Proxy Importance based Haptic Retargeting with Multiple Props in VR

Ziming Liu, Jian Wu, Lili Wang, Xiangyu Li, Sio Kei Im

Abstract—In virtual reality applications, in addition to visual feedback, real objects can be used as props for virtual objects to provide passive haptic feedback, which greatly enhances user immersion. Usually, real object props are not one-to-one correspondence with virtual objects. Haptic retargeting technique is proposed to establish the virtual-real correspondence by introducing an offset between the virtual hand and the real hand. Sometimes, the offset is too large to cause user discomfort, and it is necessary to introduce a reset between two haptic retargeting operations to force the virtual hand and the real hand to coincide in order to eliminate the offset. However, too many resets can interfere with this immersion. To address this problem, we propose a haptic retargeting method based on proxy importance calculation using multiple props in virtual reality. The concept of proxy importance for props is introduced first, and then a proxy importance based prop selection and placement method for moving virtual objects are proposed. We also improve the performance of our method by using the props' weighted proxy importance strategy for multi-user collaboration. Compared to the state-of-the-art methods, our method significantly reduces the number of resets, the task completion time, hand movement distances, and task load without the cost of cybersickness in the single-user task. In the multi-user collaborative task, our method also achieves significant improvement using the strategy that weights the proxy importance of the props.

Index Terms—Haptic retargeting, hand redirection, perception, reset techniques, virtual reality

1 Introduction

Virtual reality (VR) generates multi-sensory stimuli through computer simulation to act on the user. At the same time, it accepts the user's feedback, thus realizing the user's interaction with the virtual world. With advancements in rendering technology, the visual experience presented to users through head-mounted displays (HMDs) has become increasingly realistic, enhancing the sense of immersion. However, when users attempt to touch virtual objects without corresponding physical objects providing haptic feedback, they may experience a sense of passing through the virtual objects, which disrupts the sense of immersion. Therefore, haptic feedback is crucial for enhancing user immersion as it complements the visual information received. However, the virtual world is dynamic and complex, and it is often challenging to replicate the same number and distribution of physical objects in the real world. As a result, a technique called haptic retargeting (HR) has been proposed to address the mismatch between virtual and physical objects [1]–[3].

Haptic retargeting methods utilize the dominance of vision in human perception to provide haptics to virtual objects by redirecting the user's real hand to the corresponding real object. This is achieved by employing mapping between the virtual and physical spaces. The previous haptic retargeting methods can be divided into two categories. The methods in the first category focus on static virtual scenes. Some methods have only one physical prop, and they redirect the hand of the user to this physical prop when the user touches the virtual object in the virtual scene [1]. Other methods have multiple physical props, and they select one of the props to represent the virtual object the user wants to touch with consideration of the distributions or shapes of physical props and virtual objects [2], [4]. The method in the second category focuses on dynamic virtual scenes [5]. It allows virtual objects to be moved to other locations in the virtual world, so choosing the prop to represent the virtual object and determining the physical location to place the prop is necessary. The limitation of these studies is that only simple rules are used to guide the prop selection and placement, for example, using the nearest physical prop to represent the virtual object and placing the prop at the location that minimizes the sum of the distances to all virtual objects. Most of the existing haptic retargeting methods use the touch reset method, which requires the user to touch a location to force the virtual hand to coincide with the real hand in order to remove the offset between the virtual hand and the real hand.

Since the above methods do not fully utilize the distribution relationship between physical props and virtual objects in the scene, there is some irrationality in the selection of physical props and placement locations, leading to a larger number of hand positions that need reset during user interaction, making the interaction inefficient and the user feeling fatigued. For example, when selecting physical props for the selected virtual object $V1$, the physical props $R1$ and $R2$ to $V1$ are the closest

- *Ziming Liu, Jian Wu, Lili Wang, and Xiangyu Li are with State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, 100191 and Peng Cheng Laboratory, Shenzhen, Guangdong, China, 518000.*
- *Sio Kei Im is with Macao Polytechnic University, Macao, China, 999078.*
- *Lili Wang is the corresponding author: wanglily@buaa.edu.cn*
- *Ziming Liu and Jian Wu are co-first authors for this paper.*

Manuscript received April 00, 0000; revised August 00, 0000.

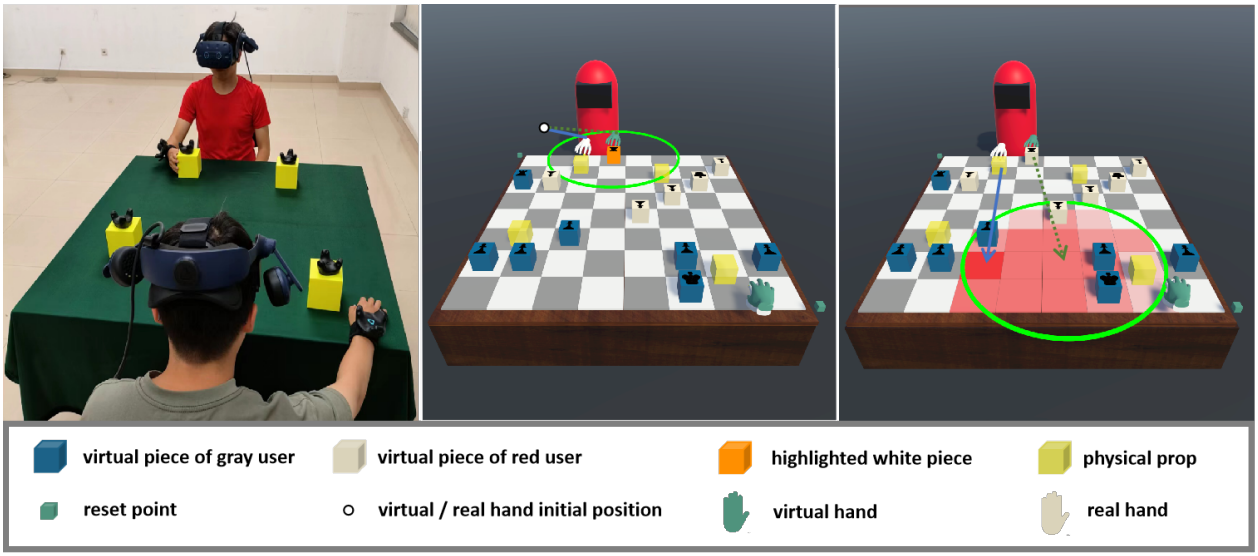

Fig. 1. With our method, two users play chess in VR using four physical props. The left image shows the real-world scene. The middle and right images show the VR views of the gray user, while the red user is selecting and placing virtual pieces.

and the next closest, yet $R1$ is also closest to another virtual object $V2$, then it may be better to select R2 as a proxy for $V1$ because when the user selects $V2$ in subsequent interactions $R1$ may be selected, which may prevent the user from resetting.

In this paper, we propose a proxy importance based haptic retargeting method (PIHR) with multiple props to address the above problem. We introduce the concept of proxy importance and propose a method to compute the proxy importance. Then, we propose a proxy importance based prop selection and placement method for moving virtual objects in virtual scenes. For multi-user collaboration, we propose the weighted proxy importance strategy to further reduce the number of resets. Figure 1 shows two users playing chess in VR with our method. The image on the left is a photo of the real scene. Two users are wearing VR HMDs on their heads with trackers attached to their right wrists. Four yellow cubes with trackers are physical props. The middle and right images show VR views of the gray user as the red user selects and places virtual pieces. For better illustration, we visualized the four props in yellow in these images, and two users cannot see them. After the red user selects the highlighted orange chess piece (middle), virtual and real hands are redirected from the initial position to the orange virtual piece and the yellow physical prop. The red user then places the selected virtual piece at the position of the green arrow (right). At the same time, his real hand is redirected to the position of the blue arrow, as this position has the highest proxy importance, indicated by the solid red.

We design two user studies to evaluate the performance of our method. The first user study uses our PIHR with a single user. Compared with the state-of-the-art methods, our method significantly reduces the number of resets, task completion time, and hand movement distances. Our method also significantly reduces task load without the cost of cybersickness. The second user study uses our method with the weighted proxy importance strategy for collaborative interaction between two users. The results show that using weighted proxy importance can further significantly reduce the number of resets, task completion time, and average hand movement distance.

In summary, the contributions of our work are as follows: 1) A completely new concept, proxy importance, and its computation method; 2) A framework for proxy importance based haptic retargeting, including three steps: new scene parameters estimation, haptic retargeting with proxy importance based selection of physical props, haptic retargeting with proxy importance based selection of placement locations; 3) A weighted proxy importance strategy to improve the performance of multi-user collaboration; 4) User studies to evaluate the performance of our method in virtual environments.

2 Related Work

In this section, we discuss the previous research on passive haptics, haptic retargeting, perception and reset techniques in haptic retargeting, which are highly related to our method.

2.1 Passive Haptics

Haptics in virtual reality can be divided into two main categories: active and passive haptics. Active haptics uses special devices with small motors to provide vibrational tactile feedback in response to touching virtual objects in the scenes [6]–[8]. Passive haptics is an interaction method in virtual reality introduced by Insko et al. [9]. It employs physical objects to convey feedback to the users via their shapes, thereby giving a tangible dimension to their virtual counterparts. Passive haptics involves integrating low-fidelity physical objects into a high-fidelity visual virtual environment, which significantly enhances the sense of presence and improves spatial knowledge training transfer. Kohli et al. [10] used the user's own non-dominant hand as a physical object to provide haptic feedback for virtual control panel interactions. Some researchers focused on the embodiment and avatar of the user's body and hands when the user touched the object in the virtual scenes. Bovet et al. [11]

investigated the effect of limb self-contact on users' judgment of whether the visual feedback was the same as the real situation when they touched a virtual object. Elbehery et al. [12] used NoHands, RigidHands, and RigidHands+ to represent the user's hands, and investigated whether the use of simple hand avatars would have some effects on user perception when the user touched virtual objects in the scene. Some researchers focused on the design of hardware systems and devices for passive haptics. Nakamura et al. [13] proposed that electrostatic haptic feedback could be added to visual displays to provide users with multi-finger haptic feedback and developed a prototype system. Achibet et al. [14] designed 'FlexiFingers', a novel multi-finger device that constrained each finger individually to produce elastic force feedback. In this paper, we also explore the method in the context of passive haptics.

