International Journal of Human–Computer Interaction

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/hihc20

VVIR-OM: Efficient Object Manipulation in VR with Variable Virtual Interaction Region

Qinwen Zheng, Lili Wang, Wei Ke & Sio Kei Im

To cite this article: Qinwen Zheng, Lili Wang, Wei Ke & Sio Kei Im (27 Nov 2023): VVIR-OM: Efficient Object Manipulation in VR with Variable Virtual Interaction Region, International Journal of Human–Computer Interaction, DOI: 10.1080/10447318.2023.2286126

To link to this article: https://doi.org/10.1080/10447318.2023.2286126

Published online: 27 Nov 2023.

INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION
https://doi.org/10.1080/10447318.2023.2286126

VVIR-OM: Efficient Object Manipulation in VR with Variable Virtual Interaction Region

Qinwen Zheng[a], Lili Wang[a,b], Wei Ke[c], and Sio Kei Im[c]

[a]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China; [b]Peng Cheng Laboratory, Shengzhen, China; [c]Faculty of Applied Sciences, Macao Polytechnic University, Maco, China

ABSTRACT

Manipulating with virtual objects is a fundamental requirement in virtual environments. A good manipulation method needs to consider efficiency, accuracy, and comfort. This paper proposes VVIR-OM, an object manipulation method in virtual reality (VR) based on a variable virtual interaction region (VVIR). A hand interaction hemisphere region (HIHR) is introduced and constructed in real space, where the user is more comfortable manipulating the objects. Then a VVIR is introduced, and an interaction heat volume (IHV) based method is proposed to update VVIR during the process of the manipulation. At last, a mapping algorithm is proposed to map the user hand position in HIHR to the position in VVIR. Two user studies are designed to evaluate the performance of VVIR-OM. Compared to the state-of-the-art methods, VVIR-OM achieves significant improvements in task completion time, manipulation precision, and a significant reduction in fatigue. Moreover, VVIR-OM outperforms other methods in terms of task load and usability without the cost of cybersickness.

KEYWORDS

Virtual reality; object manipulation; variable virtual interaction region; interaction heat volume

1. Introduction

Virtual object manipulation is one of the fundamental interactions in VR applications. Much previous research has been devoted to designing efficient and accurate object manipulation.

Techniques focused on manipulation efficiency typically adjust the control-to-display ratio based on hand motion parameters, allowing users to achieve large object movements with small hand movements (Bowman & Hodges, 1997; Frees & Kessler, 2005; Frees et al., 2007; Grandi et al., 2019; H. Kim et al., 2015). However, the dynamically changing mapping ratios may affect the stability of the manipulation, thereby influencing the precision and comfort of the manipulation (H. Kim et al., 2015). Techniques emphasizing manipulation precision, such as multi-point interaction (Aguerreche et al., 2009; Gloumeau et al., 2020; Nguyen & Duval, 2013), allow users to specify manipulation points to constrain the manipulation axes, thus enhancing precision. But these techniques are often inefficient and unsuitable for distant object manipulation. Techniques prioritizing manipulation comfort mainly include extending the user's arm (Lisle et al., 2022; Poupyrev et al., 1996) and using virtual proxies (Pierce et al., 1999; Pohl et al., 2021). While extending the user's arm to aligns with the user's intuition is inefficient to manipulate and prone to arm fatigue (Bowman & Hodges, 1997). Virtual proxies, including object proxies and space proxies, require significant time for proxy selection and noticeably disrupt the user's sense of immersion. The efficiency, accuracy, and comfort of object manipulation methods still need to be improved.

In this paper, we propose VVIR-OM, an object manipulation method in VR based on a VVIR. First, we introduce and construct a HIHR in the real space, in which the user is more comfortable manipulating the objects. Second, we introduce a variable virtual interaction region and propose an IHV-based method to update VVIR during the process of the manipulation. Then, we propose a mapping algorithm to map the user hand position in HIHR to the position in VVIR. Finally, We designed two user studies to evaluate the performance of VVIR-OM. Compared to the state-of-the-art methods, VVIR-OM demonstrates significant improvements in task completion time, manipulation precision, and a significant reduction in fatigue. Moreover, VVIR-OM outperforms other methods in terms of task load and usability without the cost of cybersickness. Figure 1 shows the object manipulation process of VVIR-OM.

In summary, the main contributions of our work include:

- introducing a new object manipulation method, VVIR-OM, for manipulating the virtual object efficiently and accurately;
- introducing and constructing a hand interaction hemisphere region in real space;
- introducing a variable virtual interaction region and proposing an interaction heat volume-based method to update it during the manipulation;

CONTACT Lili Wang wanglily@buaa.edu.cn State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China; Peng Cheng Laboratory, Shengzhen, China

