Shape Aware Haptic Retargeting for Accurate Hand Interactions

Brandon J. Matthews * Bruce H. Thomas † G. Stewart Von Itzstein ‡ Ross T. Smith §
Australian Research Centre for Interactive and Virtual Environments
Wearable Computer Lab, University of South Australia

2022 IEEE on Conference Virtual Reality and 3D User Interfaces (VR) | 978-1-6654-9617-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/VR51125.2022.00083

Abstract

This paper presents Shape Aware Haptic Retargeting, an extension of "state-of-the-art" haptic retargeting that is the first to support retargeted interaction between any part of the user's hand and any part of the target object. In previous haptic retargeting algorithms, the maximum retargeting is applied only when the hand position aligns with the target position. Shape Aware Haptic Retargeting generalizes the distance computation process to instead consider the hand and target geometry. The shortest hand-target distance is then used to calculate the applied retargeting offset. This ensures the full amount of haptic retargeting is applied at the point of contact with the passive haptic regardless of contact position on the hand or target. We leverage existing geometry algorithms to implement three distance computation methods: Multi-Point, Primitive and Mesh Geometry, in addition to conventional single position approaches. These are evaluated through a set of simulated interactions instead of the single position representation used in previous haptic retargeting systems. The evaluation demonstrated all three approaches can provide improved interaction accuracy over a Point distance computation method, with Mesh Geometry being the most accurate and Primitive being the preferred method for combined performance and interaction accuracy.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality; Computer graphics—Shape modeling—Shape analysis

1 Introduction

This paper presents the first Shape Aware Haptic Retargeting (SAHR) algorithm that models the geometric mesh of both the physical hand and passive haptic target, to provide improved accuracy over current single point approaches. This enables accurate interactions between any part of the user's hand and the target during retargeted interactions.

Introducing physical objects to provide passive haptics for a matching Virtual Reality (VR) experience has been shown to improve the sense of presence and immersion in Virtual Reality systems [24]. However, they are often fixed in place or simple objects that cannot dynamically change their physical form unlike the Virtual Environment (VE). Haptic retargeting enables the reuse of passive haptics for dynamic virtual objects within the VE. This is achieved through redirection of the user's real hand in which an undetectable angular and/or translational gain are introduced to a virtual replica of their hand as it approaches the target. The user subconsciously corrects for the offset, resulting in them touching a physical passive haptic object with their real hand while their virtual hand touches a virtual object in a different spatial position.

During VR interactions using haptic retargeting, the spatial relationship between the virtual target and hand should be identical to the physical target and hand, throughout all interactions. Herein this is referred to as the Retargeted Interaction Alignment (RIA), where a "correct" RIA indicates an identical spatial relationship.

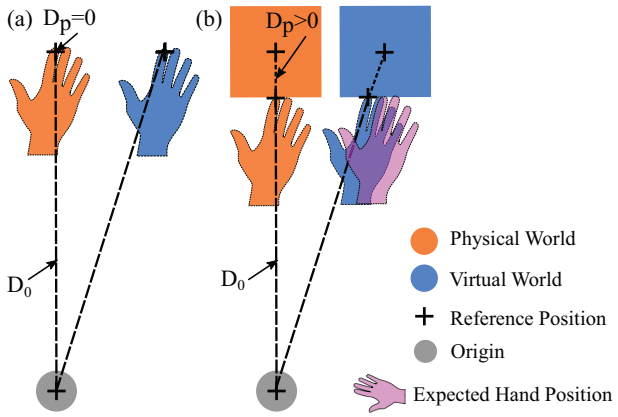


Figure 1: The relative virtual hand and target position matches the physical relationship if the reference points align (a). If they do not align, such as when the hand contacts the surface of the target, the relative virtual hand and target does not match the physical relationship (b).

In previous state-of-the-art haptic retargeting, the algorithm computes the applied offset using the distance between the hand and target as shown in Figure 1. The hand is represented by a single position, often on the fingertip [13, 19, 32, 46, 50] or palm [4, 14, 49]. In many cases, the target is also represented by a single position [4, 32, 49], while others incorporate some understanding of the target geometry [13, 50]. As such the RIA is not correct until the representative hand position and representative target position are perfectly aligned [30]. This can result in a discrepancy between the contact point of the physical and virtual hands with their respective targets depending on their size, shape, and the level of translational and rotational gain.

An assumed contact point, such as the index finger with a button, or the palm grasping a cube, may provide sufficient accuracy and minimal error in some situations. However, interactions in the real world often make use of the entire surface of the hand. For example operation of an interface usually incorporates multiple fingers or users may prefer to press a button with their palm, or another finger. A single point representing the hand limits accurate interactions to only that part of the hand. Furthermore, representing the target as a single position limits the user's flexibility by forcing the user to grab or touch targets in the required position.

An incorrect RIA can also result in visual interpenetration where the user's hand passes through a virtual object, which has been shown to break the virtual world illusion [10]. The likelihood of overshooting or missing the target, particularly during rapid interactions, is a known issue in outlined in previous research [4, 30, 32]. From a user's perspective this discrepancy can impact the usability of remapped physical-virtual interfaces [32] and other applied haptic retargeting systems. There is a need to further improve the performance and the functionality of existing haptic retargeting techniques

*e-mail: brandon.matthews@mymail.unisa.edu.au
†e-mail: bruce.thomas@unisa.edu.au
‡e-mail: gsa@vonitzstein.com
§e-mail: ross.smith@unisa.edu.au

978-1-6654-9617-9/22/$31.00 ©2022 IEEE
DOI 10.1109/VR51125.2022.00083

and the introduction of a geometric understanding of the user's hand is the natural next step in this direction. Since the advent of Haptic Retargeting in 2016 [4], VR devices such as the Oculus Quest 2 have developed to provide integrated hand pose tracking [38]. This provides an understanding of the user's hand pose and an approximate understanding of user's hand size and shape. These hardware advances have made it possible to reconsider the haptic retargeting algorithm by leveraging the ever-increasing precision of hand pose and geometry made available by commercial technologies.

The SAHR presented in this paper is a generalized, elegant extension upon existing haptic retargeting algorithms to provide correct RIA for interaction between any part of a user's hand and their interaction target. In SAHR, the distance function used in state-of-the-art haptic retargeting is replaced with a generalized method that considers the shortest distance between the physical shape of the hand and the target. This provides an understanding of the hand-target contact point and ensures correct RIA upon contact. Three shape approximations are presented and explored for both the hands and targets: Multi-Point, Primitive and Mesh Geometry.

The primary contributions of this paper include:

- Shape Aware Haptic Retargeting, an extension of haptic retargeting supporting natural interactions between any part of a user's hand geometry and any part of the interaction target.
- Implementation and evaluation of three alternative hand and target approximations for Shape Aware Haptic Retargeting.

This paper begins with a summary of related work in haptic retargeting implementations and collision/distance calculation in computer graphics. Next, the generalized distance method for haptic retargeting is outlined followed by the implementation of the hand and target approximations. Then the simulated interaction evaluations are explained along with their results. Finally, the results and contributions are discussed, and the paper concludes with limitations and future work.

2 Related Work

Three areas of prior related works from which this paper draws knowledge from are presented including; visuo-Haptic illusions, haptic retargeting and geometric proximity and collision detection algorithms.

2.1 Visuo-Haptic Illusions

Visuo-haptic illusions leverage the dominance of visual perception over proprioception and haptic perception. When a conflict between the senses of vision and touch is introduced it is reported that an observer is more likely to perceive the visual shape over the tactile shape [16]. This also applies to discrepancies between vision and proprioception. The human proprioception sense provides an understanding of limb position through motor senses. When visually altered, a person is likely to perceive their limb where they visually see it, overriding their proprioception [11]. Furthermore, a person will compensate for visually presented fake, or adjusted movements that are spatiotemporally similar to their real movements [2, 20].