2.2 Haptic Retargeting

In some passive haptic applications, there may not be a one-to-one mapping between physical props and virtual objects. Researchers have proposed haptic retargeting techniques to provide passive haptics to all virtual objects.

As early as 1933, Gibson et al. [15] pointed out that when visual and haptic information conflicted, observers relied more on information obtained through vision than on information obtained through haptic. Burns et al. [16] studied the differences in users' sensitivity to two types of visual feedback, hand proximity to virtual objects and hand penetration into virtual objects when they touched a virtual object, and the users were more sensitive to the latter case than the former one. After this, they proposed the management of avatar conflict by employment of a technique hybrid method [17], which reduced the difference between the actual position of the hand and the position of the visual feedback as soon as possible, and during the touch process, the user could not feel this difference.

Based on the idea that vision tends to dominate when sensory conflict occurs, and on the basis of the work described above, a great deal of research on redirected touching and haptic retargeting has been conducted by researchers in the last decade. Redirected touching was introduced by Kohli et al. [18] in 2012. Redirected touching warped virtual space according to the real world and mapped many virtual objects onto a real object, using that real object as a prop to provide passive haptics for the virtual objects. However, this method was limited to single-finger touch and only supports rotational angle offsets. To reduce the limitations of Kohli's method, Azmandian et al. [1] introduced haptic retargeting, which dynamically mapped physical and virtual objects, and three warping techniques: body warping, world warping, and hybrid warping. The difference between these three techniques was that offsets were introduced for virtual bodies, virtual worlds, or a combination of both.

Next, several researchers have investigated the offset between the user's virtual and real hands. Montano et al. [19] proposed the Erg-O technique, which defined two spatial partition trees and optimized the redirection process to enable more comfortable user interaction. Han et al. [20] evaluated two physical-virtual remapping methods for haptic retargeting, static offset and dynamic interpolation. The results showed that the former is more robust for larger physical-virtual mismatch cases than the latter. In order to provide more fine-grained passive haptic feedback, some other researchers have developed special devices for haptic retargeting and conducted related experiments. Cheng et al. [2] designed a hemispherical physical prop that was capable of providing haptic proxies on the surface of virtual objects at different angles for different types of virtual scenes through haptic retargeting. Zenner [21] designed a new dynamic passive haptic feedback device, which could change the weight distribution and combined haptic retargeting method to enhance the effect of weight-shifting.

However, previous haptic retargeting methods are limited to static scenes with only one or more props, or to dynamic scenes with high reset rates due to the lack of collaboration between physical props during the retargeting. Our method is a haptic retargeting method for dynamic scenes with multiple props. It always considers the distribution of all virtual objects and physical props in the process of retargeting, which reduces the number of resets and improves the interaction efficiency.

2.3 Perception

When users use a system that employs a haptic retargeting method, the magnitude of the redirection gain has a significant impact on their experience, i.e., it determines whether or not the user can perceive the difference between a real hand and a virtual hand [22]–[25]. A smaller redirection gain threshold makes this difference imperceptible to the user, which leads to a better user experience, while a larger redirection gain threshold provides a larger range of virtual hand manipulation. There are two types of redirection gains: angular gain and translation gain. Angular gain refers to the angle between the direction of movement of the virtual hand and the direction of movement of the real hand, and translation gain refers to the ratio of the distance traveled by the virtual hand to the real hand.

Zenner et al. [22] found that when haptic and visual cues were small in terms of angular deviation and translation difference, i.e., $angular\ gain < 4.5^\circ$, $0.88 < translation\ gain < 1.07$, users were unable to discriminate the difference between the real hand and the virtual hand, and were unable to perceive the retargeting. Clarence et al. [23] investigated the haptic retargeting limits that humans can tolerate when conducting haptic retargeting in different directions. The researchers conducted experiments with offsets of up to 30°, and they identified an overall haptic retargeting limit, finding that a physical proxy can be remapped to virtual objects up to 16.14° away. Cheng et al. [2] found that the maximum angle of haptic retargeting that the user could perceive but still find acceptable was 40°. On the other hand, Ogawa et al. [24] focused on the relationship between user embodiment and detection redirection threshold. They found that more realistic user embodiments can increase the value of the detectable redirection threshold by nearly 1/3 compared to less realistic user embodiments. Benda et al. [25] found that users could perceive retargeting with different translation gains for different directions. The thresholds were from 7.8 cm to 13.4 cm in different directions, i.e. the user could not perceive the offset between the real hand and the virtual hand if the offset of the retargeting was smaller than these corresponding thresholds in each direction.

Our method also considers the potential gains during haptic retargeting. In our user study, we used three groups of thresholds for the gains according to previous work [2], [22], [23].

2.4 Reset Techniques

When successive hand retargeting is performed, the cumulative offset between the virtual and real hands becomes larger. Therefore, when the user needs to perform successive retargeting, especially when multiple virtual objects that are continuously moved share the same physical prop, we have to insert a reset step between the multiple interactions to prevent the offsets from accumulating above a threshold value. After the user moves the virtual hand to a reset point or a reset area upon receiving a prompt that a reset needs to be performed, the virtual hand and the real hand are aligned here [3].

There is also some current research on making the reset interaction easier or reducing the number of resets. Zhao et al. [26] proposed a spatial warping method that allowed users to touch complex shapes without introducing reset. It also allowed continuous retargeting between two virtual objects that did not share the same physical props without resetting. Matthews et al. [3] required the users to touch a nearby physical button between redirection operations to force a reset of the virtual hand position so that the user did not need to move their hand far away to perform the reset. They [4] also investigated how predicting a user's target of choice based on gaze points could be combined with multi-physical prop mapping and on-the-fly retargeting techniques to reduce resets. Then, they proposed an adaptive method [27] to determine the reset point within the path of redirection based on factors such as the offset threshold and minimizing hand movements. The previous work used a fixed reset position, and then Matthews et al. proposed to adaptively and dynamically determine the reset position [27] so that each reset occurred at the optimal position. Yang et al. [5] proposed that when guiding the user in placing virtual objects, the physical prop could be placed in a location that was better able to be closest to the sum of the distances of multiple other virtual objects, thus reducing the reset probability.

The previous methods optimize the number of resets during the haptic retargeting process. Some only consider the static scene, and some only consider the distribution of virtual objects and the physical prop of the current step. There is still a lot of room for optimization to reduce the number of resets during haptic retargeting, and our method comprehensively considers the distribution of all virtual objects and physical props in the dynamic scene so as to reduce the number of resets further.

3 Proxy Importance Based Haptic Retargeting Method

The pipeline of our method for each interaction is shown in Figure 2, which has three steps. The first step is the scene parameters estimation. Two parameters of the scene, valid proxy distance and reset region radius, are computed according to the distribution of physical props and virtual objects. The second step is haptic retargeting with proxy importance based physical prop selection. After a virtual object that needs to be moved is chosen, we determine the prop candidates nearby with consideration of scene parameters. Then, the proxy importance of each prop candidate is calculated. The prop with minimal importance will be the proxy for the selected virtual object. We redirect the user's hand to this prop when he or she picks up the selected virtual object using the body warping based haptic retargeting method [1]. The third step is haptic retargeting with proxy importance based prop placement. The location in the virtual scene to place the picked-up virtual object is chosen. We determine the region in the physical world to place the corresponding prop according to the target location of the virtual object and the scene parameters. Candidate placement locations are generated discretely within the region. After this, we compute the proxy importance of each placement location candidate and redirect the user to place the physical prop to the location candidate with the maximum proxy importance [1]. Using our method, the user can move the virtual object continuously without having to reset the hand at each step. Resets occur only when the redirection gain exceeds the user perceivable threshold [2], [22], [23].

Our method focuses on proposing scene parameter estimation (Section 3.1); proxy importance computation (Section 3.2); and an algorithm for selecting and placing physical props (Section 3.3). Moreover, we propose a weighted proxy importance strategy. The weighted strategy is designed for multi-user cases, it computes the importance based on the ownership of different virtual objects(Section 3.4).

3.1 Scene Parameters Estimation

When selecting props and placing props using the "brute force" method, it is computationally inefficient to compare the importance of all physical props and candidate locations in the scene. In addition, considering virtual objects that are far away from the candidate physical props or candidate placement locations can negatively affect the results. To address these problems, considering only the effect of virtual objects close to physical props or physical placement locations, we propose two new scene parameters, namely the valid proxy distance and the reset region radius, for describing the scene layout, as well as the algorithms to compute them.

The valid proxy distance Given a virtual object, a physical prop can be a proxy for that object only if the distance between

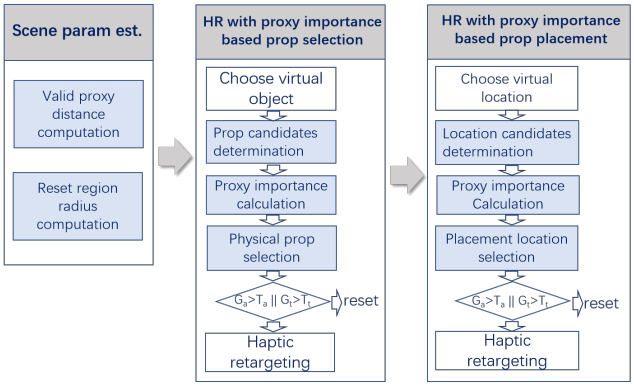

Fig. 2. Pipeline of our proxy importance based haptic retargeting method

the physical prop and the virtual object is less than the valid proxy distance. The larger the valid proxy distance is, the more virtual objects a physical proxy can proxy, i.e., the more physical proxy candidates can represent the same virtual object, which results in high computational overhead. In order to reduce the computational overhead, and at the same time make as many virtual objects as possible have corresponding physical props, we set a threshold $Ratio_D$ to regulate the balance between the computational overhead and the proxy coverage rate of the described virtual objects (set to 80% in the implementation), and use binary search to find the minimum valid proxy distance D that satisfies this threshold (Algorithm 1).