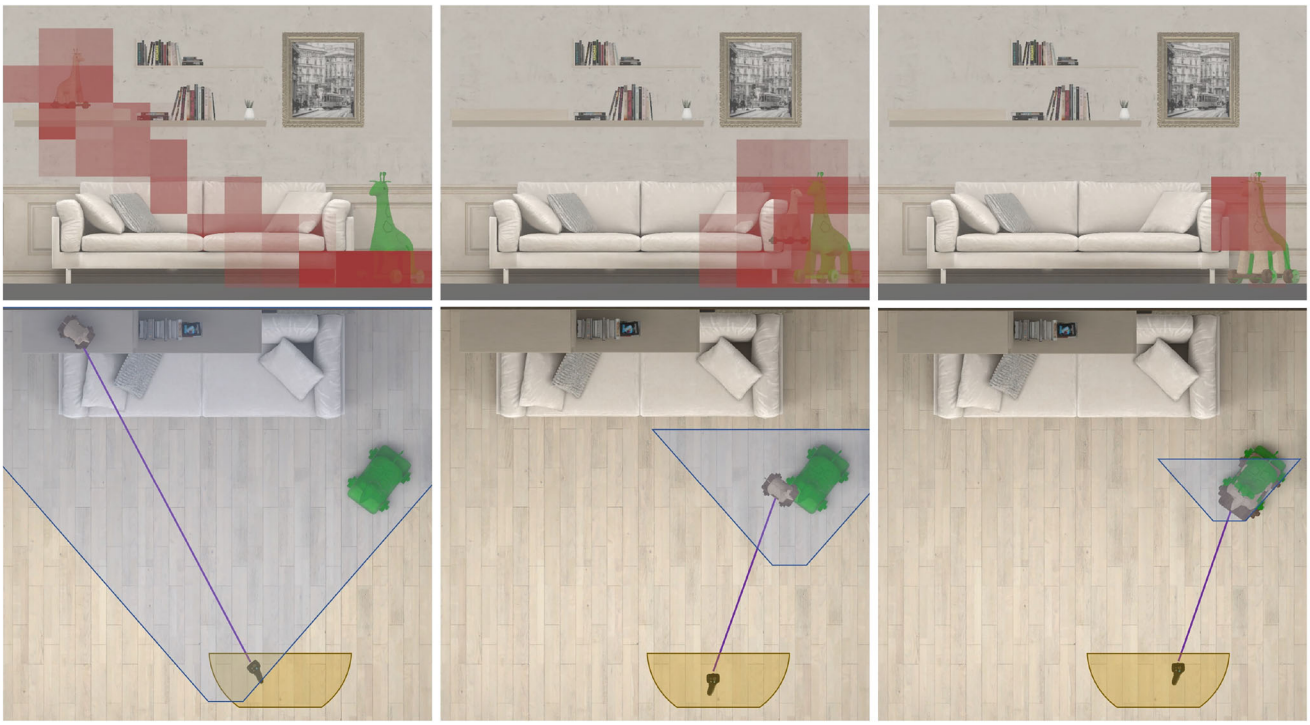

Figure 1. The manipulation process of VVIR-OM. Interaction heat volume is visualized with red from the user viewpoint in the top row, and the corresponding variable virtual interaction region is visualized with blue in the top view of the scene in the bottom row. The constructed hand interaction hemisphere region in real space is marked with yellow. The user manipulates the giraffe on the shelf on the left wall to reach the target location indicated by the green giraffe on the right floor during the coarse phase (column 1). the interaction heat volume at the top of column 1 indicates how often the user gazes at different locations in the scene during manipulation, with darker red representing higher gazing frequency. The user gaze area becomes more focused, indicating that the manipulated giraffe is close to the target. Therefore, the manipulation moves from the coarse phase to the fine phase. The variable virtual interaction region becomes smaller (column 2). after entering the fine phase, the position of the user's hand in the virtual space is used to update the interaction heat volume. As the heat area becomes more concentrated, indicating that the manipulation is moving from the fine phase to the enhanced fine phase, the variable virtual interaction region becomes further smaller, and the final fine-tuning of the object is performed (column 3).

- designing user studies to evaluate the efficiency and accuracy of VVIR-OM.

2. Related work

In this section, we briefly review the previous work of object manipulation in VR, which includes hand-based manipulation for distant objects, multiple modal and low fatigue manipulations. For a more comprehensive review of the manipulation methods, we recommend readers read the surveys (Abtahi et al., 2022; Mendes et al., 2019).

2.1. Distant object manipulation

One of the challenges in VR interaction is how users can interact with distance objects. Go-Go (Poupyrev et al., 1996) was an early representative technology that extended users' arms in the virtual environment to select and manipulate distant objects. Building upon the evaluation of Go-Go, HOMER (Bowman & Hodges, 1997) was introduced, which utilized ray casting for object selection. Once selected, a virtual hand would move to the object, establishing mappings based on the distance between the user's body-real hand and the user-virtual object. By adjusting the scaling factor, HOMER aimed to improve manipulational efficiency. To further enhance speed, a non-linear mapping method (H. Kim et al., 2015) was proposed, which dynamically determined the mapping ratio based on hand velocity and acceleration, without setting an upper limit on the mapping ratio. In terms of precision, some researchers have proposed the PRISM method (Frees & Kessler, 2005; Frees et al., 2007), which divided manipulations into direct mode and precise mode. In the direct mode, hand movements were proportionally mapped to virtual objects, while in the precise mode, the mapping ratio was reduced based on hand velocity, reducing the sensitivity of virtual objects to hand movements. Recently, the MGF method (X. Liu et al., 2023) was proposed to improve the accuracy and efficiency of the manipulation by directing the user to different manipulation viewpoints, but the method requires the target state of the virtual object to be known, which has some limitations in application.

Another approach is to bring distant objects or spaces closer to the user's body, transforming long-distance interactions into short-distance interactions. The Voodoo Dolls technique (Pierce et al., 1999) pioneered the research on virtual object proxies. After selecting an object, a synchronized replica appeared in front of the user, allowing them to manipulate distant objects using both hands. The Poros technique (Pohl et al., 2021) expanded the research on virtual object proxies, which allowed users to reorganize the space in front of them. Marked distant spaces would be mirrored in front of the user, enabling close-range interactions. On the other hand, the Diorama technique (M. G. Kim et al., 2022) focuses on virtual object sizing and creates

proxies by estimating the optimal size, thus improving the efficiency and convenience of proxy-based manipulation.

