Communications in Information and Systems
Volume 23, Number 4, 393–422, 2023

Natural hand remapping: velocity adaptive hand manipulation for VR

Yike Li, Chen Wang, Ge Yu, Yu He, Stefanie Zollmann, and Song-Hai Zhang

In virtual reality (VR), hand remapping modifies the commonly used one-to-one mapping between tracked and virtual hand positions, which extends the interaction scope of hands while also sacrificing naturalness. To address this issue, we propose remapping techniques that are both natural and efficient, taking hand movement velocity into consideration. Our approach is based on the insight that slow and fast hand velocities indicate the intent for precise or rapid manipulations respectively. Therefore, the hand movement should be scaled down or up accordingly. We first estimated the detection thresholds for remapped hands using a 2-alternative-forced-choice (2AFC) design. Based on these thresholds, we then designed a hand remapping function that automatically adjusts the remapping scale based on hand movement velocity. To evaluate the effectiveness of our proposed velocity-adaptive hand remapping technique, we further conducted a user study in which participants performed both rapid and precise tasks. Results showed that our proposed velocity-adaptive hand remapping, though imperceptible to participants, is able to significantly outperform other techniques. Overall, our work demonstrates the potential of velocity-guided redirection techniques for hand interactions in VR.

Keywords and phrases: Hand remapping, detection thresholds, redirection techniques.

1. Introduction

Virtual reality (VR) aims to provide an immersive experience for users. As part of such immersive experiences interaction techniques that closely follow natural physical interaction are often desired. In recent years, we have seen advances in technology that now easily enable users to interact with 3D virtual environments (VE) using their hands, heads, and bodies. An approach that is widely used to determine the movements in VE as in many commercial products is a direct one-to-one mapping of the tracked body movements. However, it is often restricted by physical constraints, such as limited physical space around the user, inconvenience or fatigue caused by physical movements, or body restrictions. For example, for seated VR, users can only rotate their heads at most 180 degrees and their arms are only capable of reaching a small portion of space. Such predicaments prompt the emergence of remapping techniques, especially for hand interaction.

To improve the sense of presence and enrich ways of interaction, hand input is widely used in VR applications through controllers, gloves, or directly tracked hand movements and recognized gestures. Prior work has shown that vision has a dominant role when visual, auditory, or other signals influx into the brain [1]. By leveraging this visual dominance effect [2], non-isomorphic hand mappings are explored to decouple real and virtual hand movement, enabling virtual hand ends into a position different from where the real hand is. These mappings consist of adding a fixed offset [3], gain-based movement [4], rotating the VE [5].

However, hand repositioning techniques cannot be applied without consideration of amplitude, since large deviations will possibly be noticed by uses and induce a sense of unnaturalness. An ideal hand remapping should increase user immersion and create realistic interactions. Thus, the estimation of detection thresholds within which remapping is undetectable to users is important for applications that aim to maintain a sense of realism while still leveraging the benefits of the aforementioned techniques. While previous work has estimated the detection thresholds for constant hand offsets [3], scaled hand movements [6], and hand redirection using haptic retargeting [4], there is still no investigation into how these thresholds might be affected by the change of velocity. Existing research [7] has already shown that manipulation of human sensitivity towards path curvature in redirected walking is correlated to walking speed, so it is also necessary to understand the relationship between the sensitivity towards remapped hands and hand movement velocity. Meanwhile, a significant drawback of using VR equipment is that we can not control our body parts precisely, like rotating our head with an HMD at a specific angle or speed or moving our hand with VR controllers at a restricted distance or velocity in virtual 3D space. There are two main reasons for this phenomenon: first, although hardware and tracking technology for VR have been quite robust, tracking devices continuously have inherent jitter with the real world [8]; Second, although our bodies are flexible, the rotational structure of human joints makes it challenging for us to move our hands in a straight line without any physical support in the real world [9].

Combining the two considerations, we see a large potential to implement novel remapping techniques that are dependent on hand movement velocity. We assume that the velocity reflects the user's intention of movements.

Therefore remapping techniques that alter virtual hand movements according to velocity could possibly be more natural and more efficient. To keep a balance between interaction efficiency and naturalness, we aim to restrict the manipulation within detection thresholds. Firstly, we designed and conducted a psychophysical experiment utilizing a two-alternative forced-choice (2AFC) methodology to test user perception of constant hand movement manipulation under three levels of controlled speed and estimate detection thresholds using a psychometric function. Participants were asked to follow a moving plate in front of them with a designated physical movement speed. In contrast manipulation of the hand movements were applied with varying magnitudes and directions in each trial of the task. After each trial, participants reported if their hand movement was *normal* or *not normal*. These responses were recorded and analyzed to determine the detection thresholds in different velocities and directions. Secondly, we present a velocity-sensitive hand remapping technique with imperceptible scaled manipulation in VR according to the determined detection thresholds mentioned above applied at the target speed interval. Our hand remapping method estimates the application context for fast or precise manipulation based on real-time behavior. When the context of rapid manipulation is detected, our enhancing remapping technique dynamically adjusts the manipulation to an increased scale, making it more sensitive to the user's hand movement and vice versa. Lastly, we designed an experiment in which users perform both precise and rapid tasks. Our results showed that the velocity-adaptive remapping strategy outperformed other techniques both in terms of efficiency and accuracy.

2. Related work

Comfortable and natural remapping of user movements is an important and vigorous research topic in VR. In this section, we give a brief review of redirection techniques in VR research, hand remapping techniques, as well as previous velocity based remapping techniques.

2.1. Redirection techniques and threshold detection

As mentioned before, the most common and intuitive way to model virtual camera motions is to directly do one-to-one mapping of physical movements. However, users' interactions in virtual scenes are often constrained by physical limitations. Therefore, in order to provide a better user experience and overcome real-world limitations, prior works have explored remapping techniques for a wide variety of VR applications, including seven league boots [10], redirected walking [7, 11], head rotations [12, 13, 14, 15] and hand positions [4, 16, 5]. At the same time, it is necessary to also consider the negative effects introduced by these remapping techniques, such as sickness [14, 13] or spatial orientation [13].

Razzaque et al. [11] proposed a new technique called redirected walking that allows users to walk much longer distances in virtual environments even in constrained physical spaces. It changes the real walking direction of users by applying additional rotations proportional to their angular velocity to the virtual movement. Experiments in a CAVE [17] further demonstrated the potential of redirected walking by showing that redirection increased the sense of presence. Steinicke et al. [18, 19] later summarized redirected walking as three different gains, including translation gains, rotation gains and curvature gains, defined by scaling factors applied to the movement in the virtual world. They also established standards for the estimation of a threshold within which those gains would be unnoticed by users. Grechkin et al. [20] then concluded that curvature gains and translation gains could be used simultaneously without a conflict. Curvature gains can be used more effectively with controlled velocity [7].

In seated VR scenarios, such as sitting on a couch at home, rotation gains are also a common consideration to enable people to explore virtual scenes both comfortably and exhaustively. While redirected walking techniques often alter the movement of a virtual camera or view direction according to the physical movements of the user, users can only move their upper body and head freely when seated. In these situations, the rotation angles of the head are captured by VR devices, mostly VR HMDs. To allow users to view a larger range of a virtual scene with fewer physical turns, head rotations are often amplified through a factor called amplification factor [14] or rotation gains [19, 12]. Sargunam et al. [13] proposed guided head rotations that made the 360 degree virtual scene available within the limitations of physical rotations. Their method also rotated the virtual scene back to neutral position to avoid neck twisting for a long time. Their experiments revealed that large rotation angles would cause more sickness. Stebbins et al. [15] expanded this work to immersive narrative experiences by rotating the virtual content slowly and encouraging people to align their bodies to a more comfortable viewing direction. The effect of amplification depends on the displays. Ragan et al. [14] found that compared to HMDs, it was easier for users to maintain spatial orientation in CAVE. Langbehn et al. [12] later gave a detailed comparison between different amplification methods regarding their linearity as well as dynamism and they deduced that dynamic rotation gains had higher usability and produced less sickness.

