


Stuet: Dual Stewart Platforms for Pinch Grasping Objects in VR

Ulan Kelesbekov , Gabriele Marini, Zhongyi Bai, Wafa Johal, Eduardo Velloso, and Jarrod Knibbe

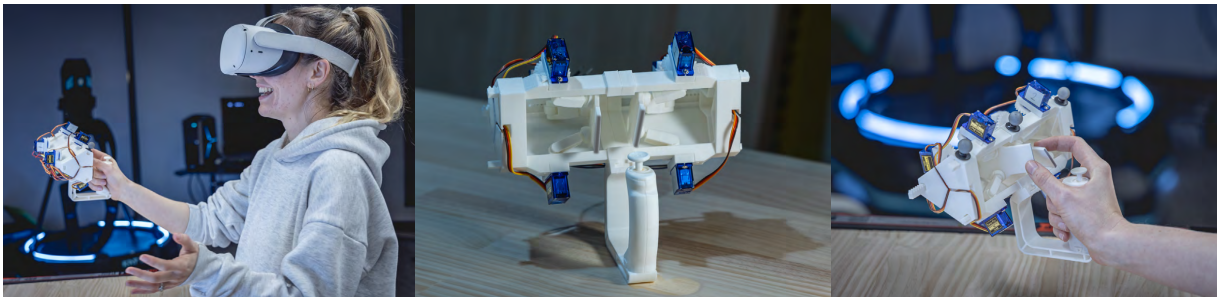


Figure 1: Stuet, a novel twin Stewart Platform haptic controller for thumb-index finger grasping and exploration in virtual reality. The controller includes a grip for the hand and two platforms for thumb-finger interaction.

ABSTRACT

Complex 3D shapes' surfaces can be characterised using three shape descriptors: zeroth-order for rendering width; first-order to convey slope; and second-order for curvature. These shapes can be symmetric or asymmetric. To date, controllers in VR have been unable to render these properties in 3D. We present Stuet - a handheld virtual reality controller that can render complex asymmetrical 3D objects for two-finger grasping and shape exploration. Stuet leverages dual 3 degrees of freedom (3-DOF) Stewart Platforms. This enables the contact plates for the fingers to be controlled individually, rendering objects widths up to 75 mm and individual plate angles up to 30° in any tilt direction with respect to the vertical plane. We present the design and implementation of Stuet. We explain and benchmark its mechanical capabilities, present the inverse kinematics model required for its use, and report on a feasibility demonstration. Our results reveal that dual Stewart platforms offer new capabilities for asymmetric, advanced haptic interactions in VR.

Keywords: Haptics, Controllers

1 INTRODUCTION

Grasping virtual objects is a canonical VR interaction. However, providing real-time sensory feedback for this action is non-trivial, and current off-the-shelf controllers support this poorly. Commercial solutions like the Quest and Vive controllers provide little or low-resolution feedback for grasping actions, typically either visually coupling the object to the controller (*visual feedback*) or by providing collision-based vibration (*haptic feedback*). This, however, fails to provide insight into the geometric properties of the virtual object.

Many research prototypes have attempted to simulate shape-based information, including position and orientation [7, 14, 18, 22]. To do this effectively, controllers must change shape at runtime, so recent work turned towards designing active, shape-changing, actuated devices. Recent examples include exoskeletons (e.g., [15, 18]), fingertip attachments [34], handheld devices [7, 14], and even drones [16].

However, these devices have limitations: exoskeletons are broadly impractical to wear; drones raise safety concerns; handheld devices have inherently constrained degrees of freedom; among other issues. Motivated by the popularity and prevalence of handheld controllers

for VR, we introduce a handheld controller that simulates grasping more complex shapes.

To date, most controller designs have been optimised for grasping as a means to lift and hold an object. This action requires balancing various factors [40], leading people to grasp around parallel or close-to-parallel points (allowing force closure, preventing slipping, etc.) This is not the case for exploring objects, when your fingers may trace non-parallel edges, nor when attempting to grasp objects without direct visual attention, when arbitrary, non-parallel points may be touched. To facilitate this broader interaction space, we must be able to render asymmetric shapes physically.

We present Stuet, a double-sided, Stewart platform-based, high-degree-of-freedom, handheld controller for rendering complex, asymmetrical shapes for grasping and interaction. Stuet is designed to be held in one hand with a normal grip and consists of a pair of 3-degree-of-freedom (3-DOF) motion platforms actuated by servo motors. Each platform can independently actuate its 'height' (heave) and two axes of rotation (pitch and roll).

When coupled, these motion platforms provide the sensation of grasping a virtual object with the index finger and thumb fingerpads, known as a multi-pulpar grasp [13]. Because our design is tailored for two fingers specifically, we will refer to this grasp simply as a *pinch grasp*. Our twin platform system can render different sizes and shapes dynamically, according to the requirements of the VR application.

We make the following contributions: first, we provide a theoretical framing for designing haptic controllers for complex shapes, discussing conceptual finger models and shape descriptors. Next, we present the design, implementation, and technical evaluation of Stuet, a virtual reality controller for rendering complex, asymmetrical shapes. This includes both the joint design for the mechanisms and the inverse kinematics model required for its use. Finally, we present the results of a feasibility demonstration and discuss the novel capabilities that this new kind of haptic platform can offer for interaction with objects in VR.

2 RELATED WORK: DESIGNING HIGHER-RESOLUTION HAPTIC CONTROLLERS

'Higher-resolution' haptic controllers are those that convey greater information to the body about the objects being interacted with, beyond simple visual cues or contact-based vibration. Early controllers that aimed to achieve this were typically grounded or fixed to a surface. Devices such as the PHANTOM [29], HIRO [9], and SPIDAR [37] offered high-resolution feedback. However, more recent trends in VR development have emphasised portability and

Table 1: Comparisons of different haptic controller devices in terms of the fingertip model that fits the design, possible shape dimensions rendered, shape orders, asymmetry, and grasping design paradigm.

Controller	Fingertip Model	Shape Dim.	Shape Rendering Order			Asymmetry	Design Paradigm
			0th	1st	2nd		
NormalTouch [2]	disc	2.5D	Yes	Yes	Partial	N/A	N/A
TextureTouch [2]	def. surface	2.5D	Yes	Yes	Yes	N/A	N/A
HUGO [17]	def. surface	2.5D	Yes	Yes	Yes	N/A	N/A
TapeTouch [49]	def. surface	2.5D	Yes	Yes	Yes	N/A	N/A
TORC [24]	point	3D	No	No	No	N/A	Inside-Out
Claw [7]	point	3D	Partial	No	No	No	Outside-In
CapstanCrunch [39]	point	3D	Partial	No	No	No	Outside-In
SpinOcchio [22]	point (for shape)	3D	Yes	No	No	No	Outside-In
X-Rings [14]	point	3D	Yes	Partial	Partial	No	Inside-Out
PaCaPa [42]	point	3D	Yes	Partial	No	Yes	Inside-Out
Stuet	disc	3D	Yes	Yes	Partial	Yes	Outside-In

more mobile exploration. As a result, in many ways, the research community has been working to recreate the resolution of grounded devices in portable form factors.

There are three primary forms of portable controllers: *wearable*, *handheld*, and *just-in-time*. Typically, wearable and handheld controllers are in constant contact with the users’ hands, while just-in-time controllers are only available when interaction is taking place. We briefly examine the controller types in turn, discussing their strengths and weaknesses, before motivating our exploration of handheld controllers.

2.1 Wearable Controllers

Wearable controllers, such as exoskeletons and gloves, have many benefits. High-resolution examples can easily control and constrain individual finger movement, such as DextrES [18] and ExoGlove [20]. Further, wearing such a device opens opportunities for mounting components and actuators on the back of the hand and the forearm (e.g., [21]), providing higher haptic fidelity and, thus, enabling more sophisticated interactions.

