm could not generate the non-flat 3D shape (e.g. mug) with single RT, as it currently only considers the central cross section of the 3D model. This issue could be potentially solved by sampling the model into layers, and the volumetric shape could be formed by assembling all the RT layers.

Lastly, while HapTwist significantly shortened the time spent on creating interactive haptic proxies using RTs, the time needed increases along with the size of the virtual object (i.e. average: ~1.5 minutes for the mug model, ~2.5 minutes for the mobile-phone model, ~3 minutes for the pistol model, and ~8.5 minutes for the machine-gun model). While VR object switching can be potentially achieved by leveraging other users' physical efforts [7], it may be hard to create shapes in real time for fast-pace VR interaction. In future work, we will investigate twistable structures with embedded electro-mechanical modules for automatic transformation.

9 Conclusion

We present a series of studies on using Rubik's Twist to create VR haptic proxies. Our pilot studies validated the feasibility and the effectiveness of Rubik's-Twist-based haptic proxies, and it also revealed the challenges users encountered in shape creation, motivating the development of the HapTwist toolkit. HapTwist provides an accessible software+hardware platform, and supports the creation of interactive haptic proxies for a comprehensive set of hand-graspable VR objects. Our user study showed that HapTwist was easy to learn and use, and it significantly improved user performance in creating interactive haptic proxies with Rubik's Twist. HapTwist haptic proxies offered a similar VR experience to the real objects. With further development of the toolkit, we aim to bring the joy of customizing interactive VR haptic proxies and controllers with low-cost shape-changable artifacts to more users.

Acknowledgments

This research was partially supported by grant from the Centre for Applied Computing and Interactive Media (ACIM) of School of Creative Media, the Strategic Research Grants (Project No. 7005021 & 7005172), the Teaching Development Grant (Project No. 6000623 & 6000639), City University of Hong Kong, and grants from the Research Grants Council (Project No. CityU 21200216) and the Environment and Conservation Fund (Project No. EECA1956) of the Hong Kong Special Administrative Region, China. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for the high-quality VR rendering in this research.

References

[1] Abtahi, P., & Follmer, S. (2018, April). Visuo-Haptic Illusions for Improving the Perceived Performance of Shape Displays. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (p. 150). ACM.