Inspired by both redirect walking and head amplification techniques, in this paper, we remap hand movement in virtual environments to provide more effective and natural manipulation and interaction for users.

2.2. Hand remapping techniques

Hands are an indispensable channel through which users interact within virtual worlds when they are using VR devices. Therefore, researchers have investigated many hand remapping and repositioning techniques, including how hands might be remapped in certain situations [16, 4], and to which extent hands can be remapped without being noticeable to users [6].

The Go-Go technique proposed by Poupyrev et al. [16] prolonged arm reaches through a smooth and non-linear mapping which scaled up the virtual arm after reaching a threshold distance (set as 2/3 of arm length). However, Poupyrev et al. did not investigate the naturalness of their design. Dominjon et al. [21] filled in this gap, by demonstrating that the C/D ratio (the ratio between the amplitude of the real hand and virtual hand position) influences the perception of the mass of objects that users manipulate. Frees et al. [9] designed a dynamic C/D ratio function and proved that a C/D ratio less than 1.0 provided more precise control. Re-positioned hands may influence the amount of immersion as well. Several works aim at maintaining high body ownership when designing their interaction techniques, such as slower shifting for overhead targets [22], using realistic avatars [23], or an improved ergonomic design [24, 25]. Altered perception also makes other hand redirection techniques like haptic retargeting possible. A single physical object can provide haptic feedback for multiple virtual objects located at different places [5], even when their shapes and sizes vary [26, 27].

Besides perception, researchers also probed into how remapping technologies influence the performance in different tasks. Li et al. [28] examined four cursor offset techniques (no offset, fixed-length offset, linear offset and nonlinear offset) for navigation tasks in a CAVE system. Their results showed that the linear offset technique outperformed other techniques. They also evaluated the same four techniques on target selection when objects are beyond arm's reach [29]. However, their work is difficult to generalize since it does not apply to tasks that place objects near users and they did not take direction into consideration.

To accurately determine the range of a remapping technique that is imperceptible to the user, psychometric analysis has been employed in previous research. This often features a two-alternative-forced-choice (2AFC) design in which users have to choose from two opposite options. Also, previous research has shown that directions play an important part in the dimensions of virtual hand movements. Zenner and Kruger calibrated the detection thresholds for horizontal, vertical, and gain-based hand warping in three desktop-scale VR scenarios [4]. Later, Benda et al. [3] identified significant threshold differences across dimensions when applying fixed positional offsets. Whether the offset is within the suitable range correlated heavily with target selection. Similarly, Esmaeili et al. [6] studied scaled hand movements by estimating 28 types of linear scale from 0.5 to 2.0 in three dimensions. They also found the detection thresholds differed significantly between different hand movement directions.

2.3. Velocity based remapping

In VR, velocity has been found to be associated with users' sensitivity and perception of the change of virtual scenes [7]. Neth et al. [7] found that when walking at lower speeds, users were less sensitive to walking on a curved path through a psychological experiment. Thus, they proposed a velocity-dependent algorithm for redirected walking, allowing the curvature gains to alter dynamically according to the walking speed, which was more flexible than previous static reorientation techniques. Similarly, head rotation velocity has also been found to be related to rotation gains [30, 31]. Speed can also be used as a factor to alter the precision of manipulation. In PRISM [9], users indicate whether they try to be precise or rapid through their hand speed. The hand movement would be scaled down to enable more precise control. However, it uses more of a hand-crafted mapping and lacks the understanding of how different levels of hand movement velocity would affect user perception towards the scale. Adaptive pointing that adjusts the C/D ratio according to speed has also been used in 2D screen touch [32, 33, 34].

To the best of our knowledge, there is no previous research that studies how users' sensitivity to the scale of remapping differs when the level of speeds changes and controls the remapping function within the detection thresholds, which is the aim of our work.

3. Experiment 1: detection thresholds with controlled hand movement velocity

Before developing an immersive hand interaction technique that is imperceptible and comfortable, we first investigated the detection thresholds at

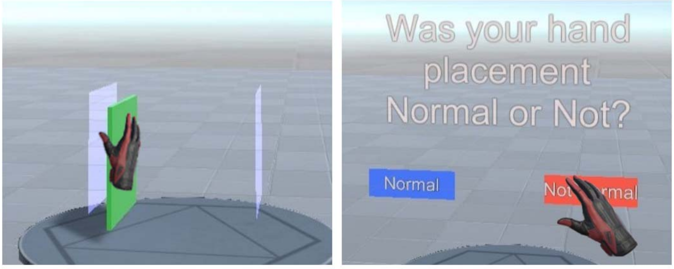

Figure 1: The virtual environment for Experiment 1. (*Left*) The moving plate users followed during the task. (*Right*) The question presented to users after completing each trial.

different hand movement speeds, since previous studies never control the hand movement speed while measuring detection thresholds. Therefore, we designed an experiment to apply scale values of varying magnitudes in three directions while users follow a moving plate that moves at designated speeds. The users then had to respond if they feel their hand movements were normal or not during each trial.

3.1. Experiment design

We performed a 3 *Reference Speeds* (fast, medium, slow) × 3 *Directions* (horizontal, vertical, depth) experiment to the estimate detection thresholds of each group. In order to determine the appropriate *Reference Speeds* to follow, we conducted a pilot study involving 15 participants to measure the range of hand movement velocities in virtual environments. We instructed users to move their hands between two plates in each direction with a briefing on either performing fast or slow actions in the virtual environment. The slow mode is when people are moving slowly and carefully towards a target, i.e. holding and moving a slider, and the fast mode is when people move fast but still feel they are in control of their body and sense the movement. It is important to note that we are not exploring the complete range of potential movements, as we do not require participants to move extremely slowly or extremely fast. The default distance between the two plates was 0.4 m, slightly tuned according to participants' heights and we logged how much time they spent on each movement. Through data analysis, we found that in the slow conditions, the average hand movement speed is 0.15 m/s, and in the fast condition, the average hand movement speed is 0.56 m/s, which we set as *basic range*. Therefore, in the first experiment we used the following three hand movement speeds that roughly covered the range from slow movements to fast movements:

- Fast speed mode: 0.56 m/s
- Medium speed mode: 0.32 m/s
- Slow speed mode: 0.15 m/s

Although previous work choose a scale factor of 0.5 (which is the slowest scale) and a scale factor of 2.0 (which is the fastest scale) as the two most extreme scale values [6], we found that 0.6 and 1.8 are enough to make sure participants detected it as not normal in our pilot experiments. In order to control the experiment time and reduce participants' fatigue to maintain the quality of their responses when performing the experiment, we sampled fewer values compared to [6] ranging from 0.60 to 1.80, listed as follows:

- Slow-scaled values (7 values): 0.60, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95
- Fast-scaled values (7 values): 1.10, 1.20, 1.30, 1.40, 1.50, 1.60, 1.80

Since we aim to control participants' physical hand movement speed, the plane movement speed was changed according to the tested scale values in a trial by the following formula:

$$V_p = s \times R \tag{1}$$

- V_p: plate movement speed
- R: reference speed
- s: scale value

For the first experiment, we followed a study design close to the study design of Esmaeili et al. [6]. We used a with-subjects design and each participant completed all conditions and trials. The independent factors include scale value, axis, and velocity level. We used 21 (3 × 7) scale values, consisting of 7 different values of faster hand movements, 7 different values of slower hand movements, and 7 normal hand movements (scale = 1.0). We repeated all of the scales for movements on the three different axes: horizontal (X), vertical (Y), and depth (Z). In each direction, we set the first two scale values to normal (scale = 1.0) in a practice trial. This allows users to get used to the change of movement direction. The two practice trials were not added to the recorded data. We also repeated all of the scale values and all three axes for three different hand speed modes. Each condition was tested twice to strengthen data analysis. Therefore, every participant completed 378 trials (3 speeds × 2 repeats × 3 axes × 21 scale values). The whole experiment was divided into 6 blocks, each block contained 3 axes ×

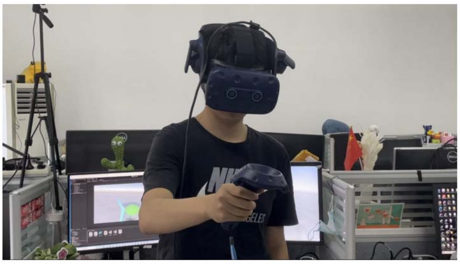

Figure 2: Setup for the first experiment: a participant is conducting the experiment, at the same time the virtual environment can be monitored through a screen.