2.2. Multimodal object manipulation

Multimodality is a new research direction in virtual object manipulation. Our main focus is on the eye and head modalities, while information about other modalities can be found in the review (Monteiro et al., 2021). In the field of eye-based interaction, an early method of hand-eye coordination (Slambekova et al., 2012) was proposed, where users would gaze at virtual objects and then manipulate them using their hands. Gaze + Pinch technique (Pfeuffer et al., 2017) further explored the combination of hand and eye interaction, which still follows the gaze selection and hand manipulation approach but introduces gesture-based object manipulation. The Gaze + RST technique (Turner et al., 2015) was introduced for operating in 3D virtual space, where the gaze point is used for translating objects, and they introduced the concept of a display mask to transition between gaze and hand interaction. The OrthoGaze technique (C. Liu et al., 2020) was proposed, which allows users to interact solely with their eyes or head gaze, using three orthogonal planes to assist with object movement. A recent study by Yu et al. (2021) extended the Gaze + RST to the VR environment, integrating eye-based interaction for the first time. They introduced Implicit Gaze, a technique where users gaze-select objects and perform rotational movements, followed by using their hands for the remaining manipulation. Implicit Gaze improved on Turner's mask by introducing a dynamically changing gaze safety zone, enabling seamless transitions between modalities.

2.3. Low fatigue object manipulation

We focus on research related to arm fatigue and manipulation techniques. In terms of arm fatigue, RULA (McAtamney & Corlett, 1993) introduced a posture rating system, indicating that a relaxed upper arm with a slight bend at the elbow is the most comfortable posture. Another study (Hincapié-Ramos et al., 2014) also found that bending the elbow at the waist level reduces the interaction burden. Some researchers (Bachynskyi et al., 2015) conducted evaluations using electromyography and recommended performing short- to medium-distance movements in the reachable space. There is relatively less research on operating techniques related to the interaction area. The Erg-o technique (Montano Murillo et al., 2017) was proposed to reduce arm fatigue by repositioning the virtual hand, which solves the problem of arm fatigue caused by large-scale limb movements. The OwnerShift technique (Feuchtner & Müller, 2018) tackled the problem of arm fatigue during overhead interaction. This technique maintains the mapping of the virtual hand's position during the manipulation, allowing the real hand to gradually move to a more comfortable area. The Thmuble technique (Lim et al., 2022) introduces a new wearable input device that controls the rotation of virtual objects based on thumb movements, reducing hand fatigue during rotation manipulations. These techniques are only applicable to close-range VR interactions and have not addressed single-hand interaction space.

In contrast to these techniques, VVIR-OM restricts the user's arm movements within a comfortable region while enabling the manipulation of virtual objects throughout the visible space. VVIR-OM use multiple interaction modalities and utilize interaction heat volume to capture and process modality inputs. By analyzing the user's intentions to manipulate and the primary manipulation space, VVIR-OM modifies the mapping relationship between hand movements and object manipulation. This allows for efficient, accurate, and comfortable distance object manipulation.

3. Variable virtual interaction region implementation details

VVIR-OM uses the ray-based method to point and press the button 'Trigger' on the controller to select the object that needs to be manipulated. Then VVIR-OM attaches the virtual hand to the manipulated object using the method in Grandi et al. (2019) as a basic hand manipulation method. In addition, VVIR-OM divides the manipulation into three phases, coarse phase, fine phase, and enhanced fine phase, which are transparent to the user.

The coarse phase is the initial phase of manipulation, where the user can quickly move the virtual object using the controller. The object can be placed anywhere in the viewport space and the precision of this phase is relatively low. In the fine phase, the object is usually moved near the target location, the object's moveable space is reduced, and manipulation becomes more precise. In the enhanced fine phase, the object's moveable space is further reduced, allowing the user to make final adjustments. If the user makes a mistake, they can revert to the previous phase by pressing the button "B." During manipulation, the user is required to gaze at the manipulated object to perform fine manipulation.

3.1. Hand interaction hemisphere region

3.1.1. Definition

The HIHR refers to the comfortable single-hand active region when the user keeps static, i.e., his/her body position and orientation are unchanged. Previous research has indicated that the most comfortable hand movements occur when the upper arm is naturally hanging down and the elbow is bent at the waist (Bachynskyi et al., 2015; Hincapié-Ramos et al., 2014; McAtamney & Corlett, 1993). Therefore, our research focuses on hand interaction in this posture.

We define HIHR as a flat truncated hemisphere with 4 parameters $H(a, o, r, d)$ (marked with yellow in Figure 2 left), where a is the anchor point, o represents the center, r represents the radius of HIHR, and d is the depth of a flat truncated hemisphere.