Another type of wearable controllers are “thimble” style haptic devices. These works focus on providing haptic feedback directly onto the fingertips with various end-effectors including flat platforms [12, 34, 41] and pin arrays [23, 35, 36] for rendering normal forces and orientation; and rigid tactors [25, 26, 44] and soft fabrics [31, 33] for conveying both normal and shear forces. Many of these devices are high-resolution and can render multiple perceivable haptic features simultaneously in a compact form factor. However, as these devices are ungrounded, their primary shortcoming is a lack of rigid normal forces.

The complexity and weight of the mechanisms of wearable controllers can make them cumbersome, and donning and removing such devices can easily become burdensome, which explains why we have yet to see the broad adoption of such devices in any domain.

2.2 Just-in-time controllers

Just-in-time controllers, which belong to a class of encounter-type haptic displays [30], are currently a rarer and more innovative approach. Such devices include drones that bring textures and objects to the user during interaction [19, 47]. This is an exciting fledgling opportunity, potentially allowing for greater controller complexity and increased realism, if the underlying safety concerns can be addressed. Moreover, they also require a larger area for interactions and sophisticated robust control algorithms, making them impractical for day-to-day VR use.

2.3 Handheld controllers

Finally, then, there are *handheld* controllers. These include shape-changing controllers [14], handheld drone controllers [16], weight-shifting controllers [38, 48], texture controllers [2, 27, 45], and more.

One popular handheld controller developed in the field is NormalTouch (and its twin, TextureTouch) by Benko et al [2]. It is one of the first devices to bring together the portability and comfort of conventional VR controllers with higher fidelity haptic feedback. Its design incorporated a miniature 3-DOF motion platform illustrating one way to simulate normal forces onto one finger when exploring a surface. TextureTouch expanded upon this to simulate surface texture using a 4x4 pin array on a user’s index finger [2]. Following this, the field started to see more handheld haptic controllers. HUGO, for example, cleverly combines a pin array and motion platform mechanism to render a higher fidelity texture sensation [17]. For all three controllers, the feedback is provided onto a single finger, while the motors, linkages, and joints, take up a lot of hand grasp space. From a size ratio standpoint, all of their mechanisms are a lot larger than the range of motion of the end-effector, raising scalability concerns that are hard to address. It is unclear how to expand the work to include grasping and exploring shapes with more fingers while maintaining high haptic resolution. Where should actuation mechanisms be placed relative to the user’s hand? Grasping controllers like PaCaPa [42] encapsulate their mechanisms in the user’s hand, but have relatively low haptic resolution due to a limited number of DOFs.

Overall, handheld controllers, especially when focusing on high-resolution shape information, have a variety of constraints: there is often less space for storing actuators and components, and working within the grasp of the hand limits the scope for actuation. Nevertheless, their handheld form factor offers a broader range of user interfaces like buttons, thumbsticks, triggers, and vibration motors, as well as allowing 6-DOF motion in space. Despite their haptic rendering limitations, this added versatility is essential for general-purpose VR usage, making them more attractive than the other types of haptic devices. In light of the observed real-world popularity of hand-held controllers, we believe this style of devices warrants further exploration.

3 PROPERTIES OF 3D SHAPES FOR GRASPING

Research on grasping simulation using handheld devices has focused on rendering simple, typically symmetric or axisymmetric shapes (e.g., [14, 22]). However, real-world objects have far more sophisticated shapes that cannot yet be haptically simulated. The rendering of increasingly complex shapes typically requires increasingly complex mechanisms. For our exploration, we focus on two-finger pinch

grasp interactions. First, this reduces the independent contact points for which we need to design, providing a feasible constraint on mechanical complexity. Second, most VR controllers primarily rely on two-finger interactions, using the thumb for buttons and thumbsticks and the index finger for triggers – we adopt and extend this interaction style. Finally, grasping virtual objects with handheld controllers creates no apparent weight change, thus seemingly heavy objects can be easily lifted with just two fingers.

As we pursue higher-fidelity shape rendering for pinch grasping, we first discuss conceptual models of the fingertip contact point and describe the relevant geometric and dynamic properties required to render 3D shapes for grasping using these models. This theoretical framing both situates our design within the literature and provides a lens for future comparisons of devices.

3.1 Fingertip Contact Models

When designing haptic feedback for pinch grasping or interaction with virtual surfaces, we conceptually simplify how the fingertip contacts the device. This results in three prevalent fingertip contact models: *fingertip-as-a-point*, *fingertip-as-a-disc*, and *fingertip-as-a-deformable-surface*. How designers choose to treat the contact of the fingertip with the haptic device directly impacts how shapes are perceived and the mechanical complexity of the haptic controller.

Fingertip-as-a-point.

The simplest way to conceptualise how a finger contacts a surface is as a single point: vertex-to-vertex contact. In effect, this serves to provide only one axis of information, resulting in contact with points, planes, and curved surfaces being indistinguishable. This is the approach taken (for shape rendering) in Grabity [5] and SpinOcchio [22], for example. An illustration of this model is shown in Figure 2-a.

Fingertip-as-a-disc.

To add further contact information, we can consider our fingertips as a disc. This small planar circle lies along any contact plane, revealing both contact location and orientation (or the normal of the tangible surface). NormalTouch [2], for example, treats the fingertip-as-a-disc.

Fingertip-as-a-deformable-surface.

In reality, however, we perceive more information when our fingers contact surfaces. Upon contact, our fingers deform to conform to the tangible surface. This serves to reveal information about points, edges, planes, and complex surfaces (including different textures). While there are haptic devices built with this model in mind (such as a deformable strip for curvature-rendering in 1D [49] and pin-based arrays for non-smooth objects in 2.5D [2, 11]), their poor scalability, high complexity, and lack of compactness make the development of these devices complex and place limitations on their integration into handheld controllers.

3.2 Complex Shape Descriptors

Alongside understanding how we conceptualise interaction between the fingertip and a 3D object, we must also understand the properties of rendering complex shapes. When exploring a smooth surface (e.g., a Bézier surface, which we'll term a 2.5D shape) with one finger, we can perceive three shape-related haptic cues, described as *orders* of shape descriptors; **zeroth order**: *elevation*, **first order**: *slope*, and **second order**: *curvature* [46]. These attributes also apply to 3D shapes.

Zeroth-Order Shape Descriptor

For a given smooth surface, the height of a point is the zeroth-order shape descriptor. In the context of one-handed grasping and 3D objects, this descriptor relates to the object's size. As seen in Table

1, most haptic controllers developed in the research field incorporate size (or 0th order) rendering [6, 7, 14, 22].

First-Order Shape Descriptor

The slope of a point on a 2.5D surface is the first-order descriptor of its shape. NormalTouch by Benko et al. [2] is a good example of a haptic controller that can render this shape descriptor for a surface. First-order descriptors in 3D allow for the definition of more complex shapes and features, such as edges, overhangs, pyramids, and prisms. X-Rings can render this descriptor for some axisymmetric shapes in 3D [14]. Even though these shapes are fundamental, no current device can render them all due to the requirement for asymmetry (see 3.3).

Second-Order Shape Descriptor

The curvature of a smooth surface is the final shape descriptor. A sense of curvature is hard to render, especially for surfaces with highly varying curvature constants. To achieve true second-order rendering, the finger needs to have a sense of the curvature without moving. One approach to approximating curvature rendering is through rectangular pins (pixels, or *texels*) [2, 11]. The limitation is that these require highly sophisticated mechanisms that do not scale well with increased resolution. Another popular approach to *approximating* second-order shape is coupling surface actuation to movement, as in NormalTouch [2]. As the user moves, the surface normal can be changed to let the finger follow a curve. This approach does not provide instantaneous curvature feedback and is, thus, not true second-order shape rendering. To the best of our knowledge, only TapeTouch [49] achieves true second-order rendering.