21 scales = 63 trials for a given reference speed and the axis in each trial was randomized. The ordering of reference speeds was counterbalanced between participants.

During the experiment, participants used a hand-held controller (since most VR applications require users to interact with a controller) to follow a moving virtual plate. The plate would change its color depending on how well the user's hand speed matched its speed. When the difference between hand velocity and plate velocity was less than 0.05 in slow speed mode or less than 0.1 in medium and fast speed mode, the moving plate would be shown in green. For differences between 0.05 and 0.1 in slow mode or between 0.1 and 0.2 in medium and fast speed mode, the moving plate would be shown in yellow. In other cases, the moving plate would be shown in red. For this purpose, we check users' hand speed in every frame and compare it with the plate movement speed. In each trial, participants had to follow the plate for 6 rounds (3 rounds for both left and right directions), and then they had to answer the question *Was your hand movement normal or not?*. Then the program logged their answers (*normal* or *not normal*), response time, and average hand movement speed for further analysis.

3.2. Technique

Considering the purpose of our experiment was to determine the detection of thresholds for scaled hand movement, in an isolated axis under specific hand movement velocity control, we used the predecessor's techniques proposed by Esmaeili et al. to isolate scaled movements on one axis at a time [6]. We used the following formula to calculate the user's hand position in the virtual scene depending on their hand position in the previous frame and add a scaled offset. Let $x_{\text{vir}}(t), y_{\text{vir}}(t), z_{\text{vir}}(t)$ denote the virtual (display) position of the tracked hands and $x_{\text{real}}(t), y_{\text{real}}(t), z_{\text{real}}(t)$ denotes the real (motor) position of the tracked hands. For the sake of brevity, we report only the $x_{\text{vir}}(t)$ calculation in the following which $y_{\text{vir}}(t)$ and $z_{\text{vir}}(t)$ are applied to the same calculations.

$$x_{\text{vir}}(t) = x_{\text{vir}}(t-1) + s \times (x_{\text{real}}(t) - x_{\text{real}}(t-1)) \tag{2}$$

- $x_{\text{vir}}(t)$: virtual hand position in current frame
- $x_{\text{vir}}(t-1)$: virtual hand position in previous frame
- $x_{\text{real}}(t)$: motor hand position in current frame
- $x_{\text{real}}(t-1)$: motor hand position in previous frame
- s: current scale value

3.3. Apparatus

The experiment was conducted in our laboratory using an HTC Vive Pro headset, tracked by two Base Station 2.0, as well as the right-hand controller since all participants finished the experiment with their right hands. The software was developed with Unity3D 2020.2.3f1c1 and one unit in the virtual space represents one meter in real-world space. The desktop computer ran on 64-bit Windows 10 using a 16 GB RAM and 3.6 GHz 8-Core processor with a GeForce RTX 2060 SUPER graphics card and logged related data for analysis.

3.4. Participants

Twelve university students (7 females, 5 males) participated in the experiment. Their age ranged from 22 to 26 with a median of 23 years old. Four participants reported they have never used a VR headset before. All participants completed the experiment with their right hand.

3.5. Procedure and task

Before the beginning of the experiment, we obtained informed consent from participants. We first explained the purpose and procedure of our study and clarified the confusing points. They would also fill in a demographic questionnaire and a Simulator Sickness Questionnaire (SSQ) [35]. Then the experimenter helped them to adjust and wear the headset.

The main goal of this experiment is to analyze detection thresholds for simple, axis-isolated movements depending on velocity via completing a simple target-following task. Once participants hit the still plate, it started to move along one axis back and forth at a specific speed. We then applied an offset to the user's hand representation per frame. The plate movement speed depends on the reference speed (0.15 m/s, 0.32 m/s, or 0.56 m/s) and the scaled values. Participants were asked to follow the moving plate and try their best to keep the moving plate green. After the plate moved back and forth six times, questions would be displayed on top of the scene, and participants were asked to decide whether the virtual hand movement was normal or not normal. After completing a 21-scale session, users would be given a 15 seconds mandatory break to relax their hands. The moving direction of the plate then changed to another direction. Once participants completed a block of the experiment (3 directions × 21 scaled values), the experimenter helped them take off the headset, instructed them to fill out another SSQ form, and had a 3-minutes break. Each block lasted about 15 minutes and the whole experiment contained 6 blocks and took about 90 minutes. Participants were allowed to experience three speed modes prior experiment using 1.0, 0.6, and 1.8 scale factors before starting the study.

3.6. Results

The estimation of detection thresholds was conducted from the records of participants' responses on the normality of their hand movements for each scaled movement value/direction pair. By aggregating the responses on normality into a total probability, we could fit a prediction function and a curve. Detection thresholds were determined by this prediction function for a 50% probability level using 0.5 as a threshold. We used the *Quickpsy* [36] package in R to compute the point of subjective equality (PSE) and the associated standard deviation (SD). CIs are calculated with parametric bootstrapping using a percentile method with 95% PSE confidence exists in the interval. We utilized it to fit 18 curves for each data point grouping aggregated probability by direction and speed mode, generated 18 thresholds, and used parametric bootstrapping to estimate confidence intervals (CIs) for each threshold. Fitted functions are shown in Figure 3 and precise PSE and SD values are presented in Table 1. *Quickpsy* fits, by direct maximization of the likelihood [37], psychometric functions of the form:

$$\Psi(x;\alpha,\beta,\gamma,\lambda) = \gamma + (1-\gamma-\lambda)F(x;\alpha,\beta) \tag{3}$$

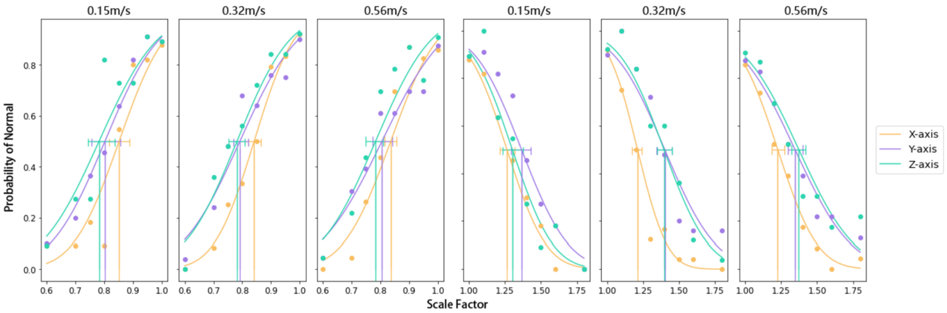

Figure 3: Fitted logistic functions for each speed level, paired by the axis of movement. Dropdown lines mark the threshold point. Error bars indicate 95% confidence intervals expressed in Table 1.