Pseudo-haptic illusions are a type of visuo-haptic illusion that leverages this visual dominance to create haptic feedback without physical haptic objects [41]. It can also increase the detail, or add feedback to physical haptics that does not match the real object [28, 35, 44]. This is achieved by changing or moving the visually perceived hand to emulate the effect of haptic feedback that is not present. Similarly, redirected touch is an illusion in which physical surfaces are warped by adjusting the perceived position of the user's hand in virtual reality as it touches the surface to be distorted [25]. Such illusions have also found applications in the optimisation of ergonomics in virtual environments [34], improvement of shape display performance [1], and deforming props [29, 50].

2.2 Haptic Retargeting

Haptic retargeting is a visuo-haptic illusion that enables the reuse of passive haptics for multiple virtual objects within a VE. This is achieved by gradually introducing an undetectable discrepancy between the physical hand, and the virtual hand that the user sees [4]. As the user moves to grab the virtual object, they unconsciously correct for the discrepancy and grab the physical object in a different spatial position. On-the-fly haptic retargeting is the current state-of-the-art haptic retargeting method that enables the user's interaction target to change after the retargeting has begun [13]. This technique has been explored for the extension of physical user interfaces [32], shifting the perception of weight with dynamic passive haptics [48], enhancing encounter type haptics [19], and general tactile interactions in virtual environments [37].

State-of-the-art haptic retargeting algorithms calculate a warp ratio and an offset to determine how far the actual and virtual positions should vary in real-time. The warp ratio r is the ratio of the distance D_p between the user's physical tracked hand position H_p and the physical target position P_p and D_0 between H_p and the origin position H_0 at which the haptic retargeting began as shown in Figure 1a.

$$D_p = \left|H_p - P_p\right|, D_0 = \left|H_p - H_0\right| \tag{1}$$

$$r = \left(\frac{D_0}{D_0 + D_p}\right) \tag{2}$$

The redirection offset W introduced between the physical and virtual hand is then a linear interpolation between the initial hand offset at the time the retargeting began I, and the offset between the physical and virtual target positions O.

$$O = P_v - P_p, I = H_{v0} - H_{p0} \tag{3}$$

$$W = rO + (1-r)I \tag{4}$$

This approach, and other implementations of haptic retargeting represent the hand as a single position on the hand; either the index fingertip, [13, 19, 32, 46, 50] or the palm [4, 14, 49]. Likewise the target is also often represented by a single position [4, 19, 32, 49].

As shown in Figure 1, the full target offset is applied to the hand when these positions perfectly align. That is $r = 1$ only when $D_p = 0$ and as such $D_0 = D_0 + D_p$. If the physical hand contacts the physical target when $D_p > 0$, there remains a discrepancy in the contact position between the virtual hand and target as compared the physical hand and target.

The discrepancy increases for larger objects such that touching the edge of a larger object using just the fingertips could result in the user missing the target entirely. The level of discrepancy is also dependant on the angular and translational gain. Previous research has determined angular gain up to 4.5°in either direction or 0.88 and 1.07 translational gain can go unnoticed [47]. The discrepancy may have minimal impact on the user's interactions at low gain values, or with small targets like a 5cm cube [4], or physical button [30, 32].

The user's hand in VR applications is typically represented by a deformable mesh linked to an associated armature. The armature is animated based on tracking information from a Leap Motion, Oculus Quest or other hand tracking systems, in turn matching the 3D mesh to the user's hand pose. Previous haptic retargeting implementations have represented the hand as a single position, typically placed at the likely location of interaction (such as the center of the hand [4] or the tip of the index finger [31, 32]). As such interactions with other parts of the hand, such as different fingers, will suffer from an incorrect RIA as described above.

Other haptic retargeting implementations do consider target shape. Cheng et al. retarget a user's hand to one of the primitives quadrilateral panels on their sparse haptic proxy to support haptic retargeting for multiple objects within a scene [13] and use a plane as the origin for their retargeting. Zhao and Follmer's functionally optimised remapping uses a geometric understanding of the targets to support mapping between different positions, as well as different orientations and shape [50]. This approach could provide support for multi-finger retargeting by sampling the fingertips of the hand against the mapping and using inverse kinematics to construct the rest of the joint angles. Each fingertip would correctly map to the surface, however, contact with any other part of the hand would likely result in incorrect RIA. The computation of the remapping is expensive and cannot be computed in real-time for VR applications (usually 90 fps).

Similarly, Carvalheiro et al. use a grid-based system to describe the amount of haptic retargeting distortion to apply [12]. The implementation also represents the hand and target as positions in space, however, it could be adapted to expand the distorted region to encompass the entire target. As the grid must be sampled by a single position the user a discrepancy would remain unless the hand's representative position is within the fully distorted region.

Other hand redirection techniques use a fixed offset introduced when the user is distracted or during a blackout period [7]. A fixed offset would provide correct RIA as the relationship between the virtual hand and target is identical to that of the physical hand and target. Similarly, the offset introduced during blink suppressed hand redirection is akin to a fixed offset [45]. When combined with a gain based offset, blink suppressed hand redirection could also make use of geometry to improve accuracy when applied to haptic retargeting.

The approach presented in this paper generalises existing haptic retargeting approaches to incorporate shape information and support accurate contact between any part of the physical and virtual hand and any part of their respective targets. Leveraging the shortest distance between the hand and target when computing r, as opposed to the distance between their positions, ensures $r = 1$ when any part of the physical hand touches any part of the physical target. Haptic retargeting systems require an understanding of the user's interaction target. Combined with target selection methods, such as head gaze [31], eye gaze [13], or Unscripted Retargeting, a neural-network-based target prediction system [14], SAHR can provide natural, unscripted interactions using haptic retargeting.

2.3 Proximity and Collision Detection

Efficient calculation of the shortest distance between primitive shapes, and between two geometric mesh representations, is a widely explored problem in computer graphics. The proximity of primitive shapes such as spheres, swept spheres (also called capsules), axis-aligned and object-oriented boxes, and infinite planes can be calculated through their implicit mathematical representations [9, 15, 23]. For example, sphere to sphere distance is the length of the vector between their center positions minus the sum of their radii. A positive distance represents their shortest distance and a negative distance indicates the occurrence of a collision and the penetration depth.

The Gilbert-Johnson-Keerthi (GJK) algorithm and Lin-Canny are widely known and explored algorithms for computing collisions between two convex sets of points or convex geometry [17, 27]. In addition to collision, these also provide the shortest distance between the convex sets and the points between which the shortest distance lies. In GJK, collision detection is achieved by algorithm computing the nearest simplex to the origin within the Minkowski Difference of the two meshes. In a 3D implementation, a tetrahedron simplex that encompasses the origin is considered a collision or intersection, while a point, line or triangle simplex indicates no collision or intersection is occurring. This algorithm can suffer from inconsistencies due to floating-point precision, and much work has been done towards a robust solution [8, 33]. GJK can operate on concave meshes however the nature of the algorithms means only the convex hull of the vertices will be considered.

An exact distance between concave meshes can be computed using a brute force approach by comparing the distance from every polygon on one mesh to every polygon on the other can produce a highly accurate and robust result. Such implementations are highly suited for parallelism which can provide significantly improved performance when run on a computer's graphics processing unit (GPU). A brute force approach could be further optimised through the use of a bounding volume hierarchy (BVH) that separates the mesh into a tree of bounding boxes, with which the nearest bounding volume can be computed more efficiently, and the requisite polygon-polygon checks can be reduced. Construction of a BVH can be computationally expensive, as such construction of a BVH for a dynamically changing mesh, such as the deforming mesh of a tracked hand, is not a viable approach. There has been extensive research to provide more efficient solutions to this problem, Tsechner et al. provide an excellent survey of collision detection and proximity algorithms for deformable meshes [42]. Gissler et al. proposed a proximity query algorithm that supports concave meshes leveraging GJK to divide the concave mesh into smaller submeshes however it required additional work to improve efficiency [18].