The inputs of the algorithm are the physical prop list $PList$ and the virtual object list $VList$. The output of the algorithm is the minimized valid proxy distance D. The algorithm has three main steps. The first step is to determine the nearest and farthest distances between all virtual objects and physical props, which initializes the range of the binary search (lines 1-2); the second step is to compute the coverage of the virtual objects being proxied by using the median of the search range as the valid proxy distance (lines 3-13); the third step is to compare this coverage with $RATIO_D$, and update the search range based on the result of the comparison(lines 14-19). Steps 2 and 3 are performed iteratively until the range interval becomes small. If none of the coverage of the virtual objects being proxied using the median in the above iterations satisfies the threshold, we set D to two times the final median (lines 20-23).

Algorithm 1 valid proxy distance computation

Input: physical prop list $PList$ and virtual object list $VList$
Output: valid proxy distance D

```
        // Step 1 binary search range initialization
 1: end = GetMaxDistance(PList, VList)
 2: start = GetMinDistance(PList, VList)
        // Step 2 proxy coverage computation
 3: while (end - start) > ε do
 4:     mid = (start + end) / 2
 5:     score = 0
 6:     for p in PList do
 7:         for v in VList do
 8:             d = Distance(p, v)
 9:             if d < mid then
10:                 score += 1
11:             end if
12:         end for
13:     end for
        // Step 3 comparison and range updating
14:     if score > RATIO_D * VList.num then
15:         end = mid
16:     else
17:         start = mid
18:     end if
19: end while
20: if mid - start < ε then
21:     mid = 2 * mid
22: end if
23: D = mid
```

Figure 3 left provides an example of the minimized valid proxy distance. The virtual scene has nine green virtual objects $V1$-$V9$, and two blue physical props $R1$ and $R2$. The valid proxy regions are circled by light blue circles centered on each prop with a radius of D. Within the valid proxy region of $R1$, four green virtual objects $V1$,$V2$, $V3$, and $V9$ can take $R1$ as their proxy. Similarly, within the valid proxy region of $R2$, four green virtual objects $V2$, $V6$, $V7$, and $V9$ can take $R2$ as their proxy. As a result, eight proxy relationships are created, and there are a total of nine virtual objects in the scene, with a relationship to virtual object ratio of 88.9%, which is greater than the set ratio threshold $RATIO_D$= 80%.

Reset region radius When placing physical props, it is necessary to consider that the number of times a reset occurs at the next pickup of a virtual object and its prop pair is as small as possible. In order to evaluate the placement location, we introduce the reset region radius. We take the midpoint of virtual objects and physical props as the center, and using this radius, we can construct a circular reset region to test whether the virtual objects and physical props pairs whose midpoints are located in this region need to be reset. The larger the radius of the reset region, the more virtual objects and physical props are located in the region, and the higher the computational overhead. we set a threshold $RATIO_R$ to regulate the balance between the computational overhead and the accuracy of whether a reset actually occurs (set to 50% in the implementation), and use binary search to find the minimum reset region radius R that satisfies this threshold (Algorithm 2).

The input is valid proxy distance D. The output is the minimized reset region radius R. The algorithm has three main steps. The first step is to initialize the range of the binary search(lines 1-2); the second step is to calculate the ratio of the sampling points on the circumference of the current reset region that needs to be reset(lines 3-10); the third step is to compare this ratio with $RATIO_R$, and update the search range based on the result of the comparison(lines 11-16). Steps 2 and 3 are performed iteratively until the range interval becomes small. CheckGain(p0, v0, p1, v1) is a function to check whether the redirection gain of an interaction from p0, v0 to p1, v1 exceeds the predefined threshold.

Figure 3 right shows an example of the minimized reset region radius. One green virtual object $V1$ is proxied by a blue prop $R1$. A light blue circle is drawn with the valid proxy distance as the radius D, centered around $R1$. $V1$ is on

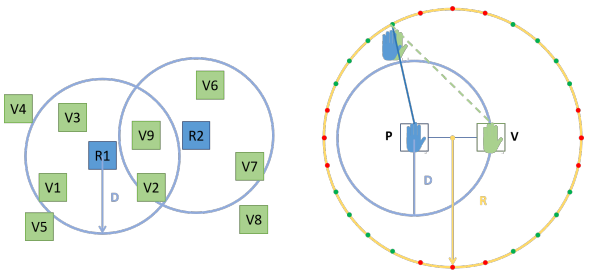

Fig. 3. Scene parameters estimation. Valid proxy distance D is computed according to the distribution of the physical props and virtual objects in the scene (left). Reset region radius R is estimated by using the positions of users' virtual and real hands (right).

Algorithm 2 reset region radius computation

Input: valid proxy distance D
Output: reset region radius R

```
        // Step 1 binary search range initialization
 1: p0 = (-1 * D / 2, 0)  v0 = (D / 2, 0)
 2: start = 0  end = 10
        // Step 2 reset ratio computation
 3: while (end - start) > ε do
 4:     mid = (start + end) / 2
 5:     score = 0
 6:     for θ = 0; θ < 360; θ += 5 do
 7:         p1 = v1 = (mid * Sinθ, mid * Cosθ)
 8:         if CheckGain(p0, v0, p1, v1) then score += 1
 9:         end if
10:     end for
        // Step 3 comparison and range updating
11:     if score > RATIO_R * 72 then
12:         end = mid
13:     else
14:         start = mid
15:     end if
16: end while
17: R = mid
```

this circle, which means that $V1$ is one of the farthest virtual objects that $R1$ can proxy. We take the midpoint between $R1$ and $V1$ as the center and construct a yellow circle to indicate the high possibility of the reset region of $R1$. We sample point c uniformly on the circle. The user's real hand (blue) is at position $R1$, and the virtual hand (green) is at position $V1$ in the beginning, and when the user places a prop, the real hand and the virtual hand need to reach position c. We calculate the angle and translation gains during both redirections, and if any gains exceed the threshold, we consider point c to be the point that needs to be reset and mark it with red; otherwise, point c that does not need to be reset is marked in green. We expect to find the minimum radius so that the ratio of sampling points that do not need reset exceeds $RATIO_R$, which also is 50%. There are 28 points along the circle, 16 of them are green, and the ratio of points that do not need to be reset exceeds $RATIO_R$.

3.2 Proxy Importance

3.2.1 Concept definition

Proxy importance is an attribute of the physical prop and placement location that indicates the score obtained by using a prop or placing a prop at that location to proxy for the virtual object in order to minimize the number of resets when performing subsequent interactions. The larger the value, the higher the importance, and vice versa.

The computation of proxy importance involves three factors: proxy closeness PC, reset avoidance possibility PR, and punishment PN. These three factors cover the characterization of proxy importance over time. Proxy closeness PC represents the proximity of a physical prop to multiple virtual objects in the virtual scene that it can proxy for. A larger value of PC indicates a reasonable distribution of all physical props and virtual objects in the scene, and maintaining a reasonable distribution helps to reduce the expectation of reset occurrence in the long run. Reset avoidance possibility PR indicates the probability that the user will continue the next interaction without a reset. Punishment PN represents the penalty of importance for the redirection gain of the current interaction. The higher the gain, the higher the penalty. If the redirection gain of the current interaction exceeds the threshold, an additional penalty is added.

3.2.2 Computation of the proxy importance

The proxy importance I can be calculated by weighting and summing the above three factors using Algorithm 3. The inputs of the algorithm are the position of prop or placement location P, selected virtual object or the placement location of the selected virtual object V, virtual object list $VList$, physical prop list $PList$, real hand start point P_0, virtual hand start point V_0, the output of the algorithm is proxy importance I

Algorithm 3 Proxy Importance Calculation

Input: Position: P, selected virtual object: V, virtual object list: $VList$, physical prop list: $PList$, real hand start point: P_0, virtual hand start point: V_0
Output: The proxy importance I

```
 1: PC, PR, PN = 0
        // Step 1 Proxy closeness computation
 2: VNUM = 0
 3: for each V' in VList except V do
 4:     d = Distance(P,V')
 5:     if validationCheck(d,D) then
 6:         r = GetRank(d,PList,V')
 7:         VNUM += 1
 8:         PC -= d * Sqrt(r)
 9:     end if
10: end for
11: PC /= VNUM^1.5
        // Step 2 Reset avoidance possibility computation
12: ValidPairNum = 0
13: for each P' in PList do
14:     for each V' in VList do
15:         if ResetCheck(P,V,P',V',R) then
16:             ValidPairNum++
17:             if CheckGain(P,V,P',V') then
18:                 PR++
19:             end if
20:         end if
21:     end for
22: end for
23: PR /= ValidPairNum
        // Step 3 Punishment computation
24: p = GainCheck(P_0,V_0,P,V)
25: PN = GetGain(P_0,V_0,P,V) + p
26: I = a * PC + b * PR + c * PN
```

We initialize PC, PR, and PN as 0 (line 1). PC is calculated with consideration of the distribution of the virtual objects and physical props in the current scene (lines 2-10). $VNUM$ keeps the number of virtual objects within a valid proxy region centered by P, and it is initialized to 0 (line 2). For each object V' in virtual object list $VList$, we calculate the distance between V' and P (lines 3-4) and check whether V' is within the valid

proxy region (line 5). If it is true, we sort each physical prop in *PList* according to the distance to reach V' from smallest to largest and determine the rank r of P (line 6). The value of $VNUM$ will be added by one (line 7). Since the distance d is inversely proportional to PC, we accumulate the product of $-d$ and the square root of r into PC (line 8). Finally PC will be normalized with $VNUM^{1.5}$ (line 11).

PR is calculated according to the probability of a reset occurring in the next interaction (lines 12-22). For each P' in *PList* and each V' in *VList* (lines 13-14), we generate the reset region centered on the midpoint of P and V, and check whether the midpoint of P' and V' are within the reset region (line 15). If it is true, the number of valid pair will be added by one (line 16). Then we check if a reset is needed when we redirect the real hand from P to P' as the virtual hand from V to V' in the next step (line 17). If it is true, PR will be added by one (line 18). After this, we normalize PR with *ValidPairNum* (line 23).