Table 1: Point of subjective equality (PSE) and the corresponding standard deviations (SD) for both fast scales (top) and slow scales (bottom) in each condition

Speed (m/s)	Axis	PSE	SD
0.15	X	1.264 [1.309,1.199]	0.215 [0.156, 0.280]
	Y	1.212 [1.246,1.187]	0.162 [0.131, 0.195]
	Z	1.227 [1.256,1.193]	0.217 [0.174, 0.258]
0.32	X	1.365 [1.298, 1.442]	0.258 [0.184, 0.353]
	Y	1.401 [1.349, 1.454]	0.280 [0.221, 0.329]
	Z	1.347 [1.289, 1.403]	0.309 [0.255, 0.384]
0.56	X	1.302 [1.250, 1.367]	0.223 [0.152, 0.297]
	Y	1.396 [1.352, 1.434]	0.245 [0.201, 0.287]
	Z	1.373 [1.311, 1.431]	0.306 [0.248, 0.376]

Speed (m/s)	Axis	PSE	SD
0.15	X	0.852 [0.820, 0.882]	0.125 [0.091, 0.156]
	Y	0.803 [0.761, 0.844]	0.148 [0.107, 0.196]
	Z	0.782 [0.728, 0.828]	0.160 [0.119, 0.211]
0.32	X	0.842 [0.822, 0.863]	0.109 [0.094, 0.131]
	Y	0.792 [0.757, 0.822]	0.160 [0.130, 0.207]
	Z	0.782 [0.751, 0.806]	0.145 [0.118, 0.183]
0.56	X	0.837 [0.813, 0.860]	0.129 [0.104, 0.153]
	Y	0.806 [0.775, 0.838]	0.172 [0.139, 0.216]
	Z	0.784 [0.752, 0.813]	0.153 [0.118, 0.182]

Table 2: To determine the significant differences in fast and slow scales, we used a parametric bootstrap test. A parameter is considered different between two groups if the confidence intervals do not contain zero. The results for both fast scales (top) and slow scales (bottom) are sorted by both speed levels and directions

		0.15 m/s			0.32 m/s			0.56 m/s		
		X	Y	Z	X	Y	Z	X	Y	Z
0.15 m/s	X	–	**0.101***	0.038	−0.052	**0.137***	**0.132***	−0.037	**0.083***	**0.109***
	Y	**−0.101***	–	−0.064	**−0.154***	0.035	0.030	**−0.139***	−0.019	0.008
	Z	−0.038	0.064	–	**−0.090***	**−0.099**	0.039	**−0.075***	−0.045	0.071
0.32 m/s	X	0.052	**0.154***	**0.090***	–	**0.189***	**0.184***	0.015	**0.135***	**0.161***
	Y	**−0.137***	−0.035	**−0.099***	**−0.189***	–	−0.005	**−0.174***	−0.054	−0.028
	Z	**−0.132***	−0.030	−0.039	**−0.184***	0.005	–	**−0.169***	−0.049	−0.023
0.56 m/s	X	0.037	**0.139***	**0.075***	−0.015	**0.174***	**0.169***	–	**0.120***	**0.147***
	Y	**−0.083***	0.019	0.045	**−0.135***	0.054	0.049	**−0.120***	–	0.027
	Z	**−0.109***	−0.008	−0.071	**−0.161***	0.028	0.023	**−0.147***	−0.027	–

		0.15 m/s			0.32 m/s			0.56 m/s		
		X	Y	Z	X	Y	Z	X	Y	Z
0.15 m/s	X	–	0.049	**0.070***	0.010	**0.060***	**0.070***	0.015	**0.046***	**0.068***
	Y	−0.049	–	0.020	−0.039	0.011	0.021	−0.034	−0.003	0.019
	Z	**−0.070***	−0.020	–	**−0.059***	−0.010	0.000	−0.054	−0.024	−0.001
0.32 m/s	X	−0.010	0.039	**0.059***	–	**0.050***	**0.060***	0.005	0.036	**0.058***
	Y	**−0.060***	−0.011	0.010	**−0.050***	–	0.010	−0.045	−0.014	0.008
	Z	**−0.070***	−0.021	−0.000	**−0.060***	−0.010	–	**−0.055***	−0.024	−0.002
0.56 m/s	X	−0.015	0.034	0.054	−0.005	0.045	**0.055***	–	0.031	**0.053***
	Y	**−0.046***	0.003	0.024	−0.036	0.014	0.024	−0.031	–	0.022
	Z	**−0.068***	−0.019	0.001	**−0.058***	−0.008	0.002	**−0.053***	−0.022	–

To examine whether there exist significant differences in the thresholds across different speeds and directions, we employed the *thresholdcomparisons* function of *Quickpsy*. This function performs comparisons between groups for all possible pairs of groups using the bootstrap method. A parameter is considered different between two groups if the confidence intervals do not contain zero. The results of pairwise comparisons were shown in Table 2, which showed that there are significant differences between thresholds for each direction in the same speed. However, we could not find a significant difference between different speeds. When considering pairwise differences, two groupings emerge a group of X-axis and Y-axis, a group of X-axis and Z-axis, which means only directions along the X-axis were significantly different against each other which shows sensitivity to horizontal-related scaled movement.

The estimation of the detection threshold is based on the premise of hand velocity close to the desired speed, therefore we logged participants' average hand speed of completion in each trial. The results show that the average moving speeds of the hand under the three-speed modes are 0.162 m/s, 0.361 m/s and 0.610 m/s respectively, and the standard deviation are

0.023 m/s, 0.065 m/s and 0.139 m/s. The results showed that our speed control was effective.

We also calculated the *Total Score* (*TS*) of SSQ before and after the experiment, which was 2.47 and 31.4 respectively.

3.7. Discussion

In the previous work of Esmaeili et al. [6], detection thresholds for scaled hands were determined in two experiment settings: one is completed in total freedom without any restrictions and task, moving their hand alone along one of the three axes using a scale factor until an answer was determined by the participants (basic setting). In the other setting, participants were required to answer the 2AFC question after completing a game that had three-dimensional movements (complex setting). The estimated detection thresholds ranged from 0.797 to 1.390 in the basic setting, and from 0.758 to 1.430 in the complex setting, which is close to our results.

In the basic setting of [6], the differences between thresholds for all the paired directions for both slow and fast scales were significant. However, our results reveal no significant difference between vertical and depth directions in both slow and fast scales for all speed levels. This difference is likely due to our study's requirement to control hand speed, which is much more complex and difficult and users' sensitivity would decrease with the increase of task complexity. Besides, another possible reason for this difference could be that we required participants to follow a moving plate as closely as possible, control their hands precisely, and obverse the color of the plate in real-time (the color of the moving plate represents whether the relative speed between hand and plate is within an acceptable range). To meet these requirements, participants must be able to adjust the moving state of their hands in real-time according to the visual information, paying more attention to visual senses than haptic interaction, which may interfere with the sensitivity of participants to hand movement scaling. Moreover, due to the limitation of the field of view and arm extension length, the movement length we set on the X-axis is greater than that on the Y-axis and Z-axis, that is, the offset accumulated by scaling in each frame would be more noticeable on the X-axis. Thus, participants were more sensitive to movements along the X-axis, which reasonably explains that under the control of hand speed, the difference between the Y and Z axes is not significant, while the difference between the X-axis and other axes is significant.