3.3 Asymmetric Objects

When transitioning from 2.5D to 3D and focusing on pinch grasping using the index finger and thumb, it becomes essential to account for asymmetric objects as well. This consideration enables the exploration of virtual objects from various perspectives. For instance, when engaging with a cube, the sensation of grasping it by one of its faces would differ from grasping it by an edge or a vertex. To achieve asymmetrical rendering for multiple digits, it is necessary to independently actuate the contact surface of each digit. This approach also facilitates the rendering of shapes with diverse geometries and sizes. Figure 2 illustrates the conceptual idea behind asymmetric and curved object rendering. In the illustrations, the third object, curved and asymmetric, requires more sophisticated fingertip model considerations and, therefore, higher fidelity rendering techniques.

3.4 The Link Between Fingertip Models and Shape Orders

The haptic perceptual system is an active process (kinesthetic), as opposed to tactile perception which is a passive system [43]. By actively exploring an object with our hands, we can get more information about its physical properties. For shape in particular, exploration captures objects' shape descriptors more accurately, as people can integrate perception over time. All handheld haptic controllers, such as those listed in Table 1, leverage this to some degree. For example, SpinOcchio, in rendering size and slip sensations, uses two parallel plates that can vary their width over time to render zeroth- and *approximate* first- and second-order shape descriptors [22]. Although this controller's primary focus is rendering shear forces, when rendering normal forces its design assumes a *fingertip-as-a-point* model and relies only on width modulation. As such, only contact location can be immediately perceived, while orientation and curvature information must be integrated over time.

NormalTouch, conversely, treats the *fingertip-as-a-disc* [2]. In so doing, at any point in time, the finger can perceive contact location and orientation. By varying the orientation over time, the user can understand the curvature of a surface. In the same paper,

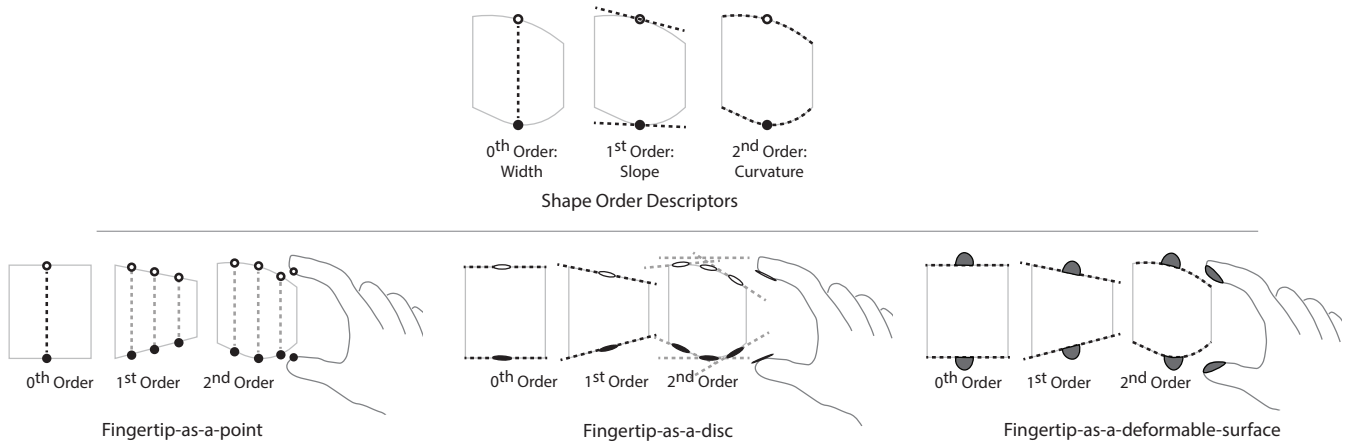


Figure 2: An illustration of the orders of shape descriptors and how they are experienced under different fingertip models. Top left: zeroth-, first-, and second-order shape descriptors convey width, slope, and curvature respectively. Top right: under the fingertip-as-a-point model, only zeroth-order information can be conveyed directly, with first- and second-order information needing multiple samples. Bottom left: considering the fingertip-as-a-disc model, both zeroth- and first-order information is conveyed directly, with only second-order requiring integration over time. Bottom right: in reality, the fingertip acts as a deformable-surface, where zeroth-, first-, and second-order information can be perceived immediately.

the authors also present TextureTouch, which treats the *fingertip-as-a-deformable-surface*. With this design, the user can experience contact location, orientation, and curvature instantaneously. More recent work, HUGO [17], also treats the *fingertip-as-a-deformable-surface* and incorporates the mechanisms of a parallel manipulator and a pin array simultaneously for high-fidelity curvature and wide-band texture rendering. Currently, expanding these approaches to multiple contact points for 3D object grasping remains impractical.

Different shape orders can be rendered or approximated regardless of the fingertip model selected as long as the user integrates surface information over time. As more complex fingertip models are selected, increasingly complex shapes can be conveyed instantaneously in a single contact. More complex models necessitate more complex mechanisms, creating a trade-off in practicality.

No existing handheld haptic controller for VR achieves full shape order asymmetric rendering for 3D shapes.

4 A CONTROLLER FOR HIGH-RESOLUTION HAPTIC SHAPE RENDERING

In this section, we build upon prior work like NormalTouch [2] to propose a dual Stewart platform mechanism to achieve high-resolution haptic 3D shape rendering. We adopt a *fingertip-as-a-disc* model to achieve zeroth- and first-degree shape rendering and approximate second-degree rendering.

4.1 Dual Stewart Platforms

Parallel manipulators are effective in rendering haptic feedback, from fingertip haptic devices [12, 34, 41] to grounded tabletop delta robots [1, 28]. The best-known parallel manipulator is the Stewart Platform—a 6-DOF mechanism that can support three translational and three rotational motions [8, 10].

Each degree of freedom requires its own linear motor. For shape rendering in 2.5D, not all degrees of freedom are essential. To render zeroth- and first-order shape descriptors, and approximate second-order descriptors, only 3 DOFs are required: height, pitch, and roll. For example, NormalTouch uses a single 3-DOF Stewart Platform for 2.5D surface rendering [2]. We propose dual, stacked Stewart Platforms to achieve complex shape rendering in 3D.

Designing for multi-finger interaction with haptic controllers for VR places inherent constraints on the prototype’s size, weight,

maximum force capabilities, mechanical intricacy, and operational method. For handheld controllers, one of these constraints relates to how we actuate the haptic feedback, choosing either an *outside-in* or *inside-out* approach.

4.2 “Outside-In” vs “Inside-Out” Controller Design

To attain reliable and rapid 3-DOF motion for each finger, it is imperative to integrate the motors, linkages, and joints directly into the body of the controller. Approaches in the literature can be classified as *outside-in* and *inside-out* according to the relative placement of the actuators and their end effectors.

In *outside-in* designs, actuators are on the *outside* of the actuation surface, typically serving to ‘pull’ the *inner* surface outwards. For handheld controllers, the actuators sit outside the hand. In CLAW [7], for example, the actuation mechanism sits on a hinge behind the back of the hand.

In *inside-out* designs, actuators are on the *inside* of the surface, typically ‘pushing’ *outwards*. In handheld controllers, then, these mechanisms are within the user’s grasp and result in constant contact with the user’s hand (e.g., [14, 24, 42]).