PN is calculated with two terms (lines 24-26). First, we check if a reset is needed when we redirect the real hand from P_0 to P as the virtual hand from V_0 to V in the current step. If it is true, the value of p will be 1; otherwise, the value of p will be 0 (line 24). Then we set PN as the summation of the redirection gain during current redirection and p (line 25). Finally, we weight and accumulate PC, PR and PN to obtain I (line 26). We consider these three factors to have an equal impact on importance, so we set a, b, and c to 2, 5, and ± 1 in our implementation, according to the magnitude of the three-factor values. The penalty weight c is reversed for the physical prop and the placement location.

Figure 4 shows an example of the influence of proxy closeness PC on physical prop proxy importance I. $V1$ is the selected virtual object, and we construct the valid proxy region of $V1$ with the valid proxy distance D of the scene (blue circle). Thus $R1$ and $R2$ are the valid props of $V1$. In the left image, $R1$ can be the proxy for $V1$ and $V4$, $R2$ can be the proxy for $V1$, $V2$ and $V3$, and proxy closeness of $R2$ is larger than $R1$'s, thus the proxy importance of $R2$ is higher than $R1$ due to proxy closeness. In the right image, $R1$ has higher proxy importance than $R2$ because another prop $R3$ is placed in the scene. $R3$ can be the proxy for $V2$ and $V3$ and is closer to $V2$ and $V3$ than $R2$, which reduces proxy closeness of $R2$.

Figure 5 shows an example of the influence of the reset avoidance probability with the next step on the proxy importance of the physical prop placement location. $R1$ is the prop of $V1$, $V1'$ is the placement location of $V1$ in the current step, and $L1$ and $L2$ are the candidate placement location of $R1$. In the left image, $L2$ is closer to $V1'$ than $L1$, so $L2$ has higher importance due to the higher proxy closeness PC. In the right image, to compute PR, we create two high-possibility reset regions (marked with orange circles), centered at the mid-points of $L1V1'$ and $L2V1'$ with the radius of R. Our interaction process is continuous and does not require the user's hand position to be reset at each step of the interaction. Hence, after positioning $V1$ to $V1'$ with prop $R1$, the user proceeds to the next step. This step typically involves a direct movement of the hand from the location of $L1$ or $L2$ to the position of $R2$, while $R2$ will serve as the proxy for $V2$. Because the relative relationship between $L1$ and $V1'$ is similar to that between $R2$ and $V2$, the redirection gain from $L1$ and $V1$ to reach $R2$ and $V2$ is smaller, and the probability of a reset occurring is also smaller. In contrast, the relative relationship between $L2$ and $V1'$ is very different from that between $R2$ and $V2$, the probability of a reset occurring is large, thus PR for $L2$ is smaller than $L1$'s, and a larger PR for $L1$ increases the proxy importance for $L1$.

3.3 Prop Selection and Placement

The scene proxy importance is the sum of the proxy importance of all physical props under the current layout, and it is inversely proportional to the number of resets for subsequent processes. The higher its value, the fewer resets are required for subsequent processes, and the lower it is, the more resets are required for subsequent processes. When picking up virtual objects, picking up the physical prop with the lowest proxy importance as the virtual object's prop will keep the scene proxy importance maximized, thus reducing the number of resets for subsequent processes. When placing virtual objects, placing the prop at the location with the highest importance also maximizes the scene proxy importance and reduces the number of resets for subsequent processes, and vice versa.

Prop Selection There are two main steps to selecting a prop to be the proxy for the chosen virtual object. The first step is to generate prop candidates. We set all physical props within the valid proxy distance from the virtual objects as the

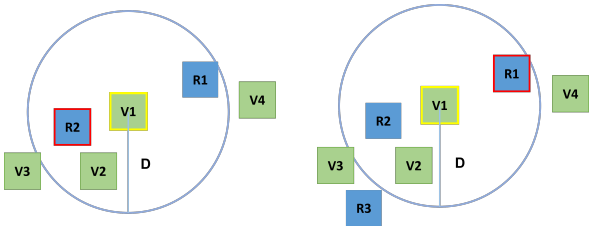

Fig. 4. The influence of proxy closeness on physical prop proxy importance of the physical prop. The left image shows $R2$ has higher proxy importance due to proxy closeness. The right image shows $R1$ becomes the prop with higher proxy importance because the extra prop $R3$ is placed in the scene.

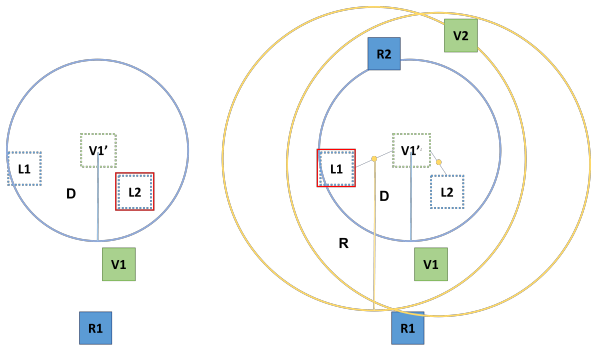

Fig. 5. The influence of the reset avoidance probability on physical prop proxy importance of placement location. The left image shows $L2$ has higher proxy importance due to the probability of reset. The right image shows $L1$ becomes the placement location with higher proxy importance because the extra prop $R2$ and virtual object $V2$ are placed in the scene.

prop candidate. For example, in the left of Figure 6, $R1$-$R3$ are the prop candidates for $V1$, and $R4$ is not. The second step is to select a prop to be the proxy for the chosen virtual object. We calculate the importance of each prop candidate with three factors: proxy closeness PC, reset avoidance possibility PR, and punishment PN. In this step, we set the value of PR to 0 because in the next step, after the user picks up the prop, the number of possible locations where the virtual object and the prop could be placed is too large, and the number of their combinations is even larger, so the prediction fails. The penalty weight c of PN is set to 1 because candidate props that would cause large redirection gains in the current interaction are not good options. After calculation, the candidate with the lowest importance value is selected as the prop of the selected virtual object to maximize the scene proxy importance.

Placement Location Selection There are also two steps to selecting a placement location. The first step is to generate placement location candidates. We create a circular area centered on the virtual object placement location V' and radiused by the valid proxy distance D. A uniform grid with a fixed cell size is generated within this region, and we use the center of each cell (blue dots in Figure 6 right) as a candidate location. In our implementation, the side length of the object prop is 10 cm, and the side length of the cell is set to 3 cm, which is about 30% of the side length of the physical prop. The second step is to select a placement location to place the prop. We compute proxy importance values for each placement location candidate. In the computation, the penalty weight c of punishment PN is set to -1 because candidate placement locations that would cause large redirection gains in the current interaction are not good options. In contrast to selecting physical props, we select the candidate with the highest importance value when selecting physical prop placement locations. Prop placement may also present a problem in that the current placement position may collide with existing physical props in the scene. To solve this problem, we check each placement location candidate to make sure it will not collide with existing physical props. If the candidate position's cell overlaps with any physical prop, then the placement location candidate within that cell is canceled. As shown in Figure 6 right, the candidate position in the cell where R is located is canceled.

3.4 Multiple User Haptic Retargeting

In this section, three proxy importance strategies: 1) **Equivalent Strategy**, 2) **Ownership Strategy**, 3) **Weighted Strategy**, are discussed to further reduce the reset numbers in the context of multiple user collaborative interactions.

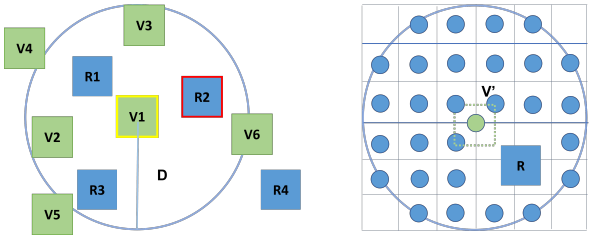

Fig. 6. Candidate props and candidate placement locations. $R1$, $R2$ and $R3$ are the candidate props of $V1$ (left). The candidate placement locations are shown with blue dots around the target placement location of the virtual object V' (right).

We categorize multiple user collaborative interactions into two cases: without virtual object ownership and with virtual object ownership. In the first case, there is no ownership of virtual objects in the interaction process, and each user can interact with all props and virtual objects, e.g., two users building a house with blocks. In this case, we use **Equivalent Strategy** for all virtual objects, i.e., all virtual objects are considered equally for each proxy importance calculation.

In the second case, virtual object ownership exists in the multi-user interaction process. Each user can only interact with virtual objects and all props that belong to him/her, e.g., two users playing chess. In this case, insisting on the use of **Equivalent Strategy** affects the overall arrangement of physical props. This is because the algorithm takes into account virtual objects that are not owned by the current user, which reduces the proxy importance of virtual objects operated by the current user and may lead to more resets. Therefore, **Ownership Strategy** can be used, where only virtual objects owned by the current user and all props are considered when calculating the proxy importance of an individual user. This ensures the overall proxy capacity of props for the current user's virtual objects, but may reduce the proxy capacity of props for other user's virtual objects by completely disregarding them.

In order to solve the above problems, in the collaborative interaction with multiple users, we propose a new strategy, **Weighted Strategy**, which applies different weights to virtual objects with different ownership when performing proxy importance calculations. We set the weight of the virtual objects owned by the current user to 1. We classify the virtual objects owned by other users into two groups: the virtual object V_i within the reset region centered on the midpoint between the corresponding user's virtual hand and the real hand, and the virtual object V_t outside that region. The weight of V_i is set to w_1 ($0 < w_1 < 1$), and the weight of V_t is set to w_2 ($0 < w_2 < w_1$). This minimizes the influence of other users' virtual objects on the current user's choice of physical prop and placement locations and also takes into account the virtual objects that are closer to the other users that may interact with them.