4. Experiment 2: velocity adaptive imperceptible hand remapping

In earlier work, Frees and Kessler presented a velocity-based precision control technique for interactions in VR environments [9]. However, they did not determine the interval of hand speed to decelerate or accelerate in their experiment, which might not match users' intention of precise or rapid manipulation well. Besides, the scale value used in their work was from 0.33 to 1 in the generic mode, much lower than the detection thresholds for scaled hands, which might make users notice the manipulation and break the sense of presence. With the detection thresholds estimated in our first experiment, we aimed to devise an imperceptible velocity-based hand remapping technique for VR applications that supports both rapid and precise tasks. We fitted a velocity adaptive function to adjust the scale index based on real-time hand velocities. To evaluate the effectiveness of this enhancing technique, we conducted an experiment that required participants to complete tasks with different difficulty levels. We apply different techniques in each experiment block while the participant is completing the task. For data analysis, we logged the completion time and failure times for a trial as well as average movement distances.

4.1. Technique

Smoothed point technique [32] presented a velocity-based precision enhancing work, which aims to eliminate the jitter of handhold devices for remote points. The basic idea comes from Fitt's Law, which indicated that hand movement is slowed down for smaller target manipulating. However, this work didn't take detection thresholds into consideration, and users would perceive the speed gain. Our *velocity adaptive function* adopted a velocity-based C–D ratio adaptation to smooth the controller movement. It uses four-speed thresholds to determine the gain value: a minimum speed ($v_{\min}$) below which any motion is considered a tracking error; a minimum scaling constant ($SC_{\min}$), which is the right extreme of the hand velocity interval in which the hand movement is scaled down; and a maximum scaling constant ($SC_{\max}$), during which and $SC_{\min}$ the mapping remains one-to-one; and a maximum speed ($v_{\max}$), which is the hand speed at which the offset recovery is automatically triggered (see Figure 4). In fact, if a user's movement speed is between the scale interval, the gain scaling produces the accumulation of an offset between the device position in the motor space and the cursor position in the display space. This offset, the maximum value of which is

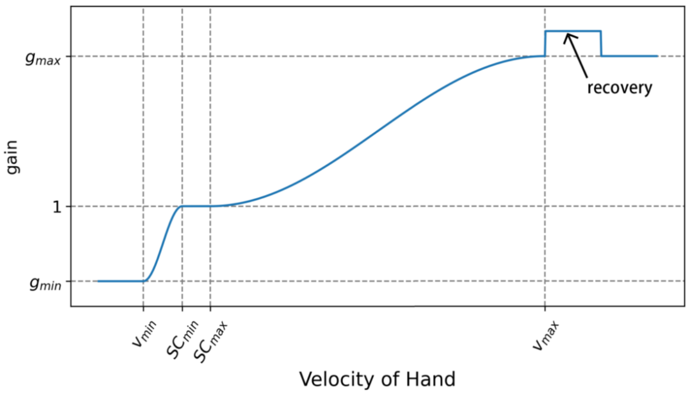

Figure 4: The hand remapping function uses speed to adjust the control-display ratio.

limited to $o_{\max}$, is recovered by setting the C–D gain to $g_{\max}$. To smooth the controller movement, in the $[v_{\min}, v_{\max}]$ interval, a modulated sine wave is used as the gain damping function.

We denoted the position of the virtual (displayed) hand and the real (tracked) hand as $p_{\text{vir}}(t)$ and $p_{\text{real}}(t)$. The move distance in the world space for each frame was denoted as $d(t)$. We also computed the offset between the virtual and real space, which is denoted as $o(t)$. In practice, we used the inverse of the HMD's frame rate (90 Hz), $\Delta t = 1/(90\text{ Hz})$.

Next, we computed several necessary normalized values. $\hat{v}_{\text{slow}}(t)$, the fast normalized velocity, was computed by diving the absolute value between velocity $v(t)$ and $v_{\min}$ by the $[v_{\min}, SC_{\min}]$ interval. $\hat{v}_{\text{fast}}(t)$, the fast normalized velocity, was computed by diving the absolute value between velocity $v(t)$ and $v_{\max}$ by the $[SC_{\max}, v_{\max}]$ interval. $\hat{o}(t)$, the normalized offset, was computed by dividing the absolute value of the offset $o(t)$ by the $o_{\max}$ value. Finally, we computed $\hat{m}(t)$, a hybrid parameter set as the maximum between the $\hat{o}(t)$ and $\hat{v}(t)$ values.

The $\hat{m}_{\text{slow}}$ and $\hat{m}_{\text{fast}}$ values are then used to derive the gain $g(t)$, which is given by the following equation

(4)

$$g(t) = \begin{cases} g_{\min} & v(t) < v_{\min} \\ g_{\min}+\frac{1}{2}(1-g_{\min}(\sin(\hat{m}_{\text{slow}}(t)\cdot\pi-\frac{\pi}{2}+1))) & v_{\min} \leq v(t) \leq SC_{\min} \\ 1 & SC_{\min} < v(t) < SC_{\max} \\ 1+\frac{1}{2}(g_{\max}-1(\sin(\hat{m}_{\text{fast}}(t)\cdot\pi-\frac{\pi}{2}+1))) & SC_{\max} \leq v(t) \leq v_{\max} \\ g_{\max}(t) & v_{\max} < v(t) \end{cases}$$

where

$$g_{\max}(t) = \begin{cases} \frac{o(t)}{d(t)} + \frac{1}{\hat{v}_{\text{fast}}(t)}(1 - \frac{o(t)}{d(t)}) & \text{if } o(t) \cdot d(t) > 0 \\ \frac{1}{\hat{v}_{\text{fast}}(t)(1+\hat{v}(t))} & \text{otherwise} \end{cases} \tag{5}$$

Finally, the $p_{\text{vir}}(t)$ value is given by

$$p_{\text{vir}}(t) = p_{\text{vir}}(t - \Delta t) + g(t) \cdot d_{\text{real}}(t) \tag{6}$$

During the process of interaction, due to the presence of scaling, the offset between virtual and real space gradually accumulates. So, it is essential to recover the offset when it exceeds the threshold. The offset recovery starts when the speed becomes larger than the $v_{\max}$ value. As shown in the equation (5), the gain on each side is set to $g_{\max}$ independently. For a velocity $v(t) \gg v_{\max}$, $p_{\text{vir}}(t)$ becomes equal to $p_{\text{real}}(t)$ in a short time, performing as a totally one-to-one pointing mode. When the hand movement speed is larger, the fraction of offset can be recovered faster. Besides, the offset would be directly eliminated when the positional offset is larger than the detection thresholds for the positional offset in [3].

4.2. Experiment design

To evaluate our remapping function in both precise and rapid phases, we designed a user study consisting of two phases. As proposed by reference [6], research related to motion detection should incorporate basic movements to establish more cautious thresholds for complex motion. To account for potential differences in cognitive load, we only cover simple test cases that assess the detection ability. The first phase involves a target selection and the second phase a trajectory-based linear movement. We compared three types of interaction methods: our *velocity adaptive function* (VA) method which adjusts the scale index depending on hand velocity, and a *one-to-one* (OTO) method in which the tracked hand and virtual hand are aligned, and a *constant scale* (CS) method in which the scale index was a different constant value in different phases of the task.