For SAHR, we draw upon primitive shapes to provide approximate distance between the user's hand and physical haptic object. We compare this to a GPU based brute force triangle-triangle approach to compute the proximity of the user's dynamically updating hand geometry to the geometry of the physical haptic target.

3 Shape Aware Haptic Retargeting

SAHR aims to ensure the correct RIA during interactions that leverage haptic retargeting. This is achieved by computing the shortest distance between the geometry of the target and the user's hand. When a user's hand approaches the target, D_p approaches zero and upon contact when $D_p = 0$ the full amount of haptic retargeting is applied and the correct RIA is achieved. Introducing target geometry corrects the RIA for index fingertip interactions (Figure 2a and b) and the addition of hand geometry further corrects the RIA for interactions with the entire hand (Figure 2c and d). For this approach we assume that the physical and virtual target geometries are the same.

When using single positions to represent the hand and target, a discrepancy occurs between the expected and actual virtual hand positions, as shown in Figure 1b. This error can be quantified as shown in Equation 5 where s is the maximum warp ratio (1.0) minus the current warp ratio r, by interpolating between I and O as found in Equation 3.

$$E = sO + (1-s)I, s = (1-r) \tag{5}$$

To correct this, SAHR incorporates a generalized distance equation that considers the hand-to-target distance D_p as the shortest distance between the surface of the hand to the surface of the target. To ensure realignment of the virtual and physical hands when any part of the hand contacts the origin, the origin-to-hand distance D_0 is also generalized to the shortest distance from the origin position (or surface for a threshold origin [13, 22, 30]) to the physical hand.

For this implementation, On-The-Fly retargeting [13] as described in Equation 2 to Equation 4 is used as the basis upon which SAHR is applied. This solution can also be integrated with other haptic retargeting implementations including hybrid warping [4] and remapped reach [22]. Equation 6 outlines the generalized method for computing D_0 and D_p in combination with three shape approximation methods outlined in subsection 3.1 to subsection 3.3: Multi-Point, Primitive and Mesh Geometry, to find the shortest distances.

If $\mathbb{H}$ describes the surface of the hand representation and $\mathbb{T}$ describes the surface of the target representation. Let $\mathbb{R}$ be the Cartesian product between all points on $\mathbb{H}$ and $\mathbb{T}$ ($\mathbb{R} = \mathbb{H} \times \mathbb{T}$). In Equation 6, S_H, S_T is the pair in $\mathbb{R}$ where the distance $|S_H - S_T|$ is the

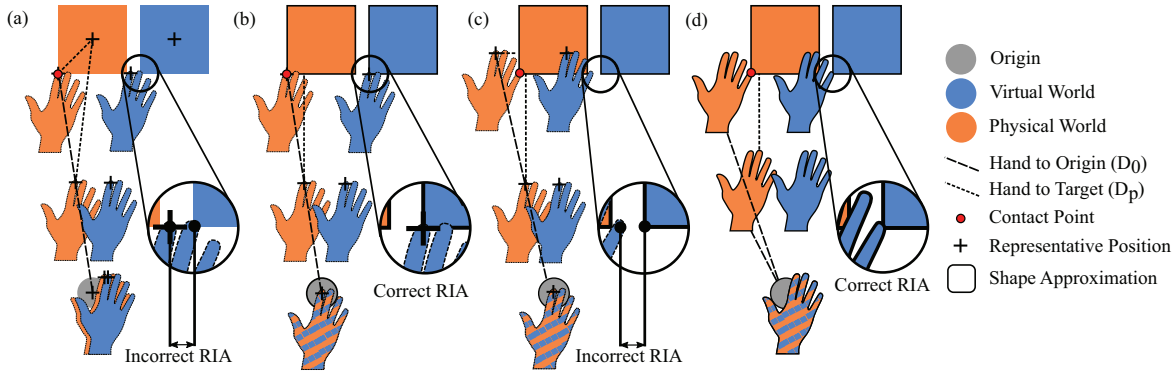


Figure 2: Using a single point to represent the target (a) results in the virtual hand contacting the target in the wrong position. Adding shape information to the origin and target corrects the error (b), however, this error returns when touching with different fingers (c). Adding shape information of the hand (d) then ensures a correct contact position.

minimum distance between $\mathbb{H}$ and $\mathbb{T}$. As such, when the surface distance between the target and hand D_H reaches zero the warp ratio r in Equation 2 reaches it's maximum value of $r = 1.0$ and the correct RIA is achieved.

$$D_p = |S_H - S_T|, D_0 = |S_{H0} - S_0| \tag{6}$$

Likewise, the origin may also not be a single point in space, rather an infinite plane [13] or a sphere threshold encompassing the target [22]. If $\mathbb{O}$ describes the surface of the origin, let $\mathbb{U}$ be the Cartesian product between all points on $\mathbb{H}$ and $\mathbb{O}$. S_{H0}, S_0 is the pair in $\mathbb{U}$ where the distance $|S_{H0} - S_0|$ is the minimum distance between $\mathbb{O}$ and $\mathbb{H}$. Provided the system understands the geometry of the hand, target and origin, the solution for Equation 6 can be found. We propose three shape approximation variants between which the nearest positions can be identified to populate the new distance equation. We have named the three methods; Multi-Point, Primitive and Mesh Geometry and provide descriptions of each.

3.1 Multi-Point Approximation

The Multi-Point approximation is most similar to existing haptic retargeting techniques and inspired by the method proposed by Zhao and Follmer to retarget the user's fingers between differing shapes [50]. Here multiple arbitrary reference points are defined for the hand, target and origin. The distance is computed from all n reference points on the hand to all m reference points on the target or origin and the minimum distance is identified. The identified minimum distance is then used to compute the redirection ratio (r), which in turn is used to compute the discrepancy to apply to the tracked and virtual hands. This approach will only guarantee that $r = 1.0$ when at least one of the reference points on the hand are exactly aligned with one of the reference points on the target. As such, the resulting relative accuracy of this approach is dependent on the number of reference points and the distribution of the reference points. Reference points should ideally be placed in positions where contact with another reference point is more likely, such as on the fingertips of the hand as in Figure 3c. Furthermore for uniformly distributed reference points the accuracy of the RIA is expected to increase for larger values of n and m.

3.2 Primitive Approximation

The Primitive approximation leverages an approximation of the tracked hand using primitive shapes for which proximity queries are considerably more efficient. For example the hand geometry can be approximated using a set of capsules attached to the tracked hand joints that encapsulate the fingers between each joint as in Figure 3d. Likewise, the target and origin can be approximated using a union of primitive shapes such as spheres, capsules, boxes and infinite planes. D_p and D_0 for Equation 6 are computed by calculating, and selecting the shortest distance between each primitive in the hand and each primitive in the target and/or origin. The algorithms used to compute the nearest distances between primitive shapes are drawn from Ericson's Real-Time Collision detection [15]. A set of primitive distance calculations were implemented for Box, Sphere, Capsule (Swept-Sphere) and Plane.

Using this approach, the shortest distance between the tracked hand approximation and both the warp origin approximation and the tracked target approximation are computed. These distances are set to zero whenever the hand primitives are intersecting with the target or origin primitives. They are then used to compute the warp ratio (r) which in turn is used to compute the discrepancy to apply to the tracked and virtual hands. This approach ensures $r = 1.0$ when the tracked hand approximation is in contact with the tracked target approximation, resulting in the virtual hand touching the virtual target in the same relative position. The primitive implementations can also compute the shortest distance to a position, providing integration with the Multi-Point approximation, for example; supporting retargeting of a Multi-Point hand to a Primitive target.