There is a trade-off between the two approaches. While compact, *inside-out* approaches typically have a limited width range, preventing users from grabbing very small objects—the plates cannot get close together because the mechanism is typically housed between them. The *outside-in* approach can solve this limitation, but the overall form factor is larger as the mechanisms sit outside the hand. As a first attempt to achieve 3D object rendering using a dual Stewart Platform design, we prioritise the range of motion and resolution over the form factor. Thus, we chose the *outside-in* paradigm for our implementation.

We present an instance of an *outside-in* haptic controller that can perform zeroth-, first-, and (approximated) second-order shape rendering. The mechanism we use is a 3-DOF 3-PRS (prismatic-revolute-spherical) parallel manipulator. This type of 3-DOF platform is convenient to analyse and control, and we present its working principle and the Inverse Kinematics model, which can be easily adapted to other subclasses of 3-DOF manipulators.

4.3 Stuet

Stuet is an example of a handheld haptic controller that can simulate the action of pinch-grasping complex objects. Like NormalTouch [2],

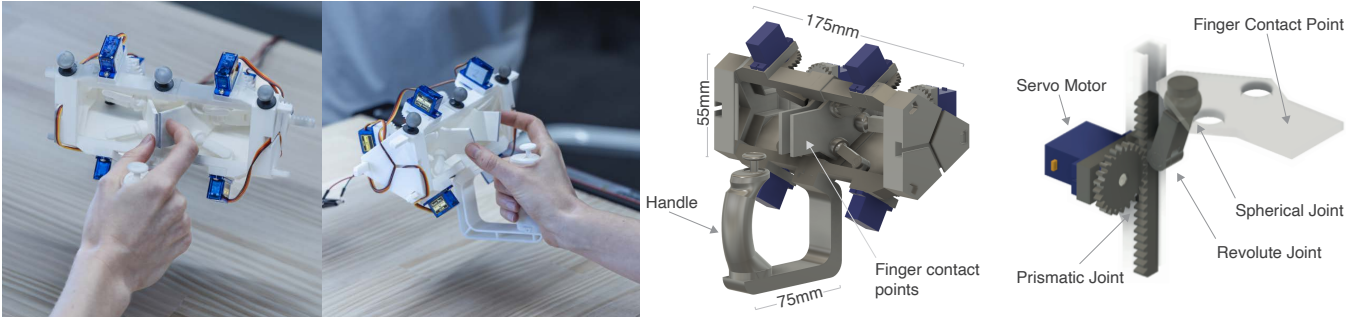


Figure 3: Left and Left-Mid: Images of Stuet in use. Thumb and finger platforms can be seen in various positions, including asymmetrical positions. Right-Mid and Right: The 3D model of Stuet from Fusion 360. Left: showing the scale of Stuet. Right: showing how each leg of each Stewart platform is formed from a prismatic joint, a revolute joint, and a spherical joint (PRS).

it produces normal forces at the user's fingertips. However, the forces are applied to two fingers instead of one (e.g., index and thumb) to provide a grasping sensation. We discuss the controller's design, including the novel nuances of its Stewart mechanism, its control and the inverse kinematics model required for its use.

Stuet consists of two inward-looking motion platforms and a handle. The platform mechanism we used is a 3-DOF 3-PRS parallel manipulator with legs consisting of two linkages interconnected with prismatic, revolute, and spherical joints to facilitate (a) motion in three degrees of freedom and (b) simplified, precise control, and (c) increased stability (as the initial prismatic joint is linear). This is the first use of this class of Stewart Platform in the HCI literature. Each platform can move up and down (width adjustment) and tilt in any direction (slope and curvature adjustments). The index finger and the thumb are placed on the bottom side of each platform to facilitate a pinching action (Figure 3). The handle is held by the remaining three fingers. This design allows Stuet to (asymmetrically) render all three shape descriptors for grasping 3D shapes which no other controller (as seen in Table 1) can perform.

All major structural components are made from 1.75 mm PLA material using a Flashforge Guider IIs FDM 3D printer. The 3D models were designed in Autodesk Fusion 360, and they can be found in our GitHub repository alongside the Arduino code (anonymised for review). An Adafruit HUZZAH32 microcontroller and a 3.7 V battery (1000 mAh) are mounted at the back of the haptic device. Communication is done via Bluetooth.

The legs are actuated using TowerPro SG90 hobby servo motors. A gear-rack mechanism converts the rotational motion of servos into the linear motion of prismatic joints. Revolute joints were made using Chicago screws, while spherical joints were fully 3D printed. A total of 6 servo motors are controlled using Adafruit PCA9685 servo driver, which is also mounted at the back of the haptic device. A cable is connected to the servo driver to power the servo motors independently of the ESP32. The controller weighs 265 g. Figure 3 illustrates Stuet's overall structure while Figure 4 (middle) shows the mechatronic diagram (in a partial configuration for performance analysis purposes).

4.3.1 Control And Operation Principle

The current implementation involves open-loop control only for simplicity purposes. The user's fingers are tracked with the VR headset's built-in hand tracking. When the user interacts with virtual objects, Stuet computes two normal vectors of the object's surfaces in contact with the user's hand. If the user is exploring virtual surfaces, the platforms continuously update their position relative to their movement. When sufficient force is applied, as measured by the current draw, the user can also pick virtual objects up.

Normal vector computation of virtual objects is used to infer the tilt and height of each platform. Information about tilt and height

is used to set pulse-width modulation (PWM) signals for the servo motors. To find the relationship between tilt/height information and PWMs, we provide the inverse kinematics model of the platforms.

4.4 Inverse Kinematics Model

The inverse kinematics (IK) of the 3-PRS parallel manipulator is a mapping from the final position of the platform to the inputs that would yield that position - an output-to-input mapping. While tuned for Stuet, this model could be used for any 3-DOF Stewart platform style control.

First, we assign a static coordinate frame, XYZ , to the base of the platform (considered to be fixed on the VR controller) and a moving coordinate frame, xyz , to the centre of the platform, as shown in Figure 4. The platform can move up/down and tilt in any direction but cannot move laterally or around its mid-axis. For simplicity, we can think of tilt as a pair of axis of rotation, \vec{v} , and angle of rotation, θ . Denote the z -coordinate of xyz frame with respect to XYZ frame as z_T . The controlled variables are the heights of each of the three legs, H_1 , H_2 , and H_3 . Thus, the inverse kinematics for this platform is a mapping between (z_T, \vec{v}, θ) triplet and (H_1, H_2, H_3) triplet.

The homogeneous transformation matrix, $[T]$, can be used to describe the xyz frame in terms of XYZ frame:

$$[T] = \begin{bmatrix} \mathbf{R} & \vec{v}_T \\ \mathbf{0} & 1 \end{bmatrix} \quad (1)$$

where \mathbf{R} and $\vec{v}_T = [x_T \ y_T \ z_T]^T$ are rotation matrix and position column vector of xyz with respect to XYZ , and $\mathbf{0}$ is a zero row vector.

\mathbf{R} is fully determined using \vec{v} and θ . The x_T and y_T are given as:

$$x_T = \frac{r}{2}(\cos\beta\cos\gamma + \sin\alpha\sin\beta\sin\gamma - \cos\alpha\cos\gamma) - h\sin\beta + \frac{R}{2} \quad (2)$$

$$y_T = h\sin\alpha\cos\beta - r\sin\alpha\sin\beta\cos\gamma = r\cos\alpha\sin\gamma \quad (3)$$

where α, β, γ are respectively roll, pitch, and yaw computed from rotation matrix decomposition $\mathbf{R} = X(\alpha)Y(\beta)Z(\gamma)$ using Tait-Bryan convention; r and R are radii of circles passing through spherical and revolute joints respectively; h - z -distance from platform's centre to spherical joints.