In the rapid target selection phase, users were asked to reach a virtual sphere at its initial position, pull the trigger button on the HTC Vive controller to pick the sphere up, and then use it to hit the target square piece which was placed at 0.4 meters away as fast as possible. Our remapping method was designed to scale the movement in three directions at the same time, so our experiment was designed to avoid moving the hand in three

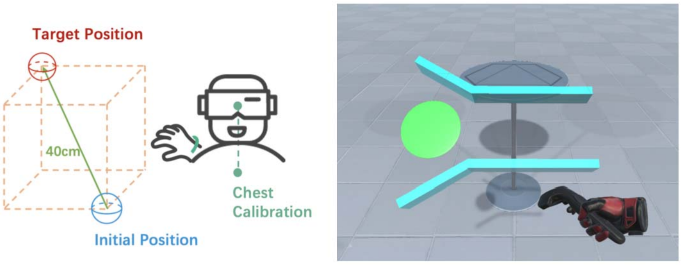

Figure 5: Two different speed mode tasks for the second experiment. (*Left*) shows the diagonally opposite corner target hit in the rapid task. (*Right*) shows the trajectory-based linear target hit in the slow task.

separate directions. The initial position and target position were set at the diagonally opposite corner of the cube, which would require users to move their hands across three directions (see Figure 5). Once the user finished, the starting sphere and target square piece would be changed to another position and the user would be asked to repeat the same process. As for the precise phase, we firstly designed a tube traversal task in which the tube was set along the diagonal of the cube (similar to the rapid task). However, because of perspective projection and overlap, failures in performing this task were mainly caused by visual issues (e.g. users not being able to see the tube clearly). In order to mitigate the visual impact, we adopted the following measures: we transformed the visual cue objects (the sphere to manipulate and the tube to limit movement) from 3D to 2.5D (the sphere changed to a round piece and the tube changed to a trajectory geometry with a shallow depth). We also discarded the depth axis since it was difficult for users to tell whether they were moving in the right direction due to occlusions. Besides, we set the angle between the plate where the round piece and trajectory are located and the ground plane to 45 degrees, keeping the objects completely within the field of view. In the trajectory-based linear movement phase, users were asked to move the round piece with a diameter of 0.295 meters through the trajectory with a width of 0.3 meters. Once a collision between the round piece and trajectory geometry happened, the round piece would be set back to the initial position. Since the error tolerance is only 0.005 meters, users were required to move the round piece very cautiously and slowly. The trajectory was positioned 30 cm below the HMD, approximately at the position of the user's chest to make it comfortable for participants to operate. It was placed at a 45 degree angle with the ground so that participants were able to see it when they bow their heads a little.

In each trial, both the rapid target selection phase and the trajectory-based linear movement phase would be conducted twice (the movement was along a diagonal back and forth for the rapid phase, and for the precise phase, it contained an X-axis and Y-axis movement). Every participant completed 70 trials (1 test × 10 trials + 3 methods × 2 repeats × 10 trials).

4.3. Participants

Voluntary participants were recruited from our campus and a total of 15 students including 11 males and 4 females participated in Experiment 2. Their age ranged from 22 to 29 with a median of 24.3 years. The majority of participants reported having used VR technology only a few times before or never at all and were only somewhat familiar with VR. Participants were physically and mentally healthy to perform the study. All participants completed the study with their right hand and reported they were comfortable with right-handed. The apparatus was the same as in the first experiment.

4.4. Procedure

Upon their arrival, participants read and signed informed consent. Afterward, we explained the goal and the whole process of the experiment. They would also fill in a demographic questionnaire and an SSQ [35] questionnaire. Then, the experimenter helped them to adjust and put on the HMD, as well as make sure the display was clear. Then, we instructed participants to move their heads around to explore the virtual environment and get familiar with the virtual hand movements using the controllers.

After the experimenter calibrated the distance between plates to ensure it was within arm's reach, participants proceeded to the first mode, determined by which counterbalanced group they were in. Before starting any trial, they would take a test block to practice which is the same as the formal block. In each block, participants were asked to complete a rapid target hit task and the other a trajectory-based linear movement task. The process was repeated until all trials for the method were finished. The other method would start following the same procedure. Participants were free to take a break between two trials and they were also reminded frequently to rest to reduce the effect of cybersickness. At the end of the experiment, participants filled out another SSQ questionnaire and a 5-question post-study questionnaire designed by ourselves. The entire procedure took about 30 minutes.

Table 3: Mean performance statistics for three methods

	VA	OTO	CS
Completion Time in Fast Phase	0.401	0.446	0.296
Completion Time in Slow Phase	2.746	3.878	4.590
Failure Times in Slow Phase	1.176	1.581	1.498

Table 4: Tukey multiple comparisons of completion time in the rapid task (left), completion time in the precise task (middle), and failure times in the precise task (right). If the confidence interval of the mean difference between two groups does not include zero, then it can be concluded that the means of these two groups are significantly different

	Completion Time in Fast Phase			Completion Time in Slow Phase			Failure Times in Slow Phase		
	VA	OTO	CS	VA	OTO	CS	VA	OTO	CS
VA	–	0.007**	0.000***	–	0.067	0.000*	–	0.003**	0.022
OTO	−0.007**	–	0.000***	−0.067	–	0.028***	−0.003**	–	0.817*
CS	−0.000***	−0.000***	–	−0.000*	−0.028***	–	−0.022	−0.817*	–

4.5. Results

4.5.1. Completion time and failure time

The central question we aimed to address was whether our *velocity adaptive function* was more effective than *direct manipulation* for both precision linear movement task and rapid target acquisition task. We listed the completion time in both phases and failure time for a precise phase in Table 3. We can see that *velocity adaptive function* technique was higher than the mean performance of direct manipulation. RM-ANOVA indicated significant differences of completion time for both rapid ($F = 38.53$, $p < 0.01$) and precise ($F = 11.5$, $p < 0.01$) tasks. There is also a significant difference in failure time in the precise task ($F = 5.977$, $p < 0.01$). We performed Tukey HSD multiple comparisons for post-hoc tests and the results are shown in Table 4.

4.5.2. Subjective rating

During the 5-question post questionnaire, the participants were asked to give their subjective rating on a 5-point Likert scale, as a score between 1 (strongly disagree) and 5 (strongly agree), in each block. Thus, the subjective rating analysis design is: 15 participants × 3 methods × 2 repeats × 5 questions = 450 data points. The results have been further analyzed with the Friedman test. These 5 questions are designed by us considering System Usability Scale (SUS) [38] as a reference.

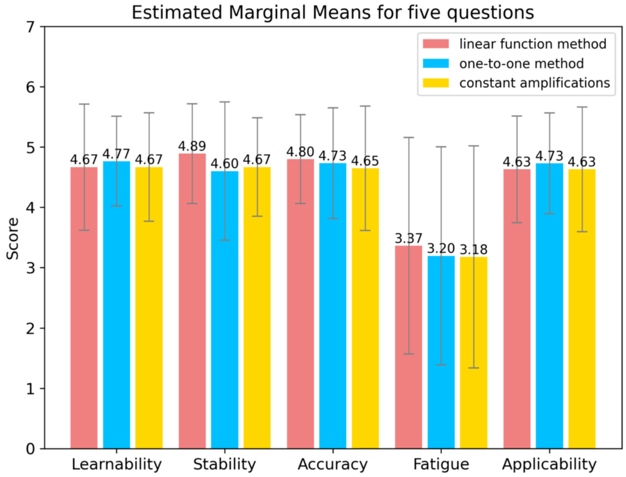

Figure 6: Overall scores of the *velocity adaptive function* method, the *one-to-one* method and the *constant amplification* method. Error bars show the standard deviation.

SUS is the most widely used standardized questionnaire to evaluate perceived usability. However, users spend too much time filling out the SUS questionnaire, which may affect user experience and the reliability of the questionnaire. We selected and adapted five questions from SUS, including five aspects (learnability, stability, accuracy, fatigue, and applicability) to measure the performance of the system. We design questions that can clearly and accurately express the issues we want to understand, capture all dimensions we are concerned about, and avoid repetition or confusion.

The 5 questions are as follows:

Q1: I would imagine that most people would learn to use this interaction technique very quickly.

Q2: I thought the interaction was relatively robust.

Q3: I thought I could accurately achieve my operation purpose when using this interaction technique.

Q4: I thought it made me very tired when I used this interaction technique.

Q5: I thought that I would like to use this interaction technique in other systems frequently.