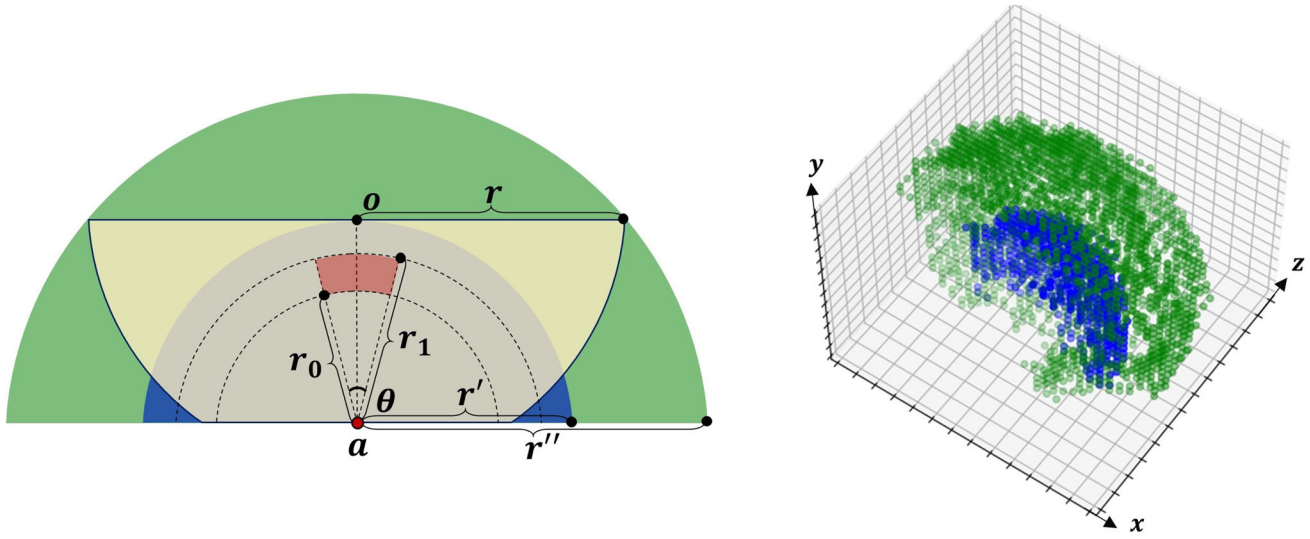

Figure 2. The construction of HIHR.

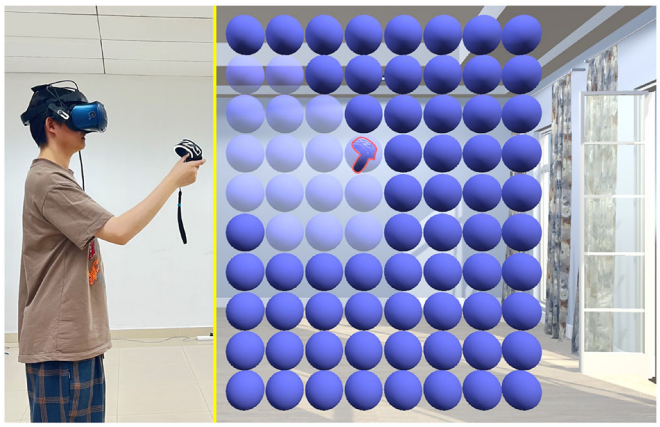

Figure 3. The pilot user study. The real world (left) and the virtual world (right). the eliminated ball will become transparent.

3.1.2. HIHR construction

To build HIHR $H(a, o, r, d)$, two hemispheres are constructed first. The first hemisphere $HC(o', r', dir')$ is the single-hand comfortable interaction region, marked with blue in Figure 2 left. The second hemisphere $HS(o'', r'')$ is the single-hand stretched interaction region (marked with green in Figure 2 left). o' and o'' are the centers, and r' and r'' are the radius of HC and HS. dir' is the orientation of HC. o' is determined by the position of the user's elbow, o'' is determined by the position of the user's shoulder. These arm parameters are captured by requiring the user to keep the arm stationary and touch the shoulder, elbow, wrist, and fingers in turn with the controller. dir' is the orientation of the user's body. We designed a pilot user study to obtain the r' and r'' of HC and HS.

Twenty participants between the ages of 20 and 40 were recruited from our university to attend this pilot user study. In the pilot user study, a lot of small balls are generated in front of the users, and the participants are required to eliminate all balls they can touch with their controllers (see Figure 3). We collected controller position data when the participants eliminate the balls. In session 1, participants performed the task with their arms naturally bent, i.e., the upper arm is close to the body, and the lower arm and hand move freely. The positions of the controller are visualized with blue points (see Figure 2 right), which indicates the single-hand comfortable interaction region. In session 2, participants performed the task with their arms stretched. The positions of the controller are visualized with green points (see Figure 2 right), which indicates the single-hand stretched interaction region. We used the least squares method according to the points to obtain r' and r''.

After this, HIHR $H(a, o, r, d)$ is determined by using HC and HS. a is set to the center o' of HC, d is set to the radius r' of HC, and o is computed according to a, d and the direction dir' of HC. We construct a plane perpendicular to dir' through the center o of H and intersect HS, and use the distance from the intersection to the center o as r. The reason we use the inverse flat truncated sphere as HIHR instead of the two hemispheres mentioned above is that the camera view is a flat truncated head, and this shape of HIHR reduces the distortion of the mapping of the two regions.

We also introduce a reset region marked with red inside HIHR (Figure 2 left), which is built according to two radii r_0 and r_1, and angle θ. r_0 is the length from the elbow position to the wrist, and r_1 is the length from the elbow to the finger. we set θ as a constant and use 30° in our implementation.

3.2. Variable virtual interaction region

3.2.1. Definition

VVIR refers to the region within the immersive virtual environment where virtual objects can be interacted with, and the size of the region is variable as the user performs manipulations.

We define VVIR as a frustum with 5 parameters $V(a, \alpha, t, dir, d)$, where a is the anchor point, α is the angle in the vertical direction, t is the ratio between the width and height of the frustum, dir is the orientation, and d is the depth of the frustum.

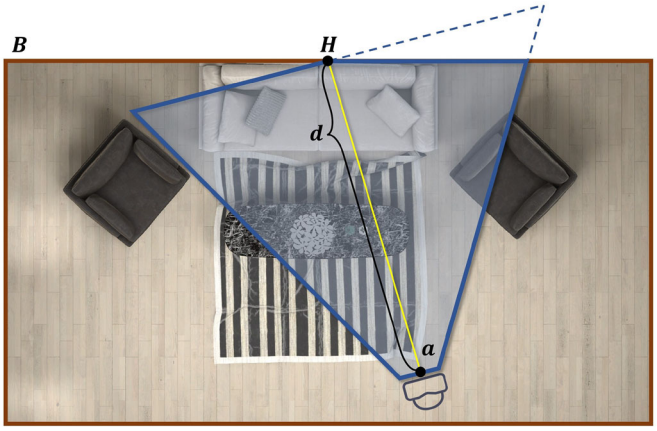

Figure 4. VVIR initialization.