In these cases with multiple users, we refer to the sum of the proxy closeness PC of all virtual objects of a given user as the proxy closeness of that user. We also consider the balance of proxy between the multiple users to avoid situations where one user has significantly more virtual objects that are closely proxied by physical props than the other user. To achieve this balance, we introduce an additional weighting factor w_3 ($1 < w_3$), which is used to adjust the influence of virtual objects owned by different users. When calculating proxy importance, the weight of the virtual object of the user with the lowest proxy closeness will be multiplied by w_3. The weight of the virtual object of the user with the highest proxy closeness will be multiplied by $1/w_3$.

The example in Figure 7 shows the advantages of our weighted proxy importance strategy. There are two users, User1 and User2, User1's hand position and virtual hand position are indicated using solid blue and solid green hand icons, respectively, and User2's hand position and virtual hand position

are indicated using hollow blue and hollow green hand icons, respectively. The physical props $R1-R3$ have no ownership marked with solid blue squares. Virtual objects $V1-V3$ belong to User1 and can only be moved by User1, and virtual objects $V4-V6$ belong to User2 and can only be moved by User2. This is a case of selecting the location to place the prop. The physical prop $R1$ is used to be the proxy for the virtual object $V1$ the user is currently picking up. In the next step, the user places $V1$ in the virtual space at location $V1'$, and $L1-L3$ are the placement location candidates of $R1$.

Using **Equivalent Strategy**, we consider $V2-V6$ and $R2-R3$ when calculating the importance of placement location from $L1-L3$. Although $L1$, $L2$, and $L3$ can all be the proxy for two virtual objects, $L2$ can be the proxy for $V5$ and $V6$ more closely, which means that $L2$ has the largest value of PC. Since the reset probability of the next step is close when physical props are placed at $L1$, $L2$, or $L3$, the values of PR are close. To summarize, the prop at $L2$ has the highest importance. The problem that this brings is that if in the next interaction, User2 chooses $V4$, $R1$ will again be used as a physical prop to be the proxy for $V4$, and User2 will need to perform a hand reset when moving $V4$ because the angular gain of using $R1$ to be the proxy for $V4$ will exceed the threshold value.

Using **Ownership Strategy**, when User1 interacts, we only consider $V2-V3$ and $R2-R3$ in calculating the importance of proxies from $L1-L3$, and $R1$ should be able to be placed as close as possible to $V2$ and $V3$, so $L1$ has higher proxy importance. This poses a similar problem to that of using **Equivalent Strategy**, i.e., if in the next interaction, User2 also selects $V4$, $R1$ will become a physical prop to be the proxy for $V4$, and User2 will need to perform a hand reset when he moves $V4$ because the translation gain of using $R1$ to be the proxy for $V4$ will be greater than the threshold value.

Using **Weighted Strategy**, our weighted proxy importance strategy, when User1 interacts, the weights of the virtual objects $V1$, $V2$, and $V3$ owned by User1 are set to 1. Then a reset region in orange is made with the midpoint of User2's virtual hand and real hand as the center of the circle, and the weights of the virtual objects $V4$ owned by User2 are set to w_1, $w_1 < 1$. The weights of the virtual objects $V5$ and $V6$, which are located outside the reset region, are set to w_2, $w_2 < w_1$. So the influence of $V5$ and $V6$, which are far away from User2, is weakened when considering the placement of $R1$, and thus $L3$ has a higher proxy importance. Therefore, even if User2 selects $V4$ in the next interaction, User2 no longer needs to perform a hand reset by selecting $R1$ to be the physical prop for $V4$

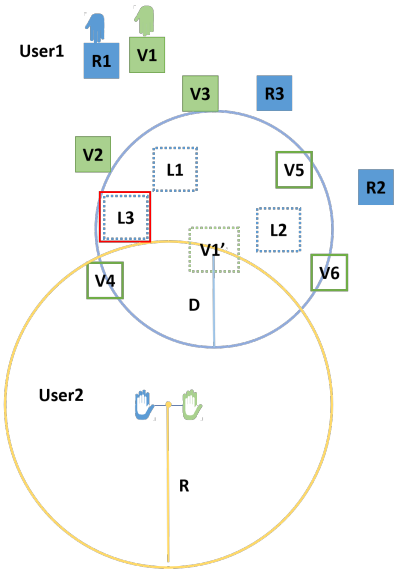

Fig. 7. Haptic retargeting of two users using our weighted strategy. $L3$ around the target location $V1'$ is selected as the placement location of $R1$, which is the selected prop of $V1$.

4 User Study 1: Single User Task

User study 1 is designed to evaluate the performance of our proxy importance based haptic retargeting method with the single user task. We conduct User Study 1 with two scenes and three groups of redirection gains containing translation and angular gains. Figure 8 shows the top views of the two scenes. Ten green virtual objects are randomly distributed in both scenes on a 2m*2m plane. In Scene U on the left, the five blue props are uniformly distributed, while in Scene C on the right, the five blue props are more concentrated. The white cube at the bottom right indicates the location of the reset point.

In order to determine the thresholds of the redirection gains, similar to the method in [27], we first conducted a hand retargeting simulation on the plane to obtain the average angular gain and translation gain. The simulation randomly generated one million start points and one million endpoints for the physical and virtual hands. The corresponding gain was computed to complete the redirection of each pair of start-to-end points. We averaged these one million pairs of angular gains and translation gains to get the average angular gain of 90° and the average translation gain of 2.5 on the plane. Our average angular gain and average translation gain were much higher than the 12.5° and 1.28 in [27] because their virtual scene is about 7.5 cm * 7.5 cm, and the touch points in the virtual scene and the touch buttons on the physical device were fixed and more centrally distributed. Previous studies have shown that users cannot perceive the redirection when the angular gain is lower than 4.5° and the translation gain is between 0.88 and 1.07 [22], and the maximum angular gain that users can tolerate is 40° [2]. All these values are clearly smaller than the average gain we obtained in our simulations. We set three levels of gain thresholds according to the previous work [2], [22], [23]: almost imperceptible ($T1$), with an angular gain threshold of 5° and a

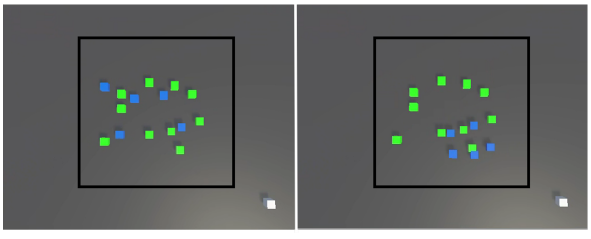

Fig. 8. The top views of the two scenes U and C. In the beginning, the blue physical props are placed randomly and uniformly in the U scene (left) and clustered in the C scene (right).

translation gain threshold of 1.1; perceptible but not annoying ($T2$), with an angular gain threshold of 15° and a translation gain threshold of 1.2; and perceptible and tolerable ($T3$), with an angular gain threshold of 30° and a translation gain threshold of 1.3. The user study had the approval by Biology and Medical Ethics Committee of Beihang University.

4.1 Study Design

Hardware and Software Implementation We used the HTC VIVE system, which consists of tracked and positioned HMDs and 6 trackers. The HMDs were connected to a computer with an Intel i7 processor, 32GB of RAM, and an RTX 3080ti graphics card. Users wore the HMDs on their heads during the experiments. The trackers are used to track the user's hand and the physical props. Our experimental platform was implemented using Unity 2021.1.28f1. The prototype of our platform was built on the HaRT toolkit [28].

Participants We recruited 24 participants for our study, including 14 males and 10 females, with ages ranging from 22 to 28 years old, the mean age of the participants is 24 years old. Among the participants, 17 of them had prior experience with immersive VR applications. All participants had normal or corrected vision, and none of them reported any visual impairments or balance disorders. All participants are right-handed, and throughout the experiment, they use their right hands, which are their dominant hands.

Conditions The study consisted of two control conditions, $CC1$ and $CC2$, and one experimental condition EC. We have compared EC to $CC1$ and $CC2$ in a within-subject controlled study. In $CC1$, the prop was selected based on the closest distance to the selected virtual object [4]. The placement position was then determined by the relative positional relationship between the virtual object and its prop. In $CC2$, the prop was also selected based on the closest distance to the selected virtual object. The placement location was then determined by minimizing the sum of the distances of the prop to all virtual objects [5]. The participants of the experimental condition EC are using our method. Each participant was exposed to each condition in random order.

Task The task requires the participant to move cubes 50 times in a virtual scene. The system will randomly generate a series of numbers indicating the cube to be moved, and the system will highlight the current cube to be moved in orange, until the required number of interactions has been completed. When a virtual cube is picked up, the system generates a random location on the virtual plane and places an orange cube there to indicate the target location of the placement. For a fair comparison, the manipulated virtual cube and the corresponding placement location indicated by the system in the virtual scene are the same under all conditions. The initial positions of the physical props and the virtual object distribution at the start of the task are also the same in all conditions.

Procedure Participants stand in front of a physical table with five real cubes with tackers as physical props. The physical cubes and table are aligned with the virtual world, and the virtual hand is aligned with the participant's real hand.

Prior to beginning the user study, the task procedure and resetting technique are explained in detail to each participant. The study begins with five random pick-and-place procedures in order to familiarize the participants with picking up and placing cubes. The process lasts approximately five minutes.

First, participants complete tasks with 2 scenes (U and C) $\times$1 Gain (T2) $\times$3 Conditions.. Then, participants complete tasks with 1 scene (U) $\times$3 Gains $\times$3 Conditions. The participants are exposed to these different scenes or thresholds in random order. In each task, the participant is required to move the virtual cubes with the physical props 50 times. Each participant has an individual reset point, and if the redirection exceeds a threshold, participants need to touch the reset point. The system will show red "Need Reset" above the table and turn the reset position to yellow to remind the participant. Otherwise, the participant can continue the interaction without interruption. Participants take 4-8 minutes to complete each task, with a 3-minute break between tasks. It takes about 2 hours for the participant to complete all tasks. The number of times the participant's hand has to be reset, the distance it has to be moved, and the total time the participant needs to complete the task are recorded automatically in the process of the tasks. Upon completion of all experiments, participants complete the NASA Task Load Questionnaire and the System Usability Scale Questionnaire after at least forty minutes of using each condition's method.