The results show that participants have a high agreement with most of these questions for all three interaction techniques, except for Q3 related to fatigue (Figure 6). The results of the Friedman test show no significant

Table 5: Hand velocity percentage for precise (top) and rapid (bottom) phase in each method in 5 intervals according to Figure 4

Method	< 0.08	$[0.08, 0.15)$	$[0.15, 0.20)$	$[0.20, 0.80)$	≥ 0.80
OTO	0.720	0.161	0.046	0.067	0.002
VA	0.696	0.180	0.051	0.069	0.002
CS	0.675	0.176	0.057	0.085	0.005

Method	< 0.08	$[0.08, 0.15)$	$[0.15, 0.20)$	$[0.20, 0.80)$	≥ 0.80
OTO	0.115	0.085	0.060	0.618	0.118
VA	0.120	0.090	0.062	0.613	0.094
CS	0.108	0.084	0.056	0.609	0.140

differences between interaction techniques (with p-values 0.305, 0.108, 0.212, 0.122, and 0.584 for Q1, Q2, Q3, Q4, and Q5). In the follow-up interview, participants mentioned that they did not feel any obvious differences in each block while completing the task, which confirms that our method was indeed imperceptible to users.

4.5.3. Velocity distribution We logged the hand velocity of each frame along three directions and corresponding scale values for further analysis. According to the velocity adaptive function we proposed, we calculated the percentage of hand speed in 5 intervals according to Figure 4. As shown in Table 5, the distributions in each task and method were almost indistinguishable. Results show that in the precise task, participants' hand velocity was mostly below the interval $[0.15, 0.20)$, while in the rapid task, the velocity was mostly above the interval $[0.15, 0.20)$. Since our proposed method (VA) up-scaled hand movements for speeds larger than 0.20 and down-scaled them for speeds less than 0.15, meaning that most of the time the hand movements were under effective control. Also, there were significant differences in failure times between each method, meaning that VA could effectively reduce the number of errors people made and improve the manipulation accuracy. However, no significant difference was found in completion time between our method and the OTO method in the precise task. A possible reason for this is that if users move their hands completely free without setting any error conditions, it would take more time to move the same distance using our VA technique because the scale is less than one. However, for tasks that may cause errors, using a scale factor of less than 1 can reduce failures caused by accidental hand shaking, thereby reducing the time to complete the entire process as a whole and improving the efficiency of completing the task. In our experiment, the tolerance of failure for precise operation is set to a

Table 6: The threshold range in each direction and each level of velocity

Speed (m/s)	Axis	Scale Range
0.15	X	[0.852, 1.264]
	Y	[0.803, 1.212]
	Z	[0.782, 1.227]
0.32	X	[0.842, 1.365]
	Y	[0.792, 1.401]
	Z	[0.782, 1.347]
0.56	X	[0.837, 1.302]
	Y	[0.806, 1.396]
	Z	[0.784, 1.373]

small value. Using our proposed scaling method can greatly reduce the participants' failure cases caused by hand shaking, so as to reduce the number of failures when completing precise tasks and speed up the whole operation process. Even with close completion time, reducing failure times would alleviate the frustration caused by a failure in the precise task and improve user experience. For the rapid task, almost all the time scale factors were large than 1.0, which means that the time required to move the hand over the same distance would be shorter. Besides, the significant differences between our method and *constant amplification* method revealed that dynamic adjustment of scaled movement could improve interaction efficiency.

5. Discussion

We determined the interval of scale applied to virtual hand movement for which users can not detect the difference between the actual physical hand movement and the mapped virtual hand movement. The thresholds in each direction and under control of each level of velocity we estimated are shown in Table 6.

The results of our research provide new insights into human perception of scaled hand movements in VR when the hand movement speed is controlled. We detected significant differences between the detection thresholds of horizontal directions and the two other directions (vertical and depth), both for slow and fast scales. The range was narrower for the detecting hand movement scaling in the horizontal plane compared to the other two. This may be due to the visual factor as the visual field of view of the horizontal plane covers a larger range than the vertical [39], which means that motion in the horizontal direction provides a clearer view of object position. Due to the perspective of the human eye, it was more challenging to distinguish depth changes with cues of vergence and relative size. These factors mean that users are more sensitive to horizontal motion change, reflected in significant differences between detection thresholds of horizontal directions and two other directions.

In our second experiment, we used these estimated scale ranges and designed a modified interaction technique while aiming to provide a realistic, natural, and immersive experience for users in VR. It is beneficial to use slow-scaled hand motion situations to enhance the accuracy of the interaction. Therefore, slow-scaled hand movements can be used in such applications to provide more controlled hand movements in VR. On the other hand, fast-scaled hand movements could be useful in VR applications that require faster hand interactions or a far distant hand reaching. By applying fast-scaled hand movements, VR users could move their physical hands to reach objects at a larger range. However, constantly using fast-scaled or slow-scaled remapping would accumulate an offset. Once beyond the detection threshold of a constant offset, users would notice the inconsistency. Besides, another problem was that it was difficult to find a suitable usage scenario for consistently fast or slow scale, even in professional or specific applications (e.g., using VR for hand rehabilitation training [40], or medical training VR applications [41]) there were many situations where user do not need the slow downscale all the time. A real-time adjusting, context determined, imperceptible, and enhanced hand interaction technique was desired in the VR environment. Therefore, we proposed an interaction technique that inferred the application context according to the moving speed of the user's hand [42]: when users move their hand at a relatively low velocity, we determine they have a precise goal and applied a slow-scaled hand movement to enhance accuracy; conversely, at a somewhat higher velocity, we determine they have the rapid manipulation and accelerated their manipulation using a fast-scaled hand movement. By applying scales within the detected thresholds, we proved that the displacement of virtual hands would not distract users and was almost completely imperceptible. Thanks to the above characteristics of our proposed interaction technique, we can envision it can be applied in any situation in VR that either requires dedicated operation and favors accuracy, or that requires efficiency or a combination of the two.

6. Conclusion and future work

In this paper, we presented a velocity-adaptive and imperceptible hand remapping technique. Firstly, we estimated the detection thresholds along three axes at three speed levels. We did not find that speeds significantly influence the detection thresholds. Secondly, we designed a velocity-adaptive hand remapping function and compared it to other remapping techniques in tasks that included both precise and rapid movements. Results showed that our proposed technique significantly outperformed the other ones both in terms of efficiency and accuracy while remaining imperceptible.

Limitations in our work include that we only tested the detection thresholds at three different velocities and did not cover very fast velocities in order to control experiment time. Future work could focus on estimating detection thresholds for fast movements and adopting related remapping techniques. The limited range of test velocities may also be the reason why the detection thresholds were not significantly different between speed levels. Additionally, since our experiment only includes simple axis movements, thresholds for compound motion also need further investigation when considering velocity as a component. A personalized version of velocity-adaptive mapping could also provide a better user experience. Another limitation of our work is that we recruited a limited number of people to participate in experiment 1. Determining the detection threshold is a quantitative study. Testing with more users can offer a more reasonably tight confidence interval.

References

[1] Eric Burns, Sharif Razzaque, Abigail T. Panter, Mary C. Whitton, Matthew R. McCallus, and Frederick 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*, pages 3–10. IEEE, 2005.

[2] Francis B. Colavita. Human sensory dominance. *Perception & Psychophysics*, 16(2):409–412, 1974.

[3] Brett Benda, Shaghayegh Esmaeili, and Eric 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)*, pages 269–278. IEEE, 2020.

[4] André Zenner and Antonio Krüqer. Estimating detection thresholds for desktop-scale hand redirection in virtual reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 47–55. IEEE, 2019.

[5] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew 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*, pages 1968–1979, 2016.

[6] Shaghayegh Esmaeili, Brett Benda, and Eric D. Ragan. Detection of scaled hand interactions in virtual reality: The effects of motion direction and task complexity. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 453–462. IEEE, 2020.