[2] Aguerreche, L., Duval, T., & Lécuyer , A. (2010, November). Reconfigurable tangible devices for 3D virtual object manipulation by single or multiple users. In Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology (pp. 227-230). ACM.
[3] Azmandian, M., Hancock, M., Benko, H., Ofek, E., & Wilson, A. D. (2016, May). Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In Proceedings of the 2016 chi conference on human factors in computing systems (pp. 1968-1979). ACM.
[4] Benko, H., Holz, C., Sinclair, M., & Ofek, E. (2016, October). Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology (pp. 717-728). ACM.
[5] Bouzit, M., Burdea, G., Popescu, G., & Boian, R. (2002). The Rutgers Master II-new design force-feedback glove. IEEE/ASME Transactions on mechatronics, 7(2), 256-263.
[6] Cheng, L. P., Chang, L., Marwecki, S., & Baudisch, P. (2018, April). iTurk: Turning Passive Haptics into Active Haptics by Making Users Reconfigure Props in Virtual Reality. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (p. 89). ACM.
[7] Cheng, L. P., Roumen, T., Rantzsch, H., Köhler, S., Schmidt, P., Kovacs, R., ... & Baudisch, P. (2015, November). Turkdeck: Physical virtual reality based on people. In Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (pp. 417-426). ACM.
[8] Choi, I., Culbertson, H., Miller, M. R., Olwal, A., & Follmer, S. (2017, October). Grabity: A Wearable Haptic Interface for Simulating Weight and Grasping in Virtual Reality. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (pp. 119-130). ACM.
[9] Choi, I., Hawkes, E. W., Christensen, D. L., Ploch, C. J., & Follmer, S. (2016, October). Wolverine: A wearable haptic interface for grasping in virtual reality. In Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on(pp. 986-993). IEEE.
[10] Choi, I., Ofek, E., Benko, H., Sinclair, M., & Holz, C. (2018, April). CLAW: A Multifunctional Handheld Haptic Controller for Grasping, Touching, and Triggering in Virtual Reality. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (p. 654). ACM.
[11] Craik, F. I., Klix, F., & Hagendorf, H. (2016). A functional account of age differences in memory. Memory, Attention, and Aging: Selected Works of Fergus I. M. Craik (pp. 409-422). Routledge.
[12] Dell, N., Vaidyanathan, V., Medhi, I., Cutrell, E., & Thies, W. (2012, May). Yours is better!: participant response bias in HCI. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1321-1330). ACM.
[13] Feix, T., Romero, J., Schmiedmayer, H. B., Dollar, A. M., & Kragic, D. (2016). The grasp taxonomy of human grasp types. IEEE Transactions on Human-Machine Systems, 46(1), 66-77.
[14] Fiore, A. (1981). Shaping Rubik's Snake. Penguin Australia.
[15] Follmer, S., Leithinger, D., Olwal, A., Hogge, A., & Ishii, H. (2013, October). inFORM: dynamic physical affordances and constraints through shape and object actuation. In Uist (Vol. 13, pp. 417-426).
[16] Fujinawa, E., Yoshida, S., Koyama, Y., Narumi, T., Tanikawa, T., & Hirose, M. (2017, November). Computational design of hand-held VR controllers using haptic shape illusion. In Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology (p. 28). ACM.
[17] Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In Advances in psychology (Vol. 52, pp. 139-183). North-Holland.
[18] Heo, S., Chung, C., Lee, G., & Wigdor, D. (2018, April). Thor's Hammer: An Ungrounded Force Feedback Device Utilizing Propeller-Induced Propulsive Force. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (p. 525). ACM.
[19] Hettiarachchi, A., & Wigdor, D. (2016, May). Annexing Reality: Enabling Opportunistic Use of Everyday Objects as Tangible Proxies in Augmented Reality. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (pp. 1957-1967). ACM.
[20] Hinckley, K., Pausch, R., Goble, J. C., & Kassell, N. F. (1994, April). Passive real-world interface props for neurosurgical visualization. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 452-458). ACM.
[21] Holman, D., & Vertegaal, R. (2008). Organic user interfaces: designing computers in any way, shape, or form. Communications of the ACM, 51(6), 48-55.
[22] Strasnick, E., Holz, C., Ofek, E., Sinclair, M., & Benko, H. (2018, April). Haptic Links: Bimanual Haptics for Virtual Reality Using Variable Stiffness Actuation. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (p. 644). ACM.
[23] Huang, D. Y., Guo, R., Gong, J., Wang, J., Graham, J., Yang, D. N., & Yang, X. D. (2017, October). RetroShape: Leveraging Rear-Surface Shape Displays for 2.5 D Interaction on Smartwatches. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (pp. 539-551). ACM.
[24] Insko, B. E., Meehan, M., Whitton, M., & Brooks, F. (2001). Passive haptics significantly enhances virtual environments(Doctoral dissertation, University of North Carolina at Chapel Hill).
[25] Ishii, H. (2008). The tangible user interface and its evolution. Communications of the ACM, 51(6), 32-36.
[26] Ishii, H., Lakatos, D., Bonanni, L., & Labrune, J. B. (2012). Radical atoms: beyond tangible bits, toward transformable materials. interactions, 19(1), 38-51.
[27] Jackson, B., Lau, T. Y., Schroeder, D., Toussaint, K. C., & Keefe, D. F. (2013). A lightweight tangible 3D interface for interactive visualization of thin fiber structures. IEEE transactions on visualization and computer graphics, 19(12), 2802-2809.
[28] Jang, S., Kim, L. H., Tanner, K., Ishii, H., & Follmer, S. (2016, May). Haptic edge display for mobile tactile interaction. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (pp. 3706-3716). ACM.
[29] Kim, H., Coutrix, C., & Roudaut, A. (2018, April). Morphees+: Studying Everyday Reconfigurable Objects for the Design and Taxonomy of Reconfigurable UIs. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (p. 619). ACM.
[30] Lindeman, R. W., Sibert, J. L., & Hahn, J. K. (1999, March). Handheld windows: towards effective 2D interaction in immersive virtual environments. In Virtual Reality, 1999. Proceedings., IEEE (pp. 205-212). IEEE.
[31] Lo, J. Y., Huang, D. Y., Sun, C. K., Hou, C. E., & Chen, B. Y. RollingStone: Using Single Slip Taxel for Enhancing Active Finger Exploration with a Virtual Reality Controller. In Proceedings of UIST 2018. ACM.
[32] MacKenzie, C. L., & Iberall, T. (1994). The grasping hand (Vol. 104). Elsevier.
[33] McClelland, J. C., Teather, R. J., & Girouard, A. (2017, October). HaptoBend: shape-changing passive haptic feedback in virtual reality. In Proceedings of the 5th Symposium on Spatial User Interaction (pp. 82-90). ACM.
[34] Möller, T., & Trumbore, B. (2005, July). Fast, minimum storage ray/triangle intersection. In ACM SIGGRAPH 2005 Courses (p. 7). ACM.