Metrics and Statistics The participants' task performance is measured using the following objective metrics: (1) Reset rate, defined as the ratio of the number of redirections involving a reset to the total number of redirections, which is our primary optimization goal. Since the task requires the participant to move the cube 50 times in the virtual scene, and each move has two interactions, picking up and placing, both of which may need to reset, a maximum of 100 resets are required. (2) Task completion time, measured in seconds, is defined as the time interval from the start of the task to the end of 100 redirections. (3) Average hand movement distance, measured in meters, is defined as the average real hand movement distance during each interaction. We also evaluate the VR experience using two subjective metrics: user task load, measured using NASA's standardized TLX questionnaire [29] [30], and usability, measured using System Usability Scale Questionnaire with six questions: intuitiveness, effectively, accuracy, natrualness, satisfaction, and easiness [31]. For each metric, the results of EC were compared to those of $CC1$ and $CC2$. First, the data were evaluated for outliers using box plots, and any outliers were removed. Then, Shapiro-Wilk test was used to assess the normality of the data distribution. Next, the assumption of sphericity was evaluated using Mauchly test [32], and if violated, Greenhouse-Geisser correction [33] was applied. Subsequently, RMANOVA [34] was conducted, followed by posthoc tests with Bonferroni-adjustment to analyze the differences. p-values from the statistical tests and Cohen's d [35] to quantify the effect sizes were reported. Cohen's d values were translated to qualitative effect size estimates of Huge ($d > 2.0$), Very Large ($2.0 > d > 1.2$), Large ($1.2 > d > 0.8$), Medium ($0.8 > d > 0.5$), Small ($0.5 > d > 0.2$), and Very Small ($0.2 > d > 0.01$).

4.2 Results & Discussion

Reset Rate Table 1 gives the reset rate for the three conditions under two scenes with redirection gain threshold $T2$. Column

three gives the average value and the standard deviation, column four gives the relative reduction of the reset rate from *CC* to *EC*, and columns five to seven give the statistical analysis of the *EC* versus *CC* differences. Statistical significance is indicated with an asterisk. *EC* gets a significantly reduced reset rate with all scenes than *CC*1 and *CC*2. When selecting objects, *CC*1 and *CC*2 both select the physical prop that is nearest to the selected virtual object. When that nearest physical prop is the only one around the other virtual objects, this can cause the other virtual objects to lose the nearby physical prop. When placing objects, *CC*1 maintains the relative positions between physical props and virtual objects without considering the distribution of other virtual objects. As a result, the overall proxy closeness of the scene may decrease, leading to an increased probability of reset. *CC*2 selects a placement position for physical props that minimizes the sum of distances to all virtual objects, typically placing objects near the center of the scene. If the virtual objects are distributed closely together, this position would have a high proxy closeness for the virtual objects. However, if some virtual objects are far away from others, the placement position might deviate from the concentrated region, resulting in a decreased proxy closeness for those objects located in the concentrated region. In subsequent interaction, this can lead to an increased probability of resets when moving those objects. In contrast, *EC* considers the distribution of all virtual objects when selecting a physical prop or placement position and excludes the influence of virtual objects that are far away from the selected virtual object or virtual object placement position. This method results in higher proxy closeness for virtual objects near the selected virtual object or virtual object placement location, reducing the probability of resets in subsequent interactions.

TABLE 1
Reset Rate (%)

Scene	Condition	Avg ± std. dev.	(CC_i-EC) / CC_i	p	Cohen's d	Effect size
U	EC	0.29 ± 0.04				
	$CC1$	0.39 ± 0.04	25.8%	$< 0.001^*$	2.5	Huge
	$CC2$	0.98 ± 0.01	70.8%	$< 0.001^*$	22.0	Huge
C	EC	0.33 ± 0.03				
	$CC1$	0.40 ± 0.03	17.7%	$< 0.001^*$	2.1	Huge
	$CC2$	0.98 ± 0.01	66.8%	$< 0.001^*$	25.9	Huge

The reset rate of *EC* for Scene U is better than that of Scene C for the following reasons. In Scene U, the initial distribution of the physical props is random and uniform, while in Scene C, the initial distribution of props is clustered. At the beginning of the interaction, the physical props in Scene C are clustered next to a part of virtual objects, and other virtual objects do not have physical props nearby. As a result, the user with Scene C tends to have a larger reset number during the initial interaction process. However, after some placements, the props in Scene C are placed in a reasonable and non-clustered manner by using our method, and then the user needs a similar number of the reset in the subsequent interactions as in Scene U.

Table 2 gives the reset rate for the three conditions with the different redirection gain thresholds of Scene U. *EC* participants completed all tasks with a significantly reduced reset rate.

Although our method achieved significant improvements under different gains, the optimization is most pronounced with $T2$, followed by $T3$, $T1$. This can be attributed to the small gains for *EC*, resulting in a small candidate range and a similar selection of physical props and placement location compared to *CC*1 and *CC*2; the large gains resulting in a large candidate range and physical props not being able to closely proxy nearby virtual objects, which may have a negative impact and lead to additional resets.

TABLE 2
Reset Rate (%)

T	Condition	Avg ± std. dev.	(CC_i-EC) / CC_i	p	Cohen's d	Effect size
$T1$	EC	0.42 ± 0.03				
	$CC1$	0.44 ± 0.02	6.0%	0.03^*	1.1	Large
	$CC2$	0.98 ± 0.01	57.5%	$< 0.001^*$	7.3	Huge
$T2$	EC	0.32 ± 0.05				
	$CC1$	0.39 ± 0.03	20.0%	$< 0.001^*$	1.9	Very Large
	$CC2$	0.98 ± 0.01	67.8%	$< 0.001^*$	19.1	Huge
$T3$	EC	0.28 ± 0.05				
	$CC1$	0.32 ± 0.04	14.0%	0.04^*	1.0	Large
	$CC2$	0.97 ± 0.01	70.4%	$< 0.001^*$	19.1	Huge

Average Hand Movement Distance Table 3 gives the average hand movement distance for the three conditions under different scenes with $T2$. Compared to *CC*1 and *CC*2, *EC* showed significantly shorter average hand movement distance in all scenes.

TABLE 3
Average Hand Movement Distance (m)

Scene	Condition	Avg ± std. dev.	(CC_i-EC) / CC_i	p	Cohen's d	Effect size
U	EC	2.12 ± 0.17				
	$CC1$	2.41 ± 0.20	11.9%	0.004^*	1.6	Very Large
	$CC2$	3.34 ± 0.18	36.5%	$< 0.001^*$	6.8	Huge
C	EC	2.22 ± 0.12				
	$CC1$	2.49 ± 0.13	10.8%	$< 0.001^*$	2.2	Huge
	$CC2$	3.47 ± 0.20	35.9%	$< 0.001^*$	7.6	Huge

Table 4 gives the average hand movement distance for the three conditions with the different values of the redirection gain threshold of Scene U.

TABLE 4
Average Hand Movement Distance (m)

T	Condition	Avg ± std. dev.	(CC_i-EC) / CC_i	p	Cohen's d	Effect size
$T1$	EC	2.28 ± 0.18				
	$CC1$	2.36 ± 0.16	4.0%	0.30	0.5	Medium
	$CC2$	3.71 ± 0.20	38.6%	$< 0.001^*$	7.3	Huge
$T2$	EC	1.95 ± 0.22				
	$CC1$	2.22 ± 0.16	12.0%	0.007^*	1.4	Very Large
	$CC2$	3.70 ± 0.19	47.3%	$< 0.001^*$	8.4	Huge
$T3$	EC	1.92 ± 0.22				
	$CC1$	2.02 ± 0.17	5.0%	0.29	0.5	Medium
	$CC2$	3.68 ± 0.19	46.0%	$< 0.001^*$	7.4	Huge

EC participants had shorter average hand movement distances for all tasks and the comparisons were significant except for $T1$ and $T3$ for *CC*1. Our approach directly optimizes the number of resets, not the hand movement distance. Since reset requires additional hand movement, the hand movement distance should be proportional to the number of resets. Therefore, when the number of resets is significantly reduced, the hand movement distance should also be significantly reduced.

However, in some specific cases, users may experience an increase in hand movement distance when using our method. For example, when there are already physical props around the virtual placement position, the physical prop being picked up might be placed near virtual objects that are not currently proxied by any physical prop. This offset can lead to an increase in hand movement distance. As a result, compared to *CC*1, our results are not statistically significant in $T1$ and $T3$ according to the data analysis. However, from the results, our method still achieves a 4% and 5% reduction in hand movement distance with $T1$ and $T3$, respectively.

Task Completion Time Table 5 gives the task completion time for the three conditions under different scenes with $T2$.

TABLE 5
Task Completion Time (s)

Scene	Condition	Avg ± std. dev.	(CC_i-EC) / CC_i	p	Cohen's d	Effect size
U	EC	196.3 ± 15.4				
	$CC1$	240.4 ± 20.8	18.4%	$< 0.001^*$	2.4	Huge
	$CC2$	407.4 ± 10.86	51.8%	$< 0.001^*$	15.8	Huge
C	EC	197.0 ± 7.5				
	$CC1$	228.0 ± 9.6	13.6%	$< 0.001^*$	3.6	Huge
	$CC2$	408.3 ± 10.2	51.8%	$< 0.001^*$	23.5	Huge

Table 6 gives the task completion time for the three methods with the different redirection gain thresholds of Scene U.

TABLE 6
Task Completion Time (s)

T	Condition	Avg ± std. dev.	(CC_i-EC) / CC_i	p	Cohen's d	Effect size
$T1$	EC	188.7 ± 9.1				
	$CC1$	198.2 ± 8.5	4.8%	0.03^*	1.1	Large
	$CC2$	382.6 ± 17.8	50.6%	$< 0.001^*$	13.6	Huge
$T2$	EC	188.0 ± 9.3				
	$CC1$	199.2 ± 9.5	5.6%	0.02^*	1.3	Very Large
	$CC2$	383.1 ± 14.7	47.4%	$< 0.001^*$	8.4	Huge
$T3$	EC	184.9 ± 5.2				
	$CC1$	195.2 ± 8.9	5.3%	0.008^*	1.4	Very Large
	$CC2$	373.6 ± 14.2	50.5%	$< 0.001^*$	17.7	Huge

EC achieves a significantly reduced task completion time with all scenes and thresholds than *CC*1 and *CC*2. The decrease in task time can be attributed to two factors. Firstly, due to our method's reduced hand movement distance, task completion time decreases, given a constant hand speed. Secondly, user interactions are smoother under our method, reducing the pause and thinking time when users need to reset.