3.2.2. VVIR initialization

We use the parameters of the user view frustum to initialize a, α, t and dir of VVIR V. We don't use the far clip plane distance of the user view frustum for the depth d of VVIR because that depth is usually much larger than the visible scene, while the object manipulations usually occur in the visible region. To obtain d, we create an AABB bounding box B for the virtual scene (red frame in Figure 4) and obtain the intersection H by intersecting the bounding box B using the centerline of VVIR. Then, d is the distance from the anchor point a to H. After this, we truncate the constructed frustum with B, and get the final VVIR (the region inside the blue solid line in Figure 4). VVIR will be updated during the manipulation based on the interaction heat volume, so we introduce IHV before giving the description of the VVIR updating algorithm.

3.2.3. Interaction heat volume

The IHV is a three-dimensional voxel array that dynamically records user interactions in the immersive virtual environment.

We voxelized the interactive region in the virtual scene. The voxel can be represented as $vox(s, h, st, f)$, where s is the size of vox, h is the heat value and st is the status. In our implementation, we set s to 0.02 meters, and initialize h as 0. st is a boolean, 1 for active and 0 for inactive, and is initialed as 0. f is the lifespan, which will be reset to 1 second when st is set to 1.

The parameters of vox are updated during the process of manipulation. At each frame during the manipulation, we can determine the voxel vox in virtual space where the user interaction position is located, with the state st set to 1, the heat value h is increased by 1, and the lifespan f is reset, giving a lifetime of 1 second. The lifespan of the voxel decreases over time. When the lifespan is 0, the voxel's heat value h starts decreasing by 1 per frame. When the heat value h is 0, the voxel returns to the inactive state.

In VVIR-OM, IHV can be updated by two modes of user interaction, gaze or hand. When IHV is applied to the gaze, for each frame, we first obtain the gaze point, calculate the voxel to which the gaze point belongs, and update the voxel. Since when we gaze at an object, the gaze point often falls on the background behind the object due to the unstable gaze point acquisition, which leads to depth errors, we introduce the ghost plane (GP), which is an invisible circular plane centered at the center of the object with a viewing angle of 10 degrees and perpendicular to the viewing direction. When a user gazes at an object, we are able to capture the gaze points around the object at similar depths by using GP to update the corresponding voxels more accurately. When applying IHV to the hand, the point of the virtual hand is obtained at each frame, and subsequently, the voxel where the virtual hand is located is updated in the same way as in gaze mode.

3.2.4. IHV-based VVIR updating

In this section, We use IHV to determine if the manipulation phase needs to be switched and to update the parameters of VVIR.

At the beginning of manipulation (see Figure 1 column 1), the user's manipulation is in the coarse phase. During the coarse phase, IHV is updated according to the gaze of the user. Firstly, we calculate the interaction center and dispersion of IHV. Secondly, the interaction deviation distance between the interaction center and the center of the virtual hand is calculated with Algorithm 1. If the interaction deviation distance is under the threshold of 0.2 meters, we set a flag as condition 1. Thirdly, we record the series of interaction dispersions for the last 20 frames, using the Augmented Dickey-Fuller test (ADF) (Cheung & Lai, 1995) to judge whether the interaction is stable, as condition 2. After these, the manipulation phase is switched from coarse to fine when the two conditions are satisfied, and then we update the anchor point a and depth d of VVIR with Algorithm 2.

At the fine manipulation phase (see Figure 1 column 2), IHV is updated by the hand interaction of the user. We also record the series of interaction dispersions for the last 20 frames, using the ADF test to determine whether the interaction is stable as the condition. Then, the manipulation phase is switched from fine to enhance fine (see Figure 1 column 3) when the condition is satisfied, and the anchor point a and depth d of VVIR are updated again with Algorithm 2.

Algorithm 1 shows the computation of the interaction center (lines 1–9) and dispersion (lines 10–14). The input to the algorithm is the IHV I. The outputs include the interaction center P and the interaction dispersion D.

Algorithm 1 IHV interaction center and dispersion computation

Input: IHV I
Output: interaction center P, interaction dispersion D
1: $VL \leftarrow$ voxels visited in the last 20 frames in I; $W \leftarrow []$
2: **for** vox in VL **do**
3: W.add(vox, $vox.h$)
4: **end for**
5: $P_s \leftarrow (0, 0, 0)$
6: **for** vox in W **do**
7: $P_s \leftarrow P_s +$ position(vox)

8: **end for**
9: $P \leftarrow P_s / W.size$
10: $D_s \leftarrow 0$
11: **for** *vox* in W **do**
12: $\quad D_s \leftarrow D_s +$ distance(*vox*, $P)^2$
13: **end for**
14: $D \leftarrow$ sqrt$(D_s/(W.size - 1))$

The algorithm obtains a series of voxels VL that have been touched and are active for the last 20 frames and creates an empty list W (line 1). Then, the algorithm iterates over each *vox* in VL (line 2), adds *vox* to W, and takes the value of *vox.h* as the number of additions (lines 3–4). Next, the algorithm creates a zero vector P_s (line 5), iterates through each *vox* in W (line 6), and adds the position of *vox* into P_s (line 7–8). Then the algorithm calculates the average position as interaction center P (line 9). After this, the algorithm creates D_s and set it to 0 (line 10), iterates through each *vox* in W (line 11), and calculates the square of the distance between the position of *vox* and P and accumulates it to D_s (lines 12–13). Finally, the algorithm calculates the standard deviation based on D_s as the interaction dispersion D (line 14).