3.3 Mesh Geometry Approximation

In most graphics engines and VR hand tracking software, objects within the VE, and the pose of a user's tracked hand is represented using triangulated mesh geometry. This representation leverages this geometry to provide an understanding of the shape of both the user's hand and the haptic retargeting target and in turn compute D_p and D_0 for Equation 6.

Modern hand tracking for VR typically leverages a deformable mesh and armature to represent the user's hand pose and shape. To find S_H and S_T, the triangles of the hand are deformed within a compute shader based on the armature pose using the bone transformations and vertex weights provided within the Unity game engine. The shortest distance and nearest points are computed using a brute force, GPU accelerated method within the same compute shader. The triangle distance computation algorithm is based on Shellshear's implementation [39] of the triangle distance function in the Proximity Query Pack [26], modified to run as an HLSL compute shader for the Unity game engine. The distance from one triangle on the hand geometry to every triangle on the target geometry is computed in independent threads and written to a collective buffer. The overall minimum is then found on the CPU. Both the Multi-Point and Primitive representations were integrated into the mesh distance compute shader and used wherever Mesh to Multi-Point or Mesh to Primitive calculations are required.

The distance between the tracked hand mesh and both the origin

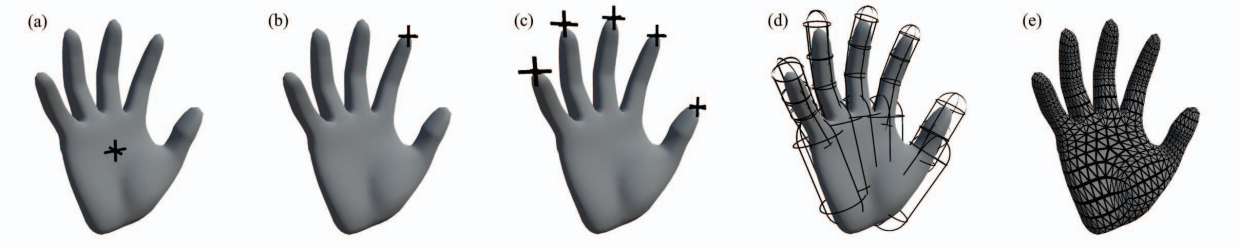


Figure 3: The five hand representations used for the simulated interactions: a) Palm (PP), b) Index (IP), c) Fingertips (FP), d) Primitive (PH), e) Mesh Geometry (MH).

(D_0), and the tracked physical object (D_p) are computed using this technique. The resulting distances are used to compute the warp ratio (r) and in turn to compute the warp offset between the tracked and virtual hands. This ensures the warp ratio reaches $r = 1.0$ when the tracked hand geometry is in contact with the tracked target geometry resulting in the correct RIA. As a result, the virtual hand touches the virtual target in precisely the same position as the physical hand touches the physical target. Likewise, measuring D_0 as the distance from the origin to the hand surface ensures the physical and virtual hands return to alignment upon initial contact with the origin.

3.4 Comparison of Proximity Algorithms

Each shape approximation and their related proximity calculations have benefits and drawbacks for the accuracy and computational performance. All of these approaches have quadratic efficiency which is sub-optimal for complex representations. This is particularly prevalent in the Mesh Geometry approximation as the number of triangles increases. The Primitive and Multi-Point approximation the proximity calculations are considerably more efficient.

To quantify the relative performance of each shape approximation, the shortest distance computation time was logged using the .NET 4 System.Diagnostics Stopwatch class, 60 times per second over 5 seconds. These tests were performed on a desktop graphics computer with an Intel i7-10700F and (2.90ghz) and an Nvidia RTX 2070 Super in the Unity 2019.4.22f1 game engine. The mean computation times for each configuration are outlined in Table 1. We consider real-time to be approximately 90 frames per second (fps), a frame rate used in many VR head mounted displays. The worst-case performance of the Hand mesh to Teapot mesh provides a theoretical maximum of 87fps, however in practice, the target mesh can be simplified to approximate the shape.

Table 1: Mean computation time in microseconds for variations for combinations of hand and target approximation over 5 seconds. Highlighted cells indicate combinations that leverage the compute shader mesh distance implementation.

Target	Multi-Point Fingertips	Capsule Hand	Hand Mesh: 3270 Tri
Single Position	3.91μs	49.90μs	1074.80μs
Cube: 12 Tri	161.17μs	527.90μs	755.07μs
Button: 1736 Tri	421.59μs	486.32μs	3845.22μs
Teapot: 6320 Tri	317.84μs	720.32μs	11432.47μs
Box Primitive	216.49μs	491.68μs	862.16μs
Sphere Primitive	5.58μs	63.06μs	594.83μs
Capsule Primitive	8.01μs	53.24μs	750.75μs
Ten Boxes Primitives	1436.75μs	2046.66μs	664.11μs
Ten Sphere Primitives	38.08μs	263.56μs	364.18μs

The brute force approach used in the Mesh Geometry provides sub-optimal computational efficiency as compared to state-of-the-art algorithms, to combat this, parallel computation is employed to reduce computation time. The GJK algorithm [17] was explored for this method, however, the general implementation is limited to convex shapes and can prove a challenge to implement due to floating-point precision and edge cases. Other deformable mesh proximity algorithms outlined in the related work were considered [18, 42], however, a parallel brute force approach was chosen for it's robustness and ease of implementation.

The Multi-Point approach requires some manual consideration during implementation to determine the ideal reference points. For example, use of the fingertips as the reference points may be unsuitable for interactions in which the fingertip does not contact the target. Furthermore, correct RIA is only achieved when the hand and target points are perfectly aligned, thus more points is likely to increase the resulting accuracy.

As the Primitive approximation typically extends beyond the hand and target, this can result in the completion of retargeting *before* the real hand makes contact with the real target. While this does guarantee correct RIA during interactions, it produces a visual artefact where the hand changes direction the offset stops increasing at the point the primitive approximations intersect. This reduces the overall distance between origin and target, increasing the angular and translational gain required to complete the retargeting, affecting the detectability of the retargeting [47].

4 Simulated Evaluation

In the previous section, we evaluated the computational efficiency of the various shape approximations. In this section, we explore the effect of SAHR and each approximation on interaction accuracy and the amount of distance error and correctness of the RIA. Equation 5 provides a simple method to find the amount of error, however, it does not capture the effect on realistic interactions.

Initial testing was performed using motion-captured interactions. The interactions were recorded using a Leap Motion mounted above a desk to provide a fixed tracking origin. Due to the inaccuracy of the Leap Motion device, particularly while touching physical objects, these were discarded in favour of artificial recreations of the motion to represent a best-case tracking scenario. The limitations of tracking technology for SAHR are further explored in section 5. Instead, simulated evaluations were performed to quantify the retargeting accuracy provided by SAHR for various interaction types. Each simulated evaluation uses an animated hand artificially posed realistically as if it were performing the given action.

The simulated interactions are drawn from previous applications of haptic retargeting with five hand and three target representations for each simulation. The five hand configurations are evaluated (see Figure 3) including two single position representations (Palm [4, 14, 49] and Index Fingertip [13, 19, 32, 46, 50]). The remaining three hand configurations use the SAHR shape approximations: Multi-Point (Fingertips), Primitive and Mesh Geometry. The Primitive hand approximation utilises capsule primitives extending the length of each of the hand's bones with a radius of 20mm for the thumb and 10mm for the other fingers. The palm is approximated with 5

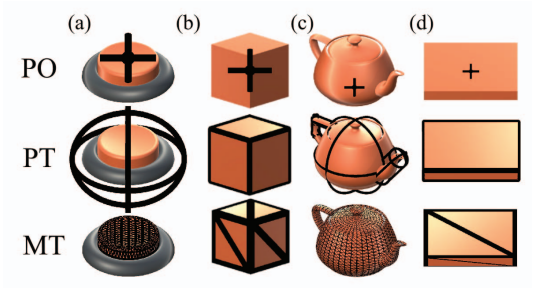


Figure 4: Shape approximations for the target in each simulation: multi-finger (a), grasping (b), complex object (c), and continuous interaction (d). Each target has a Position (top), Primitive (center) and Mesh (bottom) approximation

capsules of radius 20mm extending from the wrist to the base of each finger. The Mesh Geometry hand utilises a deformable rigged hand model with 3236 triangles tied to an armature representing the bones of the hand. The Palm and Index hands use a single point to approximate the target placed at the palm and index fingertip respectively. Finally, the Multi-Point configuration includes five reference points, one at each fingertip.