Position vectors of spherical joints in terms of XYZ can be found using $[T]$ applied to their respective constant position vectors in the xyz frame. Positions of the revolute joints in the x and y planes are always constant and can be easily computed, whereas their z -coordinates are the output of the IK model. The inverse kinematics for each leg is given by:

$$H_i = R_{iz}S_{iz} - \sqrt{l^2 - (R_{ix} - S_{ix})^2 - (R_{iy} - S_{iy})^2}, \quad i = 1, 2, 3 \quad (4)$$

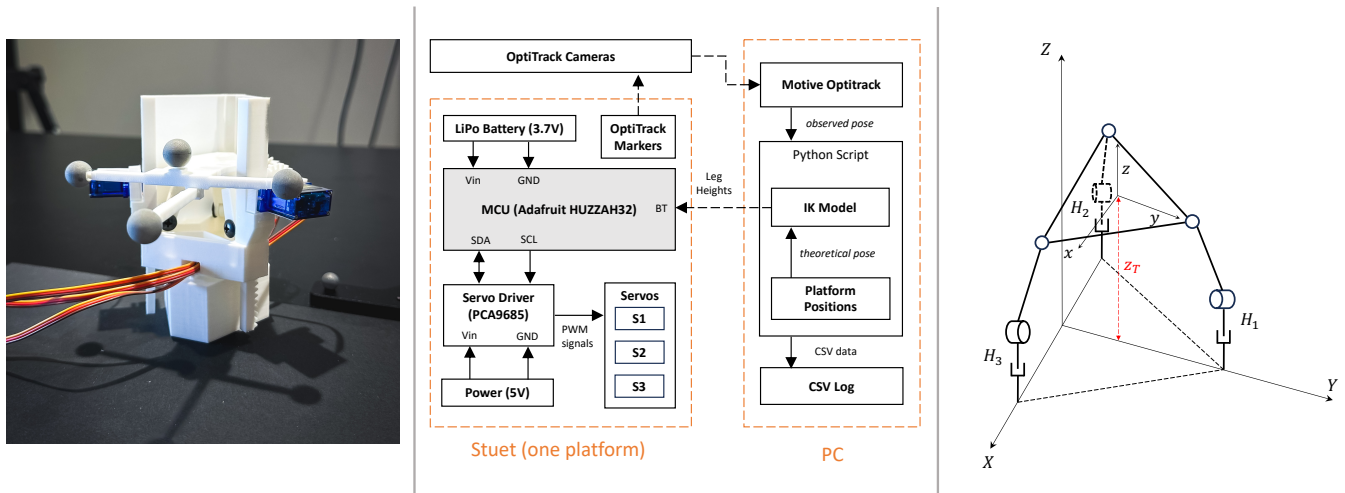


Figure 4: Left: Setup for the platform performance analysis using Optitrack markers. Middle: logic for the performance analysis including Stuet's partial component diagram. Right: Schematic diagram of the 3-PRS platform for the IK model derivation.

where S_i and R_i are position vectors of spherical and revolute joints with respect to XYZ-frame, l is the length of linkages between these joints. The full derivation of the IK model can be found on our GitHub page (anonymised for review).

5 TECHNICAL EVALUATION

We conducted a technical evaluation to assess and measure the performance of our device. After assessing the configuration space of the device, we used a vision-based motion-tracking system as ground truth to measure the accuracy of the device. We also report the results of this evaluation in terms of accuracy, repeatability, and speed.

5.1 The C-Space

First, we need to understand the configuration space (C-space) of our platforms, that is, the complete space of possible positions of the platforms, given their constraints. Apart from each joint's inherent DOF constraint, the C-space also depends on the way we designed the 3D models of each joint and linkage, but the findings should hold for a general case. Using the inverse kinematics model, we have investigated the C-space to identify potential input signals (H_1, H_2, H_3). However, for the sake of simplicity, comprehensibility and usability, we have chosen to present the data in terms of output signals (z_T, \vec{v}, θ).

Revolute joints have no range of motion (ROM) constraint and can move 180° . Spherical joints were fully 3D printed and required us to limit their maximum ROM to 35° from their default position. Each leg has approximately 43 mm ROM.

5.2 Calibration and Testing Methods

Servo motors were calibrated in a 12-bit PWM range based on Adafruit PCA9685 library¹. Using PWM frequency of 50 Hz, 4.8 V DC, the expected PWM values should be between 102/4096 and 409/4096 as per the TowerPro SG90 datasheet. The behaviour of the servo motors was observed to be mostly linear. This allowed us to make a simple mapping between the platform's leg positions and servo PWM signals. In the servos we tested, for an increment of 5 PWM units, the mean angle increment was 2.226° , and the mean squared error was 0.707° .

Following calibrations, we conducted a performance analysis of the actual behaviour of the platforms. Each 3-PRS platform

was tested independently using the Motive Optitrack system, the setup for which is shown in Figure 4. All of the tests involved the following: select a theoretical platform position (*theoretical pose*), feed it into the IK model to obtain the platform leg heights, select the corresponding PWM signals for each servo motor, and read off the actual platform position (*observed pose*) using the Motive tracking system. The difference between these two poses serves as the error metric. The whole system test logic is shown on the block diagram in Figure 4.

5.3 Accuracy and Repeatability

The first set of tests involved evaluating the position accuracy. For this, we discretised the whole C-space, and following our test method described above, we collected the (*height, axis, angle*) readings using Optitrack cameras. Finally, we compared them against the theoretical values.

Across 279 platform positions, the performance was observed to be highly accurate, with an average absolute height error of 0.54 mm ($SD = 0.31$ mm). Tilt axis and angle errors were only evaluated based on 200 positions with non-zero tilt angles. For these runs, the average absolute error in tilt angle was 0.81° ($SD = 0.74^\circ$). Tilt axis error was quantified using average cosine similarity ($M = 0.99$, $SD = 0.01$).

In the second round, we selected 20 reachable platform positions at random and ran the experiment five times for each position (100 data points in total). This was done to test how consistent the platform's motion behaviour is given the same input - its repeatability.

For each position, we computed the standard deviation of the quantity of interest and took the average. Average standard deviations for the platform's height and tilt angles were found to be 0.048 mm and 0.0833° respectively. Plots in Figure 5 show the regression lines between the reference theoretical and observed poses, both for the platform's height and tilt angles.

To quantify the variation in measurements of the tilt axis, we computed the 3x3 covariance matrix of the tilt vector at each of the 20 reachable positions. We then calculated the trace of each of the covariance matrices, and the average covariance matrix trace was equal to $6.3384e-5$, implying very low variation in tilt axis vectors. Overall, the above data suggest highly consistent performance across different runs.

¹<https://github.com/adafruit/Adafruit-PWM-Servo-Driver-Library>

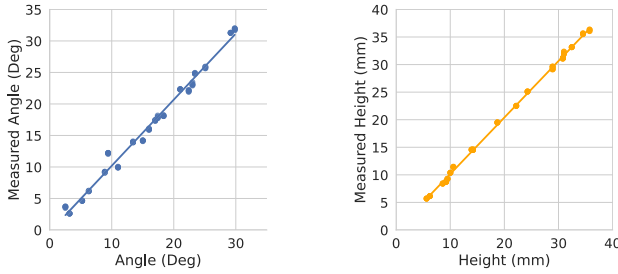


Figure 5: Regression plots between the theoretical and observed pose data during the *Repeatability test*. Left: Tilt angle component of the pose. Right: Platform height component of the pose.

5.4 Speed

We quantified speed performance using two metrics: the time it takes for the platform to start moving after the movement command is sent to the microcontroller (*signal latency*) and the time it takes for a platform to finish its movement from the time it starts moving (*platform motion speed*). Speed tests were conducted using Optitrack cameras with a refresh rate of 120 Hz (8.33 ms). For all tests, the signal latency spanned between 42 ms and 85 ms ($M = 59$ ms, $SD = 16$ ms).