Subjective Metrics We conducted a survey on users' workload using the NASA TLX questionnaire, and the results are shown in Figure 9. Overall, EC significantly reduces workload compared to *CC*1 and *CC*2. EC significantly reduces the number of reset, the average movement distance and the task completion time during the retargeting. Therefore, it can effectively reduce physical and cognitive workload, improving user experience.

The results of technical usability are presented in Figure 10. The figure shows that EC demonstrates higher usability in various measurement aspects. The categories used are Intuitiveness(IN), Efficiency(EF), Accuracy(AC), Naturalness(NA), Satisfaction(SA), Easiness(EA). EC has advantages in terms of all terms over *CC*2 and *CC*1, and the comparisons were significant except for naturalness for *CC*2, intuitiveness, accuracy, naturalness, and easiness for *CC*1. The main reason is that our method can significantly reduce the number of resets, hand movement distance, and task completion time during the interactions. These objective improvements greatly enhance the subjective perception of technical usability for users. In *CC*1, props are selected based on the closest distance to the selected virtual object, and the placement position is then determined by the relative positional relationship between the virtual object and its prop. It is based on a quite straightforward idea and achieves similar scores in intuitiveness and easiness. Accuracy refers to the degree of matching between the real hand and the virtual hand positions, measured in terms of redirection gain. Due to *CC*1 always selecting the closest physical prop to the virtual object and maintaining a fixed relative position during placement, it has a relatively small redirection gain at each step, leading to high perceived accuracy by the users. However, EC considers the current interaction and subsequent interactions, resulting in small redirection gains in both current and future steps, leading to a sustained perception of high accuracy by the users. Naturalness here refers to whether users can perceive the occurrence of redirection operations. In our user study, similar to the previous haptic retargeting methods [2], [22], [27], participants can perceive the presence of redirection when using all three methods but find them tolerable. As a result, the scores are close to each other, indicating that users perceive a similar level of naturalness across the three methods.

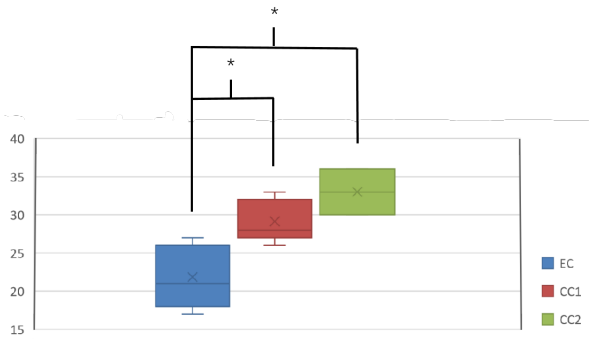

Fig. 9. Box plots for task load scores, for the three conditions. Asterisks denote statistical significance.

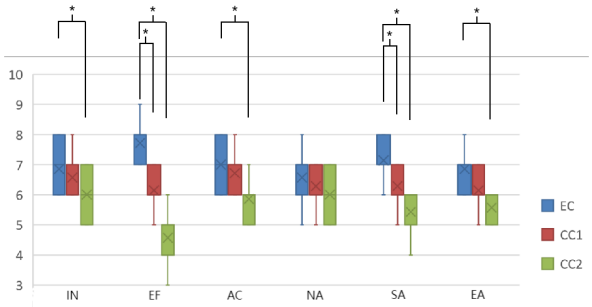

Fig. 10. Usability questionnaire results for individual questions. Significant differences are denoted with an asterisk. Scores of questions about negative features are complemented such that a higher score always means better.

5 User Study 2: Multiple User Task

User Study 2 is designed to evaluate the performance of our weighted proxy importance strategy. Figure 11 shows the top view of the scene used in the study. Ten virtual objects are randomly distributed on the 2m * 2m plane. Five virtual objects in red are owned by User1 and five virtual objects in green are owned by User2. Five blue props are randomly placed in the scene. The white cubes on the top left and the bottom right indicate the locations of two reset points for two users.

5.1 Study Design

Implementation In the study, two users were involved in the task. For each user, the interactions performed, as well as the hardware devices and software used were the same as in User Study 1. However, HaRT's design [28] is based on a single-user task, so we extended its functionality to support two-stage interactions with multiple users picking up and placing objects. We used a Unity component called Mirror for network synchronization and an intermediate component called SyncManager to keep track of the current global state of all objects. Whenever a user's state changes or needs to perform an action on the scene, the system notifies the SyncManager. Additionally, each user periodically requests the latest system state from SyncManager to update his or her own state accordingly.

Participants We recruited the same 24 participants from the previous single-user study to attend our experiment. All participants were randomly divided into 12 groups.

Conditions The study consisted of three experimental conditions $EC1-EC3$. In $EC1$, participants use our PIHR method with **Equivalent Strategy**, which considers all the virtual objects in the scene without considering the ownership. In $EC2$, participants use our PIHR method with **Ownership Strategy**, which only considers the virtual objects owned by the current user. In $EC3$, participants use our PIHR method with **Weighted Strategy**, which considers the different weights for the virtual object owned by different users.

Task This task involves two participants taking turns interacting with the virtual scene by moving cubes, like playing chess. Each participant is required to complete 25 moves (picking up and placing). Since the interaction between the two participants continues alternately, in order to avoid hand collisions between the two participants, we require the participants to move their virtual hand to the rest position (marked with a star) after completing one move and wait for the other participant to finish the interaction before picking up the next object.

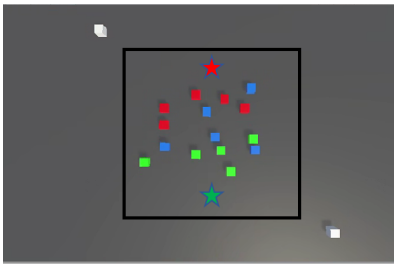

Fig. 11. The top view of the scene for two users in User Study 2. The star indicates the rest point for participants.

In the multi-user interaction task, we have introduced rest points. Therefore, when calculating the second factor of *PIHR*, we do not use the current position as the starting point for prediction. Instead, we calculate the *PR* term by placing the virtual hand at the rest point, while maintaining the real hand's offset relative to the virtual hand's position. The rest point is different from the reset point, as the rest point may not be far enough from the objects to guarantee a reset from any position. The reason for this approach is that the rest position becomes the starting position for the next move.

As in User Study 1, the virtual cubes being interacted and the corresponding placement locations in the virtual scene indicated by the system are the same in all conditions. The real prop's initial position and the distribution of the virtual objects at the task's beginning are also the same for all conditions.

5.2 Procedure

Two participants stood in front of a real table face to face, where five real cubes as props were placed. The real cubes and table were aligned with the virtual world, and the virtual hands were aligned with the participants' real hands. Before starting the user study, participants were provided with detailed explanations of the task and reset technique. Before the start of the study, a random picking and placing process was performed to familiarize participants with picking up and placing objects in the three conditions. The pair of participants conducted the task described above. Each participant had his or her own reset point, and when a hand position reset had to be performed, the user returned the hand to his or her own reset point to reset. The objective metrics, data obtaining and processing methods, and analysis were the same as in User Study 1.

5.3 Results & Discussion

Reset Rate Table 7 gives reset rates for three conditions. $EC3$ achieves a significantly reduced reset rate than $EC1$ and $EC2$.

TABLE 7
Reset Rate (%)

Condition	Avg ± std. dev.	$(EC_i\text{-}EC_3)$ / EC_i	p	Cohen's d	Effect size
$EC3$	0.412 ± 0.01				
$EC1$	0.46 ± 0.01	6.9%	$< 0.001^*$	3.94	Huge
$EC2$	0.45 ± 0.01	5.5%	0.01^*	3.48	Huge

This is because our PIHR method with **Weighted Strategy** considers not only virtual objects owned by the current user but also the proxy state of virtual objects that are not owned by the current user. Several candidates' physical props or placement positions may yield similar effects when picking up or placing objects. However, the choice that provides the optimal effect for one user may result in a poor effect for another user, while the second best choice may work well for the other user. Therefore, in such situations, $EC3$ outperforms $EC1$, which only considers virtual objects owned by the current user. Compared with $EC2$ using **Ownership Strategy**, $EC3$ prevents virtual objects not owned by the current user from having too much influence on the scene distribution, which is achieved by assigning different weights to virtual objects with different ownership and locations. Furthermore, $EC3$ avoids the situation where

physical props are distributed only around the virtual objects owned by one participant. This prevents the occurrence of local imbalances in the distribution of physical props throughout the entire scene, ensuring a more even distribution and avoiding local shortages or surpluses of physical props in the scene.

Average Hand Movement Distance Table 8 gives the average hand movement distance for the three conditions. *EC*3 achieves a significantly reduced average hand movement than *EC*1 and *EC*2.

TABLE 8
Average Hand Movement Distance (m)

Condition	Avg ± std. dev.	(EC_i-EC_3) / EC_i	p	Cohen's d	Effect size
*EC*3	2.1 ± 0.05				
*EC*1	2.30 ± 0.11	8.8%	0.01^*	2.37	Huge
*EC*2	2.31 ± 0.09	8.9%	0.004^*	2.73	Huge

Compared to *EC*1 and *EC*2, the reduction in hand movement distance of *EC*3 is due to its significant reduction in the number of resets. This reduction in resets leads to a significant decrease in the additional displacement movement introduced by resets, resulting in a smaller distance that needs to be covered.

Task Completion Time Table 9 gives the task completion time for three conditions. Compared to *EC*1 and *EC*2, *EC*3 significantly reduces the task completion time.