In *all* simulations the physical target is placed within the fifth percentile of human reach envelopes at 60cm from the origin [36] as shown in Figure 5d. The virtual target is spaced 10 cm to the right of the physical target. The exact configuration of each target is dependent on the simulation in question and is outlined in subsection 4.1 to subsection 4.4. The three target configurations: Position, a single point at the center of the target, Primitive and Mesh Geometry (static) are shown in Figure 4 and Figure 5 shows the Unity setup for each simulation. The identifiers for each combination of hand and target configuration are outlined in Table 2.

Four simulations were performed in total: pressing a button, grasping a cube, touching a complex object, and tracing the edge of a box. In all simulations, excluding the continuous interaction simulation (subsection 4.4), the RIA error was calculated and recorded one time when the hand was correctly positioned and posed. In the continuous interaction simulation, the RIA error was calculated and stored every 50ms. In each simulation, the RIA error is measured as outlined in Equation 7. The error X is the Cartesian distance in millimeters between the expected hand position H_e in Figure 1 (given by the relative position of the physical hand H_p to the physical target P_p, offset by the virtual target position P_v) and the virtual hand position H_v.

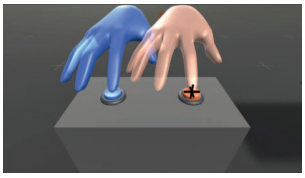
$$X = |H_e - H_v|, H_e = (P_p - H_p) + P_v \tag{7}$$

Table 2: Hand and Target identifiers for the simulations, highlighted cells indicate our SAHR shape approximation implementations

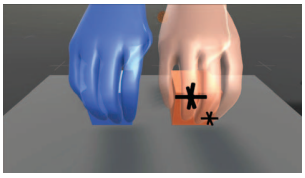
	Hand Representation		Target Representation
IP	Index Position	PO	Position
PP	Palm Position	PT	Primitive Target
FP	Fingertip Positions	MT	Mesh Geometry Target
PH	Primitive Hand		
MH	Mesh Geometry Hand		

4.1 Multiple Finger Interaction

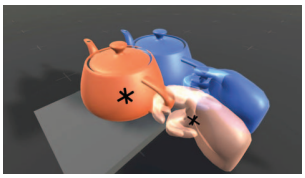
The **multiple finger interaction** simulation simulates pressing a retargeted button with each finger. Previous retargeting systems that explore interactions with physical buttons have been limited to the use of only one finger. In natural button operations, a user might use



(a) "Multi-Finger" Setup Setup



(b) "Grasping" Setup



(c) "Complex Object" Setup

(d) "Continuous Interaction" Setup

Figure 5: The virtual configuration for each simulation. Orange represents the artificial physical world that would be provided by some external tracking of the hand and target while blue represents the virtual world that a user would see.

different fingers to press buttons. Using an open hand pose shown in Figure 5a, the hand is placed in a natural position to press the button with each fingertip in the order: thumb, index, middle, ring, pinky. The RIA error is calculated for each simulated press when the hand is in position. The target is a circular button of radius 10mm approximated using a single position in the center and on the surface of the button for PO, a sphere of radius 20mm encompassing the button for PT and the mesh of the button containing 1736 triangles for MT (see Figure 4a).

Results from the simulation, outlined in Figure 6 show our new PH provided correct RIA, with an error of 0.000mm for when interacting with all fingers and all target approximations (PO, PT and MT). The new MH also provided correct RIA for MT and PT with an error of 0.000mm. The IP hand results clearly indicate the issue with multi-finger interaction that SAHR addresses. The IP hand index index finger showed an error of 0.000 across all target types, yet this increased as high as 16.951 for the pinky finger. For the new FP hand, the error is no greater than 1.436mm, demonstrating this would be a suitable approximation for button interactions while retaining multi-finger support.

4.2 Grasping

The **grasping** simulation is intended to explore the effect of SAHR on interactions with the entire hand. Research involving such interactions has involved grasping the target object and moving it around in space [4, 22]. The simulated hand was posed into a natural grasping position such that the tips of all four fingers and the thumb were in contact with the cube as shown in Figure 5b, then the RIA error was calculated. The target is a 50mm cube approximated in PO using a single position in the center of the cube, in PT with a primitive cube of the same size, and in MT with the mesh of the cube containing 12 triangles (see Figure 4b).

As shown in Figure 7 the RIA error from the grasping simulation ranges from 7.902 to 20.206mm with the sub-optimal reference position in the existing PO. In the button pressing simulation the reference position is on top of the button and thus for IP and FP the error still reached the ideal 0.000mm. However, in this case where the PO position is in the center, all position type hand simulations (IP, PP and FP) gave an error $>$10mm. The existing PP hand representation, previously used in grasping tasks [4] provided a minimum error of 12.099mm. The new PH-PT, PH-MT, MH-PT and MH-MT configurations all achieved a correct RIA with error of 0.000mm.

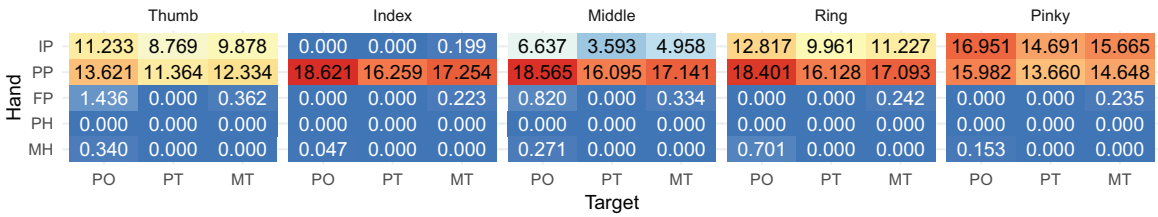


Figure 6: RIA error in millimeters for the button pressing simulation using each finger

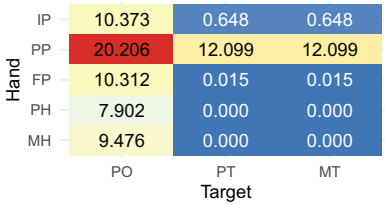


Figure 7: RIA error in millimeters for the cube grab simulation

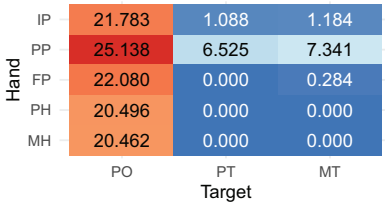


Figure 8: RIA error in millimeters for the complex object simulation

4.3 Complex Object

The **complex object** simulation explores the performance of SAHR for complex objects with a high number of polygons and concave components that are best approximated with multiple primitives. The complex object used for this simulation is the Utah teapot model scaled to a size of 260x120x160mm. Figure 4c shows the target configuration with PO using a single position in the center of the teapot, and MT using the teapot mesh containing 6320 triangles. In PT the teapot is approximated by multiple primitives, a sphere of radius 80mm encompassing the main body of the teapot, and 4 capsules, one of radius 20mm encompassing the spout and three of radius 10mm approximating the handle. As shown in Figure 5c, the hand grasps the teapot with its index and middle finger through the handle, with the thumb resting on top of the handle. The skin of the proximal and middle phalanx of the index and middle finger, as well as the tip of the thumb, are in contact with the handle. The RIA error is then computed with the hand in this position.