We measured the platform tilt speed as it rotates from the maximum tilt to one side (-28°) to the maximum tilt on the opposite (30°) around the x-axis when the platform is at middle elevation. Across multiple runs, the average motion time was 136 ms ($SD = 23$ ms).

We tested the parallel motion speed of the platform from position $(H_1, H_2, H_3) = (0, 0, 0)$ to (x, x, x) , where $H_i, i = \{1, 2, 3\}$ is the prismatic joint elevation from its lowest position. For $x = \{10, 20, 40\}$ mm, the motion times were 101 ms ($SD = 5$ ms), 204 ms ($SD = 8$ ms), and 389 ms ($SD = 15$ ms) respectively. SG90 servos are rated to $0.1s/60^\circ$ which would translate to 73 ms, 147 ms, and 293 ms for above transitions. Such discrepancy may be a result of the tight gear fit of the 3D-printed prismatic joint.

Movement of each leg individually from $H_i = 0$ mm to $H_i = 10$ mm was 78 ms ($SD = 13$ ms) for all legs on average, which is in close agreement with the rated servo speed.

6 FEASIBILITY DEMONSTRATION

We have proposed and prototyped a design for higher-resolution haptic controllers for VR. To better understand the experience of these controllers and to determine the difference in the experience over lower-order controllers, we conduct a feasibility demonstration. This is intended to be formative and guide the ongoing design of controllers in this space.

Our evaluation emphasised object exploration (tracing), where the greatest range of finger positions could be experienced. That said, grabbing and releasing were inherently included, as participants would move towards / away from objects, thus grabbing and releasing. We chose not to include moving interactions, as these have no further impact on shape resolution (the shape is determined at grab-time), and so would not have impacted our insights.

6.1 Demo Protocol

We developed two demo scenarios using Stuet. In the first, participants could freely explore a vase (see Figure 6). We implemented this demo with two versions of haptic feedback (two conditions). In the first, Stuet provided dynamic zeroth-order feedback only. That is, the platforms stayed perpendicular to the fingertips but dynamically changed their width to correspond to the vase. This is akin to the shape feedback from Grability [5], or Spinocchio [22], for example, and serves to treat the *fingertip-as-a-point*. In the second condition,

Stuet provided full zeroth-, first- and (approximated) second-order feedback, dynamically changing its width and slope configurations as participants moved around the vase. This conceptualised the *fingertip-as-a-disc*. Participants could trace the outer contour of the vase with their two fingers. For the second demo, we created a range of virtual objects for the participants to explore, including spheres, cubes, and prisms.

We recruited 8 participants to experience object exploration with Stuet (5 women, 3 men, ages 21 to 32, $M=26.25$, $SD=3.73$). Participants reported having little to no prior VR or haptic feedback experience. All participants were right-handed and used their dominant hand for the demo. Each participant was asked to complete both demo scenarios. After the first demo, we asked users: “Which haptic rendering condition do you prefer and why?” In both scenes, participants were encouraged to also verbalise their opinion about the haptic experience. Our feasibility demonstration took place in a quiet university laboratory, and lasted approximately 15 minutes. It was approved by the local ethics committee.

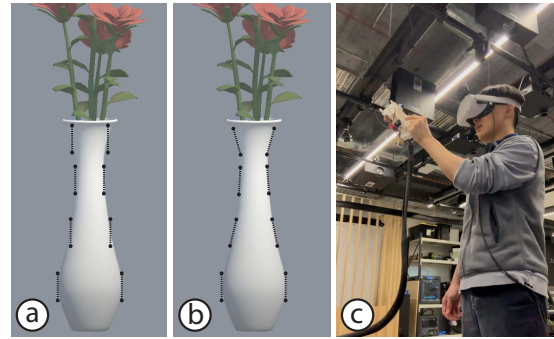


Figure 6: a) Stuet conveying the shape of the vase using zeroth-order information, as in one condition of the evaluation. b) Stuet conveying the shape of the vase through both zeroth- and first-order information, per condition two. c) A participant completing the feedback task, in which they explored different object shapes and vocalised their experience.

6.2 Feedback from Users

All participants explored the objects within the two demo applications. Participants consistently described the experience of the higher-shape-resolution Stuet as ‘smoother’ and ‘more realistic’ than the zeroth-order configuration. For example, P1 said, “I feel like this one [with all orders] is slightly smoother than the previous one [with zeroth order only]”. Anecdotally, we believe the ‘smooth’ sensation comes from the platforms not pushing perpendicular to the finger during dynamic exploration but rather rolling around the finger (by changing both width and slope simultaneously) during reconfiguration. Participants also used words like ‘more intense’ (P2), ‘more representative’ (P6), and ‘more fluent’ (P4) to articulate how the high-resolution configuration differed from the lower-resolution one. Conversely, the zeroth-order configuration was described by one participant (P5) as: “I felt like the controller was forcing me to do that action, it was pushing me, which was kind of stiff”. Interestingly, only P3 explicitly noticed that the platforms were also changing their orientation:

“In the first case, I guess, you know, there’s only one dimension of these blades. Yeah. And if you switch to the second one, you can see another dimension. Which I think is [a] pretty, pretty interesting change. And I like it. It gave you a different feeling of grasping something, [and] that’s a very new experience in the VR world” (P3).

We noticed the participants' fingers stayed in constant contact with the platforms. We assumed participants would relax their fingers around the handle and only reach for the platforms during object interactions. This is not what played out during the evaluation. In turn, one participant (P5) even suggested, "*it would be good to attach fingers to the [platforms] with straps so that fingers stay still*", making it easier to rest as the platforms reconfigure during use. While attempting to stay in contact with the platforms, across both conditions, participants (P1 and P5) mentioned that their experience felt better when the platforms were expanding, compared to when they were collapsing: "*I feel okay grabbing it when it pushes out, but when it's going inwards, my fingers are hanging*" (P1).

The configuration and design of Stuet may not suit all hand sizes. One participant, in particular, commented that the handle was too large, and a couple of participants described how reaching for the platforms in certain configurations was awkward. This was especially true for one of our prism objects, where the platforms formed a triangle pointing away from the user, requiring the user to reach around the edge of the platforms to lie their fingers along the surface: "*...I'm trying to get to it, it I can't really feel it that well*" (P1).

Finally, some participants reported the sensation of '*vibration*' during interaction with the vase in both rendering instances. For example, P7 said: "It feels a bit shaky due to motor vibrations", whereas P2 said that "in both of them, [I] could feel the vibrations in [my] hand". The source of vibration likely arises from two factors: (1) the use of cheap hobby servo motors with an open-loop control algorithm, and (2) cumulative error in 3D printing tolerances (Stuet includes 18 joints and 16 linkages). These problems can be mitigated, e.g., through more advanced actuators and lower-tolerance manufacturing processes, though this may likely add weight to the controller.

7 DISCUSSION

Stuet has a high range of motion and is, thus, able to render complex 3D shapes for grasping and exploration. Specifically, the plates can render widths between 2 mm and 75 mm, and each plate can be adjusted to angles of up to 30°. By choosing to conceptualise the *fingertip-as-a-disc*, Stuet can instantaneously convey zeroth- and first-order shape information (for width and surface orientation). The controller can also approximate second-order shape features (curvature) over time and, importantly, render asymmetrical objects. To the best of our knowledge, no controller to date has been able to convey this complexity of shape information for 3D objects. Through our feasibility demonstration, we found that there are experiential benefits to this level of complexity: participants reported greater smoothness and realism in comparison to the lower-order, *fingertip-as-a-point*-style controllers seen previously.