Algorithm 2 is to update VVIR based on IHV. The depth d (lines 1–7) and the anchor point a are updated (lines 8–10). The inputs include VVIR V, HIHR H, IHV I, the controller C, the virtual hand O, and the current manipulation phase M. The output is the updated VVIR V.

Algorithm 2 VVIR updating

Input: VVIR V, HIHR H, IHV I, controller C, virtual hand O, current manipulation phase M
Output: the updated VVIR V
1: **if** M is switching to fine phase **then**
2: $\quad V.d \leftarrow H.d$
3: **end if**
4: **if** M is switching to enhanced fine phase **then**
5: $\quad VL \leftarrow$ voxels visited in the last 20 frames in I;
6: $\quad V.d \leftarrow$ FittedVoxels(VL)
7: **end if**
8: $R \leftarrow$ Normalization(H,C)
9: $F \leftarrow$ OffsetInVVIR(V,R)
10: $V.a \leftarrow$ position(O)$- F$

When the manipulation phase M is switching to the fine phase (line 1), the algorithm updates the depth d of V to the depth d of H (lines 2–3). When the manipulation phase M is switching to the enhanced fine phase (line 4), the algorithm obtains a series of voxels VL that have been touched and are active for the last 20 frames at first(line 5). Then the user's interaction region can be further reduced according to the region where the virtual hand O has been active in recent times, so the algorithm uses the least squares method to fit the voxels in VL and compute the minimum sphere, and set the depth d of V to the diameter of this sphere (lines 6–7). In order to avoid sudden changes in the position of the virtual hand when updating the anchor point a of V, the algorithm normalizes the position of the controller C in the local coordinate system of H to obtain the normalized vector R (line 8), and then, based on the normalized vector R, the offset of the anchor point a of V relative to the virtual hand O on each of the three axes as the offset vector F (line 9). Finally, the algorithm subtracts the position of O from the offset vector F to get the new anchor position of VVIR (line 10).

3.3. Mapping from HIHR to VVIR

3.3.1. Mapping

Once the HIHR and VVIR are constructed, a mapping algorithm is required to associate the motion of the controller with the movement of the object. We design Algorithm 3 to solve it. The algorithm takes VVIR V, HIHR H, the controller C, and the bounding box B as inputs, and outputs the position P_o of the virtual hand.

Algorithm 3 Mapping from HIHR to VVIR

Input: VVIR V, HIHR H, controller C, bounding box B
Output: virtual hand position P_o
1: $R \leftarrow$ Normalization(H, C)
2: $F \leftarrow$ OffsetInVVIR(V, R)
3: $P_f \leftarrow V.a + F$
4: **if** $!(P_i \leftarrow$ Intersect$(V, P_f, B))$ **then**
5: $\quad P_o \leftarrow P_i$
6: **else**
7: $\quad P_o \leftarrow P_f$
8: **end if**

First, the algorithm gets the normalized vector R (line 1) and the offset vector F (line 2) same as the Algorithm 2 (lines 8–9). Then the algorithm adds the offset vector F to the anchor point a of VVIR V, to obtain the original mapping position P_f of the virtual hand (line 3). Then, the algorithm detects whether the line between the anchor point a of V and P_f intersects the bounding box B (line 4). If it does, the intersection point P_i is used as the new position P_o of the virtual hand (line 5), otherwise P_f is used as the new position P_o of the virtual hand (lines 6–8). Thus, the virtual hand is avoided to be mapped outside the scene.

3.3.2. Reset

When users place the virtual object near the target location, it's possible that their hand is already at the edge of HIHR. In such cases, the actual available space for hand movement is limited, and the arm may have deviated from a comfortable posture. To address this issue, we propose a controller reset interaction. When the phrase switches, we test whether the controller needs to reset. If the controller is not inside the reset region, we reset it by temporarily interrupting the mapping, i.e., the virtual hand freeze when the controller moves. After the control goes back to the reset region, the virtual hand is unfrozen.

3.3.3. Cross-viewport manipulation

When a user selects an object, the target position may be outside their viewport. If a VVIR is constructed at this moment, the object cannot be moved to the target position. We introduce a head movement phase after object selection to support cross-viewport manipulation. After the user presses the button to select the object, they do not release the button. The VVIR will continuously update with the user's head movement, always facing forward, until the user sees the target and releases the button. During this process, the mapping algorithm remains active, allowing the object to move along with the updated VVIR. Throughout the entire process, we do not add any additional buttons, making it conform to the natural flow of user interactions.

4. User study 1: Single object manipulation

We designed a user study consisting of six manipulation sessions to evaluate the efficiency, accuracy, and fatigue of VVIR-OM.

4.1. Study design

4.1.1. Participants

We recruited a total of 32 participants at our university, including 17 males and 15 females, aged between 20 and 40 years. Among the participants, 13 had previous experience using VR applications. All participants were right-handed, and had normal or corrected vision, with no visual or balance impairments. Additionally, all participants have signed the informed consent form for our studies. Each participant is required to use five manipulation techniques to complete a single object manipulation task. CC_1 employed traditional methods with a ratio of 1 (Grandi et al., 2019). CC_2 utilized the PRISM method (Frees et al., 2007). CC_3 employed a nonlinear mapping method based on velocity and acceleration (H. Kim et al., 2015). CC_4 relied on implicit interaction techniques involving gaze and hand movements (Yu et al., 2021). *EC* utilized VVIR-OM method. As our primary focus was on improving translation manipulation, traditional methods (Grandi et al., 2019) were used for rotation and scaling in all manipulation techniques.