The RIA error for the complex object simulation, outlined in Figure 8 visualises the impact of the physical object's size on the existing PO target approximation. Due to the distance between the reference position in the center of the teapot and the handle, the error for all PO simulations is >20mm, and the existing PP hand representation had an error >6mm. As with the cube grasping simulation, the new PH-PT, PH-MT, MH-PT and MH-MT all achieved the ideal error of 0.000mm, as did our new FP-PT.

4.4 Continuous Interaction

The **continuous interaction** simulation simulates interactions in which a user explores various parts of a target with their fingertips and hand, in this case, tracing the edge of a box with the index finger. This visualises the difference in hand movement between the simulated real and virtual hands. The key area of interest is the interaction itself while the hand is in contact with the box. With correct RIA, the path taken by the virtual hand around the virtual target should be identical to the physical hand path around the physical target.

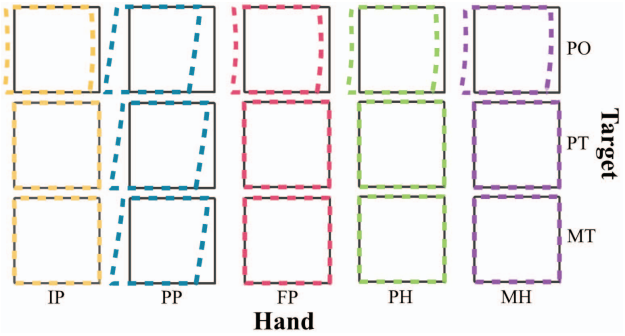


Figure 9: Path taken by the virtual index fingertip around the virtual box during the continuous interaction simulation. The black line is the path taken by the physical fingertip around the physical box.

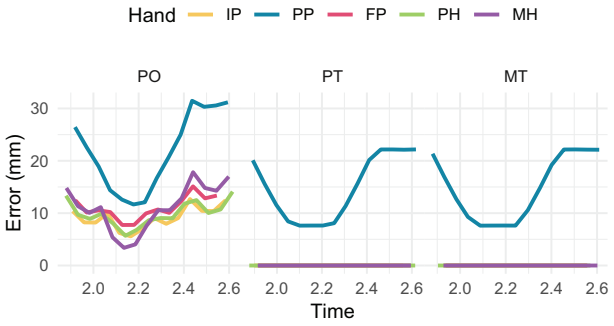


Figure 10: Error in millimeters during the Interact phase of the continuous interaction simulation

During the interaction the hand traces the top edge of the cube, starting from the bottom right corner as seen from the origin (bottom left of the target in Figure 5d). The hand moves at a rate of 40mm/s and touches the box using only the index fingertip. The target is a 100mm x 100mm x 10mm box approximated in PO using a single position in the center of the box on its top face, in PT with a primitive box of the same size, and in MT with the mesh of the box containing 12 triangles (see Figure 4d).

The path and error data from this simulation visualise the impact of an incorrect RIA. As shown in Figure 9, the path of the virtual fingertip (colored) varies from the expected path in all PO simulations and all PP simulations. Figure 10 visualises the change in RIA error during the Interact phase. From this we see that in PO the error never reaches 0.000mm, and the error varies considerably throughout the interaction. For both the new PT and MT, the error is 0.000mm for all hands except for PP. This demonstrates that SAHR, ensures consistency between the visual contact and physical contact as the hand explores the target. Where existing approaches PO and PP diverge from at the corners of the box, the new PT and MT target representations ensure the existing IP, and new FP, PH and MH hand representations precisely follow the edge and corners of the box.

5 Discussion and Limitations

The simulated interactions showed the Shape Aware Haptic Retargeting implementation coupled with the Multi-Point, Primitive and Mesh Geometry approximations demonstrably improve the Retargeted Interaction Alignment for haptic retargeting as compared to a single representative position. This ensures the virtual hand-target relationship matches the physical hand-target relationship upon first contact between the hand and target. Furthermore, the simulations demonstrate that this approach also provides accurate interactions when touching the target using other fingers and other parts of the hand, thereby enabling more natural interactions for the user.

SAHR, and in particular the Mesh Geometry approximation, make two key assumptions about the overall system implementation. Firstly, the hand and target geometry within the VE closely represent their real position, size, pose and shape. Secondly, the hand and object tracking and coordinate systems are closely aligned such that the spatial relationship between the real physical hand and target matches the relationship as represented in the VE.

The Primitive approximation used in the simulations overestimates the true size of the hand. As a result, the Primitive hand contacts the target before the actual hand, guaranteeing a correct RIA. The mesh representations show slight variations where the polygons of the mesh may not fully intersect with the target, leaving a small gap and creating a small error. This highlights that the mesh of the hand should accurately represent, or slightly over-estimate the physical hand's size and shape for the best RIA. As the hand approaches the target, an over-estimation using primitives causes the gradual increase in gain to stop prematurely when the approximations first intersect. This could result in a visual change in movement direction when the gradually increasing gain becomes a constant offset. Ideally, the Primitive approximations should match the real hand and target geometry as closely as possible to minimise the gap when the hand and target primitives first intersect.

Modern hand tracking technology is not yet able to provide such an accurate understanding of the user's hand. Devices like the Oculus Quest 2 and Leap Motion provide pose, position, and finger length [21, 38, 43], however, the mesh used is usually a generic hand model. In future, 3D scanning technology could produce accurate models of the user's hands and correct this. Smith et al. present a method for accurately fitting a mesh to the surface of a user's hand using 124 cameras [40]. In future, this technology could become a viable commercial hand tracking solution, providing an accurate understanding of the real hand surface.

The simulations show SAHR improves RIA with correctly configured shape approximations. However, a user evaluation could determine the detectability of an incorrect RIA and how SAHR affects detection thresholds of hand redirection and real interaction accuracy. SAHR has been shown to reduce both visual interpenetration and visual disconnect between the finger and target. As demonstrated in the simulation, haptic retargeting without SAHR can result in unexpected motion, particularly visible in the continuous interaction simulation paths in Figure 9. We hypothesise that users may not notice small discrepancies in movement, however, they would notice their virtual finger penetrating the surface, visually disconnecting from the surface prematurely or missing the target.

Gonzalez-Franco et al. identified the self-avatar follower effect, finding that when allowed to move freely, a person unconsciously corrects for sensory conflicts between their real body positions and that of their avatar in VR [20]. In haptic retargeting the user typically has an explicit goal, leading them to instead correct the error between their goal and their motion, as opposed to correcting for the sensory conflict. There may be differences between our simulated interactions and actual human interactions that a user evaluation could highlight.

As with most haptic retargeting implementations, SAHR assumes similarity in size, shape and orientation of the physical and virtual targets. Other visuo-haptic illusions can modify the perceived shape [5, 50], size [3, 6] and orientation of an object [50]. Zhao and Follmer's functionally optimised remapping method is similar to SAHR and does support shape and orientation remapping [50]. Where shape and orientation remapping is not required, SAHR would be the preferable method as it is simple, extends existing haptic retargeting methods, is computed in real-time, and it supports contact with any part of the user's hand.

Zhao and Follmer's Shape Changing Texture Map could be combined with SAHR. The map describes the offset between points on an object's surface and points on another object of differing shape [50]. This offset could be sampled from the map using the nearest position between the hand and mesh then added to the retargeting offset. This would enable visuo-haptic shape mapping in conjunction with SAHR and On-The-Fly haptic retargeting.

As it is an extension of existing state-of-the-art algorithms, SAHR can be easily integrated into most haptic retargeting implementations and utilised for future research. In addition, integration with more optimised state-of-the-art proximity algorithms would provide more performant distance computation, however, the implementations presented cover a broad spectrum from a single point-to-point distance to a brute force concave mesh distance calculation and provide acceptable performance for real-time systems.