We prioritised high-resolution shape rendering in the design of Stuet. To this end, we built a large, multi-servo, (relatively) power-intensive, wired controller for two-finger grasping. There are circumstances in which the trade-offs of such a controller may be acceptable (e.g., expert operation in technical contexts). However, it is hard to imagine it being widely adopted. Commercial handheld controllers today are small, robust, efficient, and ergonomic. Other controllers have attempted to pursue these design requirements (e.g., [7, 14]) and have had to trade away resolution. Further engineering work could reduce the size and weight of our device, and make it wireless. However, we believe that it is beneficial in research to also prioritise resolution and relax the other design constraints to get a complete picture of what can be achieved and, in turn, motivate the design of compact, robust, higher-resolution controllers.

We presented the concept of dual Stewart platforms for high-resolution controllers. We discussed the two possible design directions for implementing such controllers: *outside-in* (with actuation outside of the hand) vs *inside-out* (with actuation within the grasp of the hand). Here, we chose the *outside-in* approach that allows the

plates to come together, representing very thin objects (e.g., grasping a leaflet). The *inside-out* approach, conversely, would not facilitate this but would have the potential to facilitate a more robust, compact design. We believe pursuing robustness requires controller designs where actuators and actuation mechanisms are housed within the controller body, such that they cannot be accidentally collided with. This may be achieved in an *inside-out* approach by minimising the actuation mechanisms, but would likely result in a reduced range of motion. Minimisation opportunities include: (1) using passive springs coupled with brake-based mechanisms [6, 39]; (2) using kinematically equivalent origami linkages and foldable hinges that can approximate spherical and revolute joints [3, 4]. For example, an equivalent origami 3-DOF Stewart Platform can be implemented from fibreglass and Kapton tape to lower the overall weight and size of the controller prototype [32].

Designing multi-functional, ergonomic controllers for grasping is complicated due to the range of rotation of the thumb. For joystick interaction, the thumb is relatively upright, whereas for grasping, it rotates almost 90° to the side. During this rotation, the gap between the base of the thumb and the base of the index finger shrinks dramatically, reducing the size of the controller that can comfortably be held there. Stuet is currently designed for use by either hand and so does not take this into account.

We developed Stuet using commodity fabrication techniques and components. In particular, we used lightweight, low-cost, hobby servo motors. These motors need individual calibration, though we found their performance to be linear once we determined their start and stop positions. The controller performs remarkably accurately, and any potential inaccuracies may also stem from the calibration of the Optitrack system used in our evaluation. Additionally, the platform moves quickly, requiring approximately 400 ms to move between the extremities of its configuration. While very fast, for true, seamless haptic exploration, a predictive algorithm could be added to improve its interactivity. Higher-performance servos could also be used, and while this may reduce some of the vibration described in our initial user feedback, but this would likely add significant weight to the controller.

Stuet attempts to facilitate *natural* object interactions - enabling the exploration of virtual objects. In order to enable this, each of the controller's contact plates can achieve 30° of tilt in any direction. In reality, however, many of these angles would be outside of any fingers' cone of friction [40] and would result in the object slipping if lifted in this way. Stuet does not seek to convey weight, and, as such, this is out of scope in this work. If aiming to achieve 'natural' interaction and render weight, future iterations of the controller must consider the fingers' cone of friction.

8 CONCLUSION

We presented the design and implementation of a novel handheld haptic controller - Stuet - that can render complex asymmetrical 3D shapes for grasping and exploration. Our controller consists of two twin 3-DOF Stewart platforms that move independently, each actuated by three servo motors. This allows Stuet to render three shape descriptors: zeroth order as related to the graspable object's width, first order related to the object's shape angles, and approximate second-order shape descriptor information, which provides information regarding the curvature of the object's surface. Furthermore, due to the actuation independence, Stuet can also render asymmetrical objects, allowing the user to grasp arbitrary object shapes between 2 to 75 mm in width and up to 30° tilt angle in any direction with respect to the vertical plane. Stuet belongs to a class of twin Stewart platform handheld controllers. As such, we proposed the concept of dual Stewart platforms as a new paradigm for handheld VR devices that can help achieve high-resolution shape rendering. Our proof of concept controller was shown to be fast, accurate, and achieve a high range of motion. To support

our claims, we conducted a feasibility demonstration to evaluate how Stuet can be employed in various VR scenarios. Based on our evaluation, we discussed the new capabilities dual Stewart platform controllers can offer for haptic interactions in VR.

ACKNOWLEDGMENTS

The authors wish to thank A, B, and C. This work was supported in part by a grant from XYZ (# 12345-67890).