[7] Christian T. Neth, Jan L. Souman, David Engel, Uwe Kloos, Heinrich H. Bulthoff, and Betty J. Mohler. Velocity-dependent dynamic curvature gain for redirected walking. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1041–1052, 2012.

[8] Valentin Holzwarth, Joy Gisler, Christian Hirt, and Andreas Kunz. Comparing the accuracy and precision of steamvr tracking 2.0 and oculus quest 2 in a room scale setup. 03 2021.

[9] Scott Frees and G. Drew Kessler. Precise and rapid interaction through scaled manipulation in immersive virtual environments. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005*, pages 99–106. IEEE, 2005.

[10] Victoria Interrante, Brian Ries, and Lee Anderson. Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments. In *2007 IEEE Symposium on 3D User Interfaces*. IEEE, 2007.

[11] Sharif Razzaque, Zachariah Kohn, and Mary C. Whitton. Redirected walking. In Jonathan C. Roberts, editor, *22nd Annual Conference of the European Association for Computer Graphics, Eurographics 2001 – Short Presentations*, Manchester, UK, September 3–7, 2001. Eurographics Association, 2001.

[12] Eike Langbehn, Joel Wittig, Nikolaos Katzakis, and Frank Steinicke. Turn your head half round: Vr rotation techniques for situations with physically limited turning angle. In *Proceedings of Mensch und Computer 2019*, pages 235–243. 2019.

[13] Shyam Prathish Sargunam, Kasra Rahimi Moghadam, Mohamed Suhail, and Eric D. Ragan. Guided head rotation and amplified head rotation: Evaluating semi-natural travel and viewing techniques in virtual reality. In *2017 IEEE Virtual Reality (VR)*, pages 19–28. IEEE, 2017.

[14] Eric D. Ragan, Siroberto Scerbo, Felipe Bacim, and Doug A. Bowman. Amplified head rotation in virtual reality and the effects on 3d search,

training transfer, and spatial orientation. *IEEE Transactions on Visualization and Computer Graphics*, 23(8):1880–1895, 2016.

[15] Travis Stebbins and Eric D. Ragan. Redirecting view rotation in immersive movies with washout filters. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 377–385. IEEE, 2019.

[16] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, pages 79–80, 1996.

[17] Sharif Razzaque, David Swapp, Mel Slater, Mary C. Whitton, and Anthony Steed. Redirected walking in place. In *EGVE*, volume 2, pages 123–130, 2002.

[18] Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz, and Markus Lappe. Analyses of human sensitivity to redirected walking. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, pages 149–156, 2008.

[19] Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz, and Markus Lappe. Estimation of detection thresholds for redirected walking techniques. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):17–27, 2009.

[20] Timofey Grechkin, Jerald Thomas, Mahdi Azmandian, Mark Bolas, and Evan Suma. Revisiting detection thresholds for redirected walking: Combining translation and curvature gains. In *Proceedings of the ACM Symposium on Applied Perception*, pages 113–120, 2016.

[21] Lionel Dominjon, Anatole Lécuyer, J.-M. Burkhardt, Paul Richard, and Simon Richir. Influence of control/display ratio on the perception of mass of manipulated objects in virtual environments. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005*, pages 19–25. IEEE, 2005.

[22] Tiare Feuchtner and Jörg Müller. Ownershift: Facilitating overhead interaction in virtual reality with an ownership-preserving hand space shift. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 31–43, 2018.

[23] Nami Ogawa, Takuji Narumi, and Michitaka Hirose. Effect of avatar appearance on detection thresholds for remapped hand movements. *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[24] Roberto A. Montano Murillo, Sriram Subramanian, and Diego Martinez Plasencia. Erg-o: ergonomic optimization of immersive virtual environments. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 759–771, 2017.

[25] Johann Wentzel, Greg d'Eon, and Daniel Vogel. Improving virtual reality ergonomics through reach-bounded non-linear input amplification. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[26] Luv Kohli, Mary C. Whitton, and Frederick P. Brooks. Redirected touching: The effect of warping space on task performance. In *2012 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 105–112. IEEE, 2012.

[27] Joanna Bergström, Aske Mottelson, and Jarrod Knibbe. Resized grasping in vr: Estimating thresholds for object discrimination. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 1175–1183, 2019.

[28] Jialei Li, Isaac Cho, and Zachary Wartell. Evaluation of 3d virtual cursor offset techniques for navigation tasks in a multi-display virtual environment. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 59–66. IEEE, 2015.

[29] Jialei Li, Isaac Cho, and Zachary Wartell. Evaluation of cursor offset on 3d selection in vr. In *Proceedings of the Symposium on Spatial User Interaction*, pages 120–129, 2018.

[30] Songhai Zhang, Chen Wang, Yizhuo Zhang, Fang-Lue Zhang, Nadia Pantidi, and Shi-Min Hu. Velocity guided amplification of view rotation for seated vr scene exploration. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 504–505. IEEE, 2021.

[31] Chen Wang, Song-Hai Zhang, Yi-Zhuo Zhang, Stefanie Zollmann, and Shi-Min Hu. On rotation gains within and beyond perceptual limitations for seated vr. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[32] Luigi Gallo, Mario Ciampi, and Aniello Minutolo. Smoothed pointing: a user-friendly technique for precision enhanced remote pointing. In *2010 International Conference on Complex, Intelligent and Software Intensive Systems*, pages 712–717. IEEE, 2010.

[33] Luigi Gallo and Aniello Minutolo. Design and comparative evaluation of smoothed pointing: A velocity-oriented remote pointing enhancement technique. *International Journal of Human-Computer Studies*, 70(4):287–300, 2012.

[34] Werner A. König, Jens Gerken, Stefan Dierdorf, and Harald Reiterer. Adaptive pointing–design and evaluation of a precision enhancing technique for absolute pointing devices. In *IFIP Conference on Human-Computer Interaction*, pages 658–671. Springer, 2009.

[35] Robert S. Kennedy, Norman E. Lane, Kevin S. Berbaum, and Michael G Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, 1993.

[36] Daniel Linares and Joan López-Moliner. quickpsy: An r package to fit psychometric functions for multiple groups. *The R Journal*, 8(1):122–131, 2016.

[37] Kenneth Knoblauch and Laurence T. Maloney. *Modeling psychophysical data in R*, volume 32. Springer Science & Business Media, 2012.

[38] John Brooke. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189:11, 1995.

[39] Harry Moss Traquair. An introduction to clinical perimetry, Chap. 1. *London: Henry Kimpton*, pages 4–5, 1938.

[40] Lihan Chen, Xiaolin Zhou, Hermann J Müller, and Zhuanghua Shi. What you see depends on what you hear: Temporal averaging and crossmodal integration. *Journal of Experimental Psychology: General*, 147(12):1851, 2018.

[41] David Ota, Bowen Loftin, Tim Saito, Robert Lea, and James Keller. Virtual reality in surgical education. *Computers in Biology and Medicine*, 25(2):127–137, 1995.

[42] I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Hum.-Comput. Interact.*, 7(1):91–139, Mar 1992.

Yike Li
Department of Computer Science and Technology
Tsinghua University
China
E-mail address: liyike0206@gmail.com

Chen Wang
Department of Computer Science and Technology
Tsinghua University
China
E-mail address: cw.chenwang@outlook.com

Ge Yu
Technology and Engineering Center for Space Utilization
Chinese Academy of Sciences
China
E-mail address: yuge@csu.ac.cn

Yu He
Beijing Institute of Mathematical Sciences and Applications
China
E-mail address: hooyeeevan2511@gmail.com

Stefanie Zollmann
Department of Computer Science
University of Otago
New Zealand
E-mail address: stefanie.zollmann@otago.ac.nz

Song-Hai Zhang
Department of Computer Science and Technology
Tsinghua University
China
E-mail address: shz@tsinghua.edu.cn

Received February 16, 2023