TABLE 9
Task Completion Time (s)

Condition	Avg ± std. dev.	(EC_i-EC_3) / EC_i	p	Cohen's d	Effect size
*EC*3	227.7 ± 5.7				
*EC*1	243.6 ± 4.7	6.5%	0.002^*	3.1	Huge
*EC*2	240.2 ± 5.5	5.2%	0.013^*	2.2	Huge

The reduction in task completion time can be attributed to the significant hand movement distance reduction when the hand speed remains constant.

6 Conclusion, Limitations & Future work

We have proposed a proxy importance based haptic retargeting with multiple props to provide passive haptics for users in immersive VR applications. The concept of proxy importance for props is introduced first, and then a proxy importance based prop selection and placement method for moving virtual objects is proposed. The results of the user study with a single user show that, compared to the state-of-the-art methods, our method achieves a significant reduction in the number of resets, the completing task time, hand movement distances, and task load without the cost of cybersickness. In that user study, we set different distributions for real props, including random, uniform, and clustered cases, and the results show that the effect of the initial position of real props is very small. We also have proposed the weighted proxy importance strategy to improve the performance of multi-user collaboration. In multi-user tasks, the use of the weighted proxy importance strategy proposed specifically for multi-user tasks significantly reduces the number of resets, the average hand movement distance, and the task completion time, compared to the direct use of the single-user method described earlier. Our approach can be applied to the applications that use a small number of physical props to provide passive haptics for a larger number of virtual objects, such as chess playing, block stacking, and parts assembly. Our approach reduces the number of resets to a certain extent, making the user interaction more continuous and reducing the workload. This is especially beneficial for scenarios with long periods of continuous interaction.

Our method has several limitations. The first limitation is that, as with all previous methods that do not predict user behavior, our method may have situations where props and placement locations are not optimal when viewed over a long period of time. This is because our method, like most previous work, optimizes the selection of props and placement locations based only on the virtual objects and placement locations chosen for a given next step (with a script for the next interaction), regardless of the subsequent sequence of user interactions. There is also a case of completely unscripted interaction, for which Clarence et al. proposed a prediction method based on a long and short-term memory network [36], which predicts the next selected virtual object and the location where the user wants to place it based on the current trajectory and angle of the user's hand. If our method wants to integrate with such methods, we need to consider extending the fixed valid proxy distance to the variable valid proxy distance. The second limitation is that our method only considers the optimization of haptic retargeting techniques using touch resetting, and cannot be directly applied to the optimization of haptic retargeting using other resetting methods. The third limitation is that our method optimizes a 2D placement position when placing and does not take into account height variations. The proxy importance presented in this paper mainly considers objective factors related to the distribution of physical props and virtual objects in the scene. In fact, many other factors also affect the haptic retargeting process. For example, in more complex scenes, different shapes of physical props may be required, and the shape matching between virtual objects and physical props will affect the user experience; whether the location of physical props picking up/placing is ergonomically correct for more natural user interactions; and which physical props to use to minimize the duration of the reset process in situations where the reset is unavoidable. We will consider and explore these factors in our future work.

In the future, we will consider adding predictions for user interaction sequences, for example, using models of the user's gaze point and attention to estimate interaction sequences, which helps us to choose better props and placements. Another work we will do in the future is to study the characteristics of different resetting techniques and explore a general optimization method for haptic retargeting. The last work is to extend our approach to 3D environments. We only tested our method on two user cases for multi-user collaboration. In the future, the method can be used for more users directly.

Acknowledgments

This work is supported by the National Natural Science Foundation of China through Project 61932003, 62372026, by Beijing Science and Technology Plan Project Z221100007722004, and by National Key R&D plan 2019YFC1521102.

REFERENCES

[1] 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, 2016, pp. 1968–1979.

[2] 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, 2017, pp. 3718–3728.

[3] 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). IEEE, 2019, pp. 19–27.

[4] B. J. Matthews and R. T. Smith, "Head gaze target selection for redirected interaction," in SIGGRAPH Asia 2019 XR, 2019, pp. 13–14.

[5] X. Yang, Y. Kang, and X. Yang, "Retargeting destinations of passive props for enhancing haptic feedback in virtual reality," in 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW). IEEE, 2022, pp. 618–619.

[6] H. Iwata, H. Yano, F. Nakaizumi, and R. Kawamura, "Project feelex: adding haptic surface to graphics," in Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 2001, pp. 469–476.

[7] K. R. Vaghela, A. Trockels, and M. Carobene, "Active vs passive haptic feedback technology in virtual reality arthroscopy simulation: Which is most realistic?" Journal of Clinical Orthopaedics and Trauma, vol. 16, pp. 249–256, 2021.

[8] W. A. McNeely, "Robotic graphics: a new approach to force feedback for virtual reality," in Proceedings of IEEE Virtual Reality Annual International Symposium. IEEE, 1993, pp. 336–341.

[9] B. E. Insko, Passive haptics significantly enhances virtual environments. The University of North Carolina at Chapel Hill, 2001.

[10] L. Kohli and M. Whitton, "The haptic hand: providing user interface feedback with the non-dominant hand in virtual environments," in Proceedings of Graphics Interface 2005. Citeseer, 2005, pp. 1–8.

[11] S. Bovet, H. G. Debarba, B. Herbelin, E. Molla, and R. Boulic, "The critical role of self-contact for embodiment in virtual reality," IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 4, pp. 1428–1436, 2018.

[12] M. Elbehery, F. Weidner, and W. Broll, "Haptic space: The effect of a rigid hand representation on presence when interacting with passive haptics controls in vr," in Proceedings of the 19th International Conference on Mobile and Ubiquitous Multimedia, 2020, pp. 245–253.

[13] T. Nakamura and A. Yamamoto, "Multi-finger electrostatic passive haptic feedback on a visual display," in 2013 World Haptics Conference (WHC). IEEE, 2013, pp. 37–42.

[14] M. Achibet, B. Le Gouis, M. Marchal, P.-A. Leziart, F. Argelaguet, A. Girard, A. Lécuyer, and H. Kajimoto, "Flexifingers: Multi-finger interaction in vr combining passive haptics and pseudo-haptics," in 2017 IEEE Symposium on 3D User Interfaces (3DUI). IEEE, 2017, pp. 103–106.

[15] J. J. Gibson, "Adaptation, after-effect and contrast in the perception of curved lines." Journal of experimental psychology, vol. 16, no. 1, p. 1, 1933.

[16] E. Burns, S. Razzaque, A. T. Panter, M. C. Whitton, M. R. McCallus, and F. P. Brooks, "The hand is slower than the eye: A quantitative exploration of visual dominance over proprioception," in IEEE Proceedings. VR 2005. Virtual Reality, 2005. IEEE, 2005, pp. 3–10.

[17] 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. IEEE, 2007, pp. 295–296.

[18] L. Kohli, M. C. Whitton, and F. P. Brooks, "Redirected touching: Training and adaptation in warped virtual spaces," in 2013 IEEE Symposium on 3D User Interfaces (3DUI). IEEE, 2013, pp. 79–86.

[19] 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, 2017, pp. 759–771.

[20] 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, vol. 24, no. 4, pp. 1467–1476, 2018.

[21] 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, vol. 27, no. 5, pp. 2627–2637, 2021.

[22] 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). IEEE, 2019, pp. 47–55.

[23] A. Clarence, J. Knibbe, M. Cordeil, and M. Wybrow, "Investigating the effect of direction on the limits of haptic retargeting," in 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE, 2022, pp. 612–621.

[24] N. Ogawa, T. Narumi, and M. Hirose, "Effect of avatar appearance on detection thresholds for remapped hand movements," IEEE transactions on visualization and computer graphics, vol. 27, no. 7, pp. 3182–3197, 2020.

[25] 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). IEEE, 2020, pp. 269–278.

[26] 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, 2018, pp. 1–12.

[27] 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, 2021.

[28] 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, 2021, pp. 1–7.

[29] S. G. Hart, "Nasa-task load index (nasa-tlx); 20 years later," in Proceedings of the human factors and ergonomics society annual meeting, vol. 50, no. 9. Sage publications Sage CA: Los Angeles, CA, 2006, pp. 904–908.

[30] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in Advances in psychology. Elsevier, 1988, vol. 52, pp. 139–183.

[31] H. Kim, G. Lee, and M. Billinghurst, "A non-linear mapping technique for bare-hand interaction in large virtual environments," in Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction, 2015, pp. 53–61.

[32] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," Biometrika, vol. 52, no. 3/4, pp. 591–611, 1965.

[33] S. W. Greenhouse and S. Geisser, "On methods in the analysis of profile data," Psychometrika, vol. 24, no. 2, pp. 95–112, 1959.

[34] A. Gelman, "Analysis of variance," Quality Control Applied Statistics, vol. 20, no. 1, pp. 295–300, 2005.

[35] J. Cohen, Statistical power analysis for the behavioral sciences. Academic press, 2013.

[36] A. C. J. K. M. C. M. Wybrow, "Unscripted retargeting: Reach prediction for haptic retargeting in virtual reality," in IEEE Conference on Virtual Reality and 3D User Interfaces. IEEE, 2021, pp. 150–159.

Ziming Liu is a M.A student in the the School of Computer Science and Engineering of Beihang University, China. His current research focuses on virtual reality, augmented reality and visualization.

Jian Wu is a Ph.D student in the the School of Computer Science and Engineering of Beihang University, China. His current research focuses on virtual reality, augmented reality and visualization.

Lili Wang received her Ph.D. degree from the Beihang University, Beijing, China. She is a professor with the School of Computer Science and Engineering of Beihang University, and a researcher with the State Key Laboratory of Virtual Reality Technology and Systems. Her interests include virtual reality, augmented reality, mixed reality, real-time rendering and realistic rendering.

Xiangyu Li is a M.A student in the the School of Computer Science and Engineering of Beihang University, China. His current research focuses on virtual reality and visualization.

Sio Kei Im received his Ph.D. degree in Electronic Engineering from Queen Mary University of London (QMUL), United Kingdom. He is a professor at the Faculty of Applied Sciences, Macau Polytechnic University, and a researcher at the Engineering Research Center of Applied Technology on Machine Translation and Artificial Intelligence, Ministry of Education. His research interests include video coding, image processing, machine learning for NLP and multimedia.