REFERENCES

- [1] J. Arata, H. Kondo, M. Sakaguchi, and H. Fujimoto. Development of a haptic device “delta-4” using parallel link mechanism. In *2009 IEEE International Conference on Robotics and Automation*, pp. 294–300. IEEE, 2009. 4
- [2] H. Benko, C. Holz, M. Sinclair, and E. Ofek. Normaltouch and turetouch: 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, 2016. 2, 3, 4
- [3] L. A. Bowen, C. L. Grames, S. P. Magleby, L. L. Howell, and R. J. Lang. A classification of action origami as systems of spherical mechanisms. *Journal of Mechanical Design*, 135(11), 2013. 8
- [4] J. Cai, Z. Qian, C. Jiang, J. Feng, and Y. Xu. Mobility and kinematic analysis of foldable plate structures based on rigid origami. *Journal of Mechanisms and Robotics*, 8(6):064502, 2016. 8
- [5] I. Choi, H. Culbertson, M. R. Miller, A. Olwal, and S. Follmer. Gravity: 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, 2017. 3, 7
- [6] I. Choi, E. W. Hawkes, D. L. Christensen, C. J. Ploch, and S. Follmer. Wolverine: A wearable haptic interface for grasping in virtual reality. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 986–993. IEEE, 2016. 3, 8
- [7] I. Choi, E. Ofek, H. Benko, M. Sinclair, and C. Holz. 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*, pp. 1–13, 2018. 1, 2, 3, 4, 8
- [8] B. Dasgupta and T. Mruthyunjaya. The stewart platform manipulator: a review. *Mechanism and machine theory*, 35(1):15–40, 2000. 4
- [9] T. Endo, H. Kawasaki, T. Mouri, Y. Ishigure, H. Shimomura, M. Matsumura, and K. Koketsu. Five-fingered haptic interface robot: Hiro iii. *IEEE Transactions on Haptics*, 4(1):14–27, 2010. 1
- [10] E. F. Fichter. A stewart platform-based manipulator: general theory and practical construction. *The international journal of robotics research*, 5(2):157–182, 1986. 4
- [11] S. Follmer, D. Leithinger, A. Olwal, A. Hogge, and H. Ishii. inform: dynamic physical affordances and constraints through shape and object actuation. In *Uist*, vol. 13, pp. 2501–988. Citeseer, 2013. 3
- [12] M. Gabardi, M. Solazzi, D. Leonardis, and A. Frisoli. A new wearable fingertip haptic interface for the rendering of virtual shapes and surface features. In *2016 IEEE Haptics Symposium (HAPTICS)*, pp. 140–146. IEEE, 2016. 2, 4
- [13] A. García Álvarez, A. Roby-Brami, J. Robertson, and N. Roche. Functional classification of grasp strategies used by hemiplegic patients. *PLoS One*, 12(11):e0187608, 2017. 1
- [14] E. J. Gonzalez, E. Ofek, M. Gonzalez-Franco, and M. Sinclair. X-rings: A hand-mounted 360° shape display for grasping in virtual reality. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 732–742, 2021. 1, 2, 3, 4, 8
- [15] X. Gu, Y. Zhang, W. Sun, Y. Bian, D. Zhou, and P. O. Kristensson. Dexmo: An inexpensive and lightweight mechanical exoskeleton for motion capture and force feedback in vr. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 1991–1995, 2016. 1
- [16] S. Heo, C. Chung, G. Lee, and D. Wigdor. 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*, pp. 1–11, 2018. 1, 2
- [17] Y. Herbst, A. Wolf, and L. Zelnik-Manor. HUGO, a high-resolution tactile emulator for complex surfaces. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2023. 2, 4
- [18] R. Hinchet, V. Vechev, H. Shea, and O. Hilliges. DextrES: Wearable haptic feedback for grasping in vr via a thin form-factor electrostatic brake. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pp. 901–912, 2018. 1, 2
- [19] M. Hoppe, P. Knierim, T. Kosch, M. Funk, L. Futami, S. Schneeggass, N. Henze, A. Schmidt, and T. Machulla. VRHapticDrones: Providing haptics in virtual reality through quadcopters. In *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia*, pp. 7–18, 2018. 2
- [20] H. In, B. B. Kang, M. Sin, and K.-J. Cho. Exo-glove: A wearable robot for the hand with a soft tendon routing system. *IEEE Robotics & Automation Magazine*, 22(1):97–105, 2015. 2
- [21] J. Iqbal, N. Tsagarakis, and D. Caldwell. Four-fingered lightweight exoskeleton robotic device accommodating different hand sizes. *Electronics Letters*, 51(12):888–890, 2015. 2
- [22] M. J. Kim, N. Ryu, W. Chang, M. Pahud, M. Sinclair, and A. Bianchi. Spinocchio: Understanding haptic-visual congruency of skin-slip in vr with a dynamic grip controller. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2022. 1, 2, 3, 7
- [23] S.-C. Kim, C.-H. Kim, G.-H. Yang, T.-H. Yang, B.-K. Han, S.-C. Kang, and D.-S. Kwon. Small and lightweight tactile display (salt) and its application. In *World Haptics 2009-Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 69–74. IEEE, 2009. 2
- [24] J. Lee, M. Sinclair, M. Gonzalez-Franco, E. Ofek, and C. Holz. Torc: A virtual reality controller for in-hand high-dexterity finger interaction. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–13, 2019. 2, 4
- [25] D. Leonardis, M. Solazzi, I. Bortone, and A. Frisoli. A wearable fingertip haptic device with 3 dof asymmetric 3-rsr kinematics. In *2015 IEEE world haptics conference (WHC)*, pp. 388–393. IEEE, 2015. 2
- [26] D. Leonardis, M. Solazzi, I. Bortone, and A. Frisoli. A 3-rsr haptic wearable device for rendering fingertip contact forces. *IEEE transactions on haptics*, 10(3):305–316, 2016. 2
- [27] J.-Y. Lo, D.-Y. Huang, C.-K. Sun, C.-E. Hou, and B.-Y. Chen. Rollingstone: Using single slip taxel for enhancing active finger exploration with a virtual reality controller. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pp. 839–851, 2018. 2
- [28] S. Martin and N. Hillier. Characterisation of the novint falcon haptic device for application as a robot manipulator. In *Australasian Conference on Robotics and Automation (ACRA)*, pp. 291–292. Citeseer, 2009. 4
- [29] T. H. Massie, J. K. Salisbury, et al. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, vol. 55, pp. 295–300. Chicago, IL, 1994. 1
- [30] V. R. Mercado, M. Marchal, and A. Lécuyer. “haptics on-demand”: A survey on encountered-type haptic displays. *IEEE Transactions on Haptics*, 14(3):449–464, 2021. 2
- [31] K. Minamizawa, S. Fukamachi, H. Kajimoto, N. Kawakami, and S. Tachi. Gravity grabber: wearable haptic display to present virtual mass sensation. In *ACM SIGGRAPH 2007 emerging technologies*, pp. 8–es. 2007. 2
- [32] S. Mintchev, M. Salerno, A. Cherpillod, S. Scaduto, and J. Paik. A portable three-degrees-of-freedom force feedback origami robot for human–robot interactions. *Nature Machine Intelligence*, 1(12):584–593, 2019. 8
- [33] C. Pacchierotti, G. Salvietti, I. Hussain, L. Meli, and D. Prattichizzo. The hRing: A wearable haptic device to avoid occlusions in hand tracking. In *2016 IEEE Haptics Symposium (HAPTICS)*, pp. 134–139. IEEE, 2016. 2
- [34] D. Prattichizzo, F. Chinello, C. Pacchierotti, and M. Malvezzi. Towards wearability in fingertip haptics: a 3-dof wearable device for cutaneous

- force feedback. *IEEE Transactions on Haptics*, 6(4):506–516, 2013. 1, 2, 4
- [35] I. Sarakoglou, N. Garcia-Hernandez, N. G. Tsagarakis, and D. G. Caldwell. A high performance tactile feedback display and its integration in teleoperation. *IEEE Transactions on Haptics*, 5(3):252–263, 2012. 2
- [36] I. Sarakoglou, N. G. Tsagarakis, and D. G. Caldwell. A compact tactile display suitable for integration in vr and teleoperation. In *2012 IEEE International Conference on Robotics and Automation*, pp. 1018–1024. IEEE, 2012. 2
- [37] M. Sato. Development of string-based force display: SPIDAR. In *8th international conference on virtual systems and multimedia*, pp. 1034–1039. Citeseer, 2002. 1
- [38] J. Shigeyama, T. Hashimoto, S. Yoshida, T. Narumi, T. Tanikawa, and M. Hirose. Transcalibur: A weight shifting virtual reality controller for 2d shape rendering based on computational perception model. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–11, 2019. 2
- [39] M. Sinclair, E. Ofek, M. Gonzalez-Franco, and C. Holz. Capstan-crunch: A haptic vr controller with user-supplied force feedback. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*, pp. 815–829, 2019. 2, 8
- [40] J. B. Smeets, K. van der Kooij, and E. Brenner. A review of grasping as the movements of digits in space. *Journal of neurophysiology*, 122(4):1578–1597, 2019. 1, 8
- [41] M. Solazzi, A. Frisoli, and M. Bergamasco. Design of a novel finger haptic interface for contact and orientation display. In *2010 IEEE Haptics Symposium*, pp. 129–132. IEEE, 2010. 2, 4
- [42] Y. Sun, S. Yoshida, T. Narumi, and M. Hirose. Pacapa: A handheld vr device for rendering size, shape, and stiffness of virtual objects in tool-based interactions. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–12, 2019. 2, 4
- [43] W. M. B. Tiest. Tactual perception of material properties. *Vision research*, 50(24):2775–2782, 2010. 3
- [44] D. Tsetserukou, S. Hosokawa, and K. Terashima. LinkTouch: A wearable haptic device with five-bar linkage mechanism for presentation of two-dof force feedback at the fingerpad. In *2014 IEEE Haptics Symposium (HAPTICS)*, pp. 307–312. IEEE, 2014. 2
- [45] E. Whitmire, H. Benko, C. Holz, E. Ofek, and M. Sinclair. 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*, pp. 1–12, 2018. 2
- [46] M. W. Wijntjes, A. Sato, V. Hayward, and A. M. Kappers. Local surface orientation dominates haptic curvature discrimination. *IEEE transactions on haptics*, 2(2):94–102, 2009. 3
- [47] D. Yu, W. Jiang, A. Irlitti, T. Dingler, E. Velloso, J. Goncalves, and V. Kostakos. Haptics in vr using origami-augmented drones. In *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 905–906, 2022. doi: 10.1109/ISMAR-Adjunct57072.2022.00198 2
- [48] A. Zenner and A. Krüger. 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, 2017. 2
- [49] L. Zhu, X. Jiang, J. Shen, H. Zhang, Y. Mo, and A. Song. TapeTouch: A handheld shape-changing device for haptic display of soft objects. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3928–3938, 2022. 2, 3