Hardware and software implementation

We used an HTC Vive system, which consists of a HMD with positional tracking and two wireless handheld controllers. To capture eye-tracking data, we integrated the aSee VR eye tracker onto the HMD. Each eye of the HMD had a screen resolution of 1440×1700 pixels, providing a maximum field of view (FOV) of 110°. The eye tracker had an average sampling rate of 250 Hz and an accuracy of 0.5°, with a maximum FOV of 110° as well. The HMD was connected to a desktop computer equipped with an Intel 7 processor, 16GB RAM, and an NVIDIA 3080Ti graphics card. The software implementation was done using *C#* in Unity3D. The physical space for hosting the virtual reality program was set at $4m \times 4m$, providing sufficient room for participants to perform the manipulation tasks without the need for physical movement.

Hypothese Our method was designed to allow the user to efficiently manipulate a single object to target positions and poses. Thus, we have formulated the following hypotheses:

H1: Compared to CC_{1-4}, using *EC* allows users to manipulate a single object to the target faster, with efficiency improvements in both coarse and fine phases.

H2: Compared to CC_{1-4}, using *EC* allows users to manipulate a single object to the target position more accurately.

H3: Compared to CC_{1-4}, using *EC* allows users to manipulate a single object to the target with less fatigue.

4.1.2. Task and procedure

The task required participants to manipulate objects to the target position as quickly and accurately as possible using translation, rotation, and scaling while maintaining consistent orientation. Our method primarily focuses on enhancing the efficiency of the translation manipulation. However, typical object manipulation tasks in virtual environments encompass three distinct manipulations: translation, rotation, and scaling. Our method integrates translation with rotation and scaling manipulations. Consequently, in our task design, users are required to complete the entire sequence of manipulations rather than solely performing isolated translation tasks.

In terms of procedure, in a single experiment, first the user is assigned the manipulation method. Subsequently, the scene generates a target object and six initial objects: the target object was represented by a red highlight, located directly in front of the participant, and remained unchanged during the experiment; the initial object is a randomly generated white solid entity. the user then begins manipulation, starting with the selection of the initial object (see Figure 5). Then, the user proceeds to translate the object to the vicinity of the target object, followed by a combination of rotation, scaling, and translation manipulations to align the initial object with the target object. Finally, the user releases the initial object to conclude the manipulation, and the system records the data for this manipulation. the user repeats this process for all six initial objects.

Figure 5. The red highlighted rabbit represents the target, and the six white rabbits represent six combinations of factors. In a single task, the user needs to align each of the six white rabbits to the red rabbit.

In terms of experimental design, the study employed a $5 \times 3 \times 2$ design, with three independent variables: manipulation technique (CC_1, CC_2, CC_3, CC_4, and EC), the angular difference ($35^\circ, 55^\circ$, and 85°), and depth difference ($0\,m$, $2\,m$). In each task, six initial objects are generated in the scene, where each initial object represents a combination of angle and depth differences. The sign of the direction vector and depth difference is randomized when generating initial objects. Additionally, a random perturbation of $[-2^\circ, 2^\circ]$ was added to the angular difference, and a random perturbation of $[-0.2m, 0.2m]$ was added to the depth difference. This ensured a more uniform distribution of the tasks in the spatial domain. To minimize the interference of rotation and scaling, the rotation angle difference was fixed at 30°, and the size of the initial object is 1.1 times the size of the target object. The order of manipulation techniques was also randomized. In a single experiment, for each manipulation technique, each task was repeated six times, resulting in a total of 2160 valid data items (= 12 participants × 5 techniques × 3 angular differences × 2 depth differences × 6 repetitions).

4.1.3. Metrics

We evaluate the techniques based on three objective metrics: efficiency, accuracy, and fatigue. Efficiency metrics include (1) Total completion time, measured in seconds, which represents the average time taken by participants to complete the translation, rotation, and scaling for each object; (2) Coarse translation time, measured in seconds, represents the time taken by participants to translate the object to a position near the target location; (3) Fine translation time, measured in seconds, represents the time taken by participants to perform the translation process from the end of the coarse-grained stage until the object is released. Accuracy is measured by the translation error, in millimeters, which represents the distance between the object's coordinates and the target coordinates at the end of the manipulation. Fatigue metrics include (1) Hand travel distance, measured in meters; (2) Hand rotation angles, measured in degrees; These two metrics are recorded throughout the entire manipulation process.

Since all methods use the same rotation and scaling techniques, there are no significant differences in efficiency and accuracy for rotation and scaling. Therefore, we do not report relevant data for rotation and scaling in the single object manipulation task.

4.1.4. Statistical analysis

For each metric, the values of EC were compared to CC_1, CC_2, CC_3, and CC_4. First, the data were evaluated for outliers using box plots, and any outliers were removed. Then, the Shapiro-Wilk test was used to assess the normality of the data distribution. If the data did not satisfy the normal distribution, the Friedman test (Kendall, 1948) was used for analysis. Otherwise, the assumption of sphericity was evaluated using the Mauchly test (Shapiro & Wilk, 1965), and if violated, the Greenhouse-Geisser correction (Greenhouse & Geisser, 1959) was applied. Subsequently, RM-ANOVA (Gelman, 2005) was conducted, followed by posthoc tests with Bonferroni-adjustment to analyze the differences. In addition to the p-values from the statistical tests, we also estimated the size of the effect using Cohen's d (Cohen, 2013). The 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

4.2.1. Task completion time

Table 1 presents the task completion times for the five methods. The third column provides the mean and standard deviation, the fourth column shows the time efficiency improvement from CC to EC, and columns 5–7 present the statistical analysis of the differences between CC and EC, denoted by asterisks to indicate statistical significance.