We propose that with SAHR, future haptic retargeting systems can support more complex interactions than those explored in previous work. Representing the entire geometry of the target and hand enables the user to freely touch any part of the target with any part of their hand. In addition, much larger targets can be utilised to support and enhance new and existing applications of haptic retargeting. One such example is remapped physical-virtual interfaces [32], in which the entire interface could be retargeted, as opposed to individual buttons. This would enable a user to operate a remapped button, and immediately operate another nearby physical button with no need for additional retargeting. SAHR can also be combined with target selection methods such as Unscripted Retargeting [14] to enable a level of natural target selection and freedom of interaction previously not possible with haptic retargeting.

6 Conclusion

Shape Aware Haptic Retargeting is an elegant extension to state-of-the-art haptic retargeting that can easily integrate with existing and future implementations. Combined with knowledge of hand and target geometry, our implementation enables all parts of a user's hands to touch and explore complex, and large physical objects whilst maintaining consistency with the visual feedback presented in VR. Although these approaches are more computationally expensive, we anticipate this capability will be widely used for more complex interactions and may include implementations such as multi-touch in user interfaces, surface exploration in design applications and as a new gaming mechanic. While previous haptic retargeting systems have relied upon a single position to represent the hand, target and origin, in Shape Aware Haptic Retargeting the hand-target-origin distance is determined by the proximity between their respective shape approximation. Three shape approximations were evaluated through simulated interactions demonstrating that with accurate tracking of both the hand and target, our approach improves interaction accuracy under the effect of haptic retargeting. Combined with these approximations, Shape Aware Haptic Retargeting enables more natural interactions that could lead to new and exciting applications of haptic retargeting and hand redirection in general.

Acknowledgments

This research is supported by an Australian Government Research Training Program Scholarship.

REFERENCES

[1] P. Abtahi and S. Follmer. Visuo-Haptic Illusions for Improving the Perceived Performance of Shape Displays. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pp. 1–13. ACM Press, 2018. doi: 10.1145/3173574.3173724

[2] T. Asai. Feedback control of one's own action: Self-other sensory attribution in motor control. *Consciousness and Cognition*, 38:118–129, Dec. 2015. doi: 10.1016/j.concog.2015.11.002

[3] J. Auda, U. Gruenefeld, and S. Schneegass. Enabling Reusable Haptic Props for Virtual Reality by Hand Displacement. In *Mensch und Computer 2021*, MuC '21, pp. 412–417. Association for Computing Machinery, New York, NY, USA, Sept. 2021. doi: 10.1145/3473856.3474000

[4] M. Azmandian, M. Hancock, H. Benko, E. Ofek, and A. D. Wilson. Haptic Retargeting: Dynamic Repurposing of Passive Haptics for Enhanced Virtual Reality Experiences. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, pp. 1968–1979. ACM Press, 2016. doi: 10.1145/2858036.2858226

[5] Y. Ban, T. Kajinami, T. Narumi, T. Tanikawa, and M. Hirose. Modifying an identified curved surface shape using pseudo-haptic effect. In *2012 IEEE Haptics Symposium (HAPTICS)*, pp. 211–216, Mar. 2012. ISSN: 2324-7355. doi: 10.1109/HAPTIC.2012.6183793

[6] Y. Ban, T. Narumi, T. Tanikawa, and M. Hirose. Modifying Perceived Size of a Handled Object through Hand Image Deformation. *Presence: Teleoperators and Virtual Environments*, 22(3):255–270, Aug. 2013. doi: 10.1162/PRES_a_00154

[7] B. Benda, S. Esmaeili, and E. D. Ragan. Determining Detection Thresholds for Fixed Positional Offsets for Virtual Hand Remapping in Virtual Reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 269–278, Nov. 2020. ISSN: 1554-7868. doi: 10.1109/ISMAR50242.2020.00050

[8] G. V. d. Bergen. A Fast and Robust GJK Implementation for Collision Detection of Convex Objects. *Journal of Graphics Tools*, 4(2):7–25, Jan. 1999. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10867651.1999.10487502. doi: 10.1080/10867651.1999.10487502

[9] G. V. D. Bergen and G. J. Bergen. *Collision Detection*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1 ed., 2003.

[10] E. Burns, S. Razzaque, A. T. Panter, M. C. Whitton, M. R. McCallus, and F. P. Brooks. The hand is more easily fooled than the eye: users are more sensitive to visual interpenetration than to visual-proprioceptive discrepancy. *Presence: Teleoperators and Virtual Environments*, 15(1):1–15, Feb. 2006. doi: 10.1162/pres.2006.15.1.1

[11] E. Burns, S. Razzaque, M. C. Whitton, and F. P. Brooks. MACBETH: The avatar which I see before me and its movement toward my hand. In *2007 IEEE Virtual Reality Conference*, pp. 295–296, Mar. 2007. ISSN: 2375-5334. doi: 10.1109/VR.2007.352509

[12] C. Carvalheiro, R. Nóbrega, H. da Silva, and R. Rodrigues. User Redirection and Direct Haptics in Virtual Environments. In *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*, pp. 1146–1155. ACM Press, 2016. doi: 10.1145/2964284.2964293

[13] L.-P. Cheng, E. Ofek, C. Holz, H. Benko, and A. D. Wilson. Sparse Haptic Proxy: Touch Feedback in Virtual Environments Using a General Passive Prop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, pp. 3718–3728. ACM Press, 2017. doi: 10.1145/3025453.3025753

[14] A. Clarence, J. Knibbe, M. Cordeil, and M. Wybrow. Unscripted Retargeting: Reach Prediction for Haptic Retargeting in Virtual Reality. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pp. 150–159, Mar. 2021. ISSN: 2642-5254. doi: 10.1109/VR50410.2021.00036

[15] C. Ericson. *Real-Time Collision Detection*. CRC Press, Inc., USA, 2004.

[16] J. J. Gibson. Adaptation, after-effects, and contrast in the perception of curved lines. *Journal of Experimental Psycholology*, 16(1):1–31, 1933. doi: 10.1037/h0074626

[17] E. Gilbert, D. Johnson, and S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal on Robotics and Automation*, 4(2):193–203, Apr. 1988. Conference Name: IEEE Journal on Robotics and Automation. doi: 10.1109/56.2083

[18] M. Gissler, C. Dornhege, B. Nebel, and M. Teschner. Deformable Proximity Queries and Their Application in Mobile Manipulation Planning. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, Y. Kuno, J. Wang, J.-X. Wang, J. Wang, R. Pajarola, P. Lindstrom, A. Hinkenjann, M. L. Encarnação, C. T. Silva, and D. Coming, eds., *Advances in Visual Computing*, Lecture Notes in Computer Science, pp. 79–88. Springer, Berlin, Heidelberg, 2009. doi: 10.1007/978-3-642-10331-5_8

[19] E. J. Gonzalez, P. Abtahi, and S. Follmer. REACH+: Extending the Reachability of Encountered-type Haptics Devices through Dynamic Redirection in VR. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, pp. 236–248. Association for Computing Machinery, New York, NY, USA, Oct. 2020. doi: 10.1145/3379337.3415870

[20] M. Gonzalez-Franco, B. Cohn, E. Ofek, D. Burin, and A. Maselli. The Self-Avatar Follower Effect in Virtual Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 18–25, Mar. 2020. ISSN: 2642-5254. doi: 10.1109/VR46266.2020.00019

[21] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, and J. Sodnik. An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. *Sensors (Basel, Switzerland)*, 14(2):3702–3720, Feb. 2014. doi: 10.3390/s140203702

[22] D. T. Han, M. Suhail, and E. D. Ragan. Evaluating remapped physical reach for hand interactions with passive haptics in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1467–1476, Apr. 2018. doi: 10.1109/TVCG.2018.2794659

[23] M. Held. ERIT—A Collection of Efficient and Reliable Intersection Tests. *Journal of Graphics Tools*, 2(4):25–44, Jan. 1997. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10867651.1997.10487482. doi: 10.1080/10867651.1997.10487482

[24] B. E. Insko. Passive Haptics Significantly Enhances Virtual Environments. Technical Report 0-493-17286-6, The University of North Carolina at Chapel Hill, 2001.