[35] Mueller, S., Mohr, T., Guenther, K., Frohnhofen, J., & Baudisch, P. (2014, April). faBrickation: fast 3D printing of functional objects by integrating construction kit building blocks. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 3827-3834). ACM.

[36] Perry, J. C., Rosen, J., & Burns, S. (2007). Upper-limb powered exoskeleton design. IEEE/ASME transactions on mechatronics, 12(4), 408-417.

[37] Provancher, W. (2014). Creating greater VR immersion by emulating force feedback with ungrounded tactile feedback. IQT Quarterly, 6(2), 18-21.

[38] Rasmussen, M. K., Pedersen, E. W., Petersen, M. G., & Hornbæk, K. (2012, May). Shape-changing interfaces: a review of the design space and open research questions. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 735-744). ACM.

[39] Reniers, D., & Telea, A. (2007, June). Skeleton-based hierarchical shape segmentation. In Shape Modeling and Applications, 2007. SMI'07. IEEE International Conference on (pp. 179-188). IEEE.

[40] Schweikardt, E., & Gross, M. D. (2008, February). The robot is the program: interacting with roBlocks. In Proceedings of the 2nd international conference on Tangible and embedded interaction (pp. 167-168). ACM.

[41] Sheng, J., Balakrishnan, R., & Singh, K. (2006, November). An interface for virtual 3D sculpting via physical proxy. In GRAPHITE (Vol. 6, pp. 213-220).

[42] Simeone, A. L., Velloso, E., & Gellersen, H. (2015, April). Substitutional reality: Using the physical environment to design virtual reality experiences. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (pp. 3307-3316). ACM.

[43] Strobel, J. (2010). All the better to see you with: A comparison of approaches to delivering instructions for Lego construction tasks (Doctoral dissertation, Bowling Green State University).

[44] Strohmeier, P., & Hornbæk, K. (2017, May). Generating Haptic Textures with a Vibrotactile Actuator. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (pp. 4994-5005). ACM.

[45] Teng, S. Y., Kuo, T. S., Wang, C., Chiang, C. H., Huang, D. Y., Chan, L., & Chen, B. Y. PuPoP: Pop-up Prop on Palm for Virtual Reality. In Proceedings of UIST 2018. ACM.

[46] Weller, M. P., Gross, M. D., & Do, E. Y. L. (2009, April). Tangible sketching in 3d with posey. In CHI'09 Extended Abstracts on Human Factors in Computing Systems (pp. 3193-3198). ACM.

[47] Whitmire, E., Benko, H., Holz, C., Ofek, E., & Sinclair, M. (2018, April). Haptic Revolver: Touch, Shear, Texture, and Shape Rendering on a Reconfigurable Virtual Reality Controller. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (p. 86). ACM.

[48] Witmer, B. G., & Singer, M. J. (1998). Measuring presence in virtual environments: A presence questionnaire. Presence, 7(3), 225-240.

[49] Yem, V., & Kajimoto, H. (2017, March). Wearable tactile device using mechanical and electrical stimulation for fingertip interaction with virtual world. In Virtual Reality (VR), 2017 IEEE (pp. 99-104). IEEE.

[50] Zenner, A., & Krüger, A. (2017). Shifty: A Weight-Shifting Dynamic Passive Haptic Proxy to Enhance Object Perception in Virtual Reality. IEEE transactions on visualization and computer graphics, 23(4), 1285-1294.

[51] Zhao, Y., Kim, L. H., Wang, Y., Le Goc, M., & Follmer, S. (2017, October). Robotic Assembly of Haptic Proxy Objects for Tangible Interaction and Virtual Reality. In Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (pp. 82-91). ACM.