[25] L. Kohli, M. C. Whitton, and F. P. Brooks. Redirected Touching: Training and adaptation in warped virtual spaces. In *IEEE Symposium on 3D User Interface 2013, 3DUI 2013 - Proceedings*, pp. 79–86. IEEE, Mar. 2013. doi: 10.1109/3DUI.2013.6550201

[26] E. Larsen and S. Gottschalk. PQP - A Proximity Query Package, June 1999.

[27] M. Lin and J. Canny. A fast algorithm for incremental distance calculation. In *1991 IEEE International Conference on Robotics and Automation Proceedings*, pp. 1008–1014 vol.2, Apr. 1991. doi: 10.1109/ROBOT.1991.131723

[28] A. Lécuyer. Simulating Haptic Feedback Using Vision: A Survey of Research and Applications of Pseudo-Haptic Feedback. *Presence*, 18(1):39–53, Feb. 2009. Conference Name: Presence. doi: 10.1162/pres.18.1.39

[29] K. Matsumoto, T. Hashimoto, J. Mizutani, H. Yonahara, R. Nagao, T. Narumi, T. Tanikawa, and M. Hirose. Magic table: deformable props using visuo haptic redirection. In *SIGGRAPH Asia 2017 Emerging Technologies on - SA '17*, pp. 1–2. ACM Press, 2017. doi: 10.1145/3132818.3132821

[30] B. Matthews, B. H. Thomas, S. Von Itzstein, and R. Smith. Adaptive Reset Techniques for Haptic Retargeted Interaction. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2021. Conference Name: IEEE Transactions on Visualization and Computer Graphics. doi: 10.1109/TVCG.2021.3120410

[31] B. J. Matthews and R. T. Smith. Head Gaze Target Selection for Redirected Interaction. In *SIGGRAPH Asia 2019 XR*, SA '19, pp. 13–14. Association for Computing Machinery, New York, NY, USA, 2019. event-place: Brisbane, QLD, Australia. doi: 10.1145/3355355.3361883

[32] B. J. Matthews, B. H. Thomas, S. Von Itzstein, and R. T. Smith. Remapped Physical-Virtual Interfaces with Bimanual Haptic Retargeting. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 19–27, Mar. 2019. ISSN: 2642-5254. doi: 10.1109/VR.2019.8797974

[33] M. Montanari, N. Petrinic, and E. Barbieri. Improving the GJK Algorithm for Faster and More Reliable Distance Queries Between Convex

Objects. *ACM Transactions on Graphics*, 36(3):30:1–30:17, June 2017. doi: 10.1145/3083724

[34] R. A. Montano Murillo, S. Subramanian, and D. Martinez Plasencia. Erg-O: Ergonomic Optimization of Immersive Virtual Environments. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology - UIST '17*, pp. 759–771. ACM Press, 2017. doi: 10.1145/3126594.3126605

[35] K. L. Palmerius, D. Johansson, G. Höst, and K. Schönborn. An Analysis of the Influence of a Pseudo-haptic Cue on the Haptic Perception of Weight. In M. Auvray and C. Duriez, eds., *Haptics: Neuroscience, Devices, Modeling, and Applications*, Lecture Notes in Computer Science, pp. 117–125. Springer, Berlin, Heidelberg, 2014. doi: 10.1007/978-3-662-44193-0_16

[36] Pheasant, Stephen and Haslegrave, Christine M. *Bodyspace : Anthropometry, Ergonomics and the Design of Work, Third Edition*, vol. Third edition. CRC Press, Boca Raton, 2006.

[37] M. S. M. Y. Sait, S. P. Sargunam, D. T. Han, and E. D. Ragan. Physical hand interaction for controlling multiple virtual objects in virtual reality. In *Proceedings of the 3rd International Workshop on Interactive and Spatial Computing - IWISC '18*, pp. 64–74. ACM Press, 2018. doi: 10.1145/3191801.3191814

[38] D. Schneider, V. Biener, A. Otte, T. Gesslein, P. Gagel, C. Campos, K. Čopič Pucihar, M. Kljun, E. Ofek, M. Pahud, P. O. Kristensson, and J. Grubert. Accuracy Evaluation of Touch Tasks in Commodity Virtual and Augmented Reality Head-Mounted Displays. In *Symposium on Spatial User Interaction*, SUI '21, pp. 1–11. Association for Computing Machinery, New York, NY, USA, Nov. 2021. doi: 10.1145/3485279.3485283

[39] E. Shellshear and R. Ytterlid. Fast Distance Queries for Triangles, Lines, and Points using SSE Instructions. *Journal of Computer Graphics Techniques (JCGT)*, 3(4):86–110, Dec. 2014.

[40] B. Smith, C. Wu, H. Wen, P. Peluse, Y. Sheikh, J. K. Hodgins, and T. Shiratori. Constraining dense hand surface tracking with elasticity. *ACM Transactions on Graphics*, 39(6):219:1–219:14, Nov. 2020. doi: 10.1145/3414685.3417768

[41] M. Speicher, J. Ehrlich, V. Gentile, D. Degraen, S. Sorce, and A. Krüger. Pseudo-haptic Controls for Mid-air Finger-based Menu Interaction. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, pp. 1–6. Association for Computing Machinery, New York, NY, USA, May 2019. doi: 10.1145/3290607.3312927

[42] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, and P. Volino. Collision Detection for Deformable Objects. *Computer Graphics Forum*, 24(1):61–81, 2005. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2005.00829.x. doi: 10.1111/j.1467-8659.2005.00829.x

[43] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler. Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors*, 13(5):6380–6393, May 2013. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute. doi: 10.3390/s130506380

[44] S. i. Yabe, H. Kishino, T. Kimura, and T. Nojima. Pseudo-haptic feedback on softness induced by squeezing action. In *2017 IEEE World Haptics Conference (WHC)*, pp. 557–562, 2017. doi: 10.1109/WHC.2017.7989962

[45] A. Zenner, K. P. Heqitz, and A. Kruger. Blink-Suppressed Hand Redirection. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pp. 75–84. IEEE, Lisboa, Portugal, Mar. 2021. doi: 10.1109/VR50410.2021.00028

[46] A. Zenner, H. M. Kriegler, and A. Krüger. HaRT - The Virtual Reality Hand Redirection Toolkit. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, number 387, pp. 1–7. Association for Computing Machinery, New York, NY, USA, May 2021.

[47] A. Zenner and A. Krüger. Estimating Detection Thresholds for Desktop-Scale Hand Redirection in Virtual Reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 47–55, Mar. 2019. ISSN: 2642-5254. doi: 10.1109/VR.2019.8798143

[48] A. Zenner and A. Krüger. Shifting & Warping: A Case for the Combined Use of Dynamic Passive Haptics and Haptic Retargeting in VR. In *Adjunct Publication of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20 Adjunct, pp. 1–3. Association for Computing Machinery, New York, NY, USA, Oct. 2020. doi: 10.1145/3379350.3416166

[49] A. Zenner, K. Ullmann, and A. Krüger. Combining Dynamic Passive Haptics and Haptic Retargeting for Enhanced Haptic Feedback in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2627–2637, May 2021. Conference Name: IEEE Transactions on Visualization and Computer Graphics. doi: 10.1109/TVCG.2021.3067777

[50] Y. Zhao and S. Follmer. A Functional Optimization Based Approach for Continuous 3D Retargeted Touch of Arbitrary, Complex Boundaries in Haptic Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pp. 1–12. ACM Press, 2018. doi: 10.1145/3173574.3174118