The task completion time for EC includes the time of coarse translation, fine translation, rotation, scaling, and reset. The average frequency of reset occurrences is 0.17 times. This is attributed to the fact that the target positions are situated close to the center of VVIR, leading to a concentration of most controller positions within the comfort region. As a result, the possibility of the controller reset is minimized.

The EC showed a significantly shorter completion time compared to the other methods ($p < 0.001$), with a huge effect size. CC_1 and CC_2 had longer and similar task completion times, as both used a 1 : 1 mapping for translation. CC_3 had a relatively shorter completion time due to the absence of an upper limit on the mapping ratio. CC_4 required a lot of hand movements at large depth differences. Additionally, gaze jitter during the alignment phase caused object displacement, resulting in a noticeable increase in manipulation time. Therefore, the overall performance of CC_4 is even slightly lower than that of CC_3.

4.2.2. Coarse translation time

Table 2 presents the coarse translation time. The EC method shows a significantly shorter coarse translation time compared to the other methods ($p < 0.001$). CC_3 and CC_4 also showed advantages during the coarse translation phase. In theory, CC_4 should have a short duration for coarse translation. However, the design of the safety region resulted in discontinuous object movement as the object followed the gaze. Often, when the gaze fell on the target position, the object still had a certain distance to go. Therefore, the overall performance of CC_4 was not so well.

Table 1. Task completion time, in seconds.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	25.01 ± 2.92				
CC_1	39.38 ± 1.97	36.5%	<0.001*	5.76	Huge
CC_2	39.21 ± 2.86	36.2%	<0.001*	4.91	Huge
CC_3	32.55 ± 2.64	23.1%	<0.001*	2.71	Huge
CC_4	33.19 ± 2.74	24.6%	<0.001*	2.89	Huge

Note: The (*) indicate that the result is statistically significant at $p < 0.05$.

Table 2. Coarse translation time, in seconds.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	7.93 ± 1.52				
CC_1	17.14 ± 1.71	53.7%	<0.001*	5.69	Huge
CC_2	14.21 ± 1.91	44.2%	<0.001*	3.63	Huge
CC_3	13.80 ± 1.79	42.6%	<0.001*	3.53	Huge
CC_4	13.04 ± 0.94	39.2%	<0.001*	4.04	Huge

Note: The (*) indicate that the result is statistically significant at $p < 0.05$.

Table 3. fine translation time, in seconds.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	11.28 ± 1.11				
CC_1	12.71 ± 1.19	11.5%	0.055	1.27	Large
CC_2	13.41 ± 1.86	15.9%	0.069	1.39	Large
CC_3	11.88 ± 2.87	5.1%	0.64	0.28	Small
CC_4	12.40 ± 2.02	9.1%	0.25	0.69	Middle

Table 4. Translation error, in millimeters.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	9.9 ± 2.6				
CC_1	16.6 ± 2.1	40.1%	<0.001*	2.83	Huge
CC_2	13.4 ± 1.8	25.8%	0.019*	1.56	Very large
CC_3	15.0 ± 2.1	33.9%	0.003*	2.17	Huge
CC_4	16.1 ± 3.8	38.2%	0.006*	1.9	Huge

Note: The (*) indicate that the result is statistically significant at $p < 0.05$.

4.2.3. Fine translation time

Table 3 presents the fine translation time. *EC* did not show a significant difference in fine translation time compared to other methods ($p > 0.05$), but it was the fastest. The effect size was large when comparing *EC* with CC_1 and CC_2, while the difference between *EC* and CC_3, CC_4, was relatively small.

4.2.4. Translation error

Table 4 presents the translation error. The *EC* method exhibited significantly smaller translation errors compared to the other methods ($p < 0.05$). The effect size was huge when comparing *EC* with CC_1, CC_3, and CC_4, which did not specifically optimize for error reduction. The effect size was very large when comparing *EC* with CC_2, which focused on error reduction. CC_2 adjusted the mapping ratio based on the speed of hand movement, where a mapping ratio less than 1 was used for slower hand movements, which contributed to increased accuracy. However, hand movement exhibited instability, such as participant hand tremors or fluctuations in controller positioning, leading to sudden changes in hand speed and causing object jittering.

4.2.5. Hand travel distance

Table 5 presents the total distance of controller movement. *EC* exhibited a significantly shorter total movement distance compared to other methods ($p < 0.05$), with a huge effect size. CC_1 and CC_2 did not optimize for translation, requiring frequent resetting of the controller after each object movement, resulting in a substantial increase in the actual distance traveled by the controller. CC_3 aimed to reduce the controller's total distance by increasing the mapping ratio based on the controller's speed and acceleration. However, multiple controller resets were still required, and the instability of the mapping ratio often led to additional adjustments in object positions. CC_4 relied on gaze-based object manipulation, significantly reducing the physical burden on the hand. However, the hand still had to perform additional movements when there was a large depth difference.

Table 5. Hand travel distance, in meters.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	2.57 ± 0.91				
CC_1	12.67 ± 2.93	79.7%	<0.001*	4.65	Huge
CC_2	10.98 ± 2.41	76.6%	<0.001*	4.63	Huge
CC_3	5.31 ± 1.13	51.7%	<0.001*	2.68	Huge
CC_4	5.99 ± 0.91	57.2%	<0.001*	3.81	Huge

Note: The (*) indicate that the result is statistically significant at $p < 0.05$.

Table 6. Hand rotate angles, in degrees.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	157.73 ± 58.91				
CC_1	954.28 ± 78.19	83.5%	<0.001*	4.62	Huge
CC_2	582.79 ± 55.65	72.9%	<0.001*	7.41	Huge
CC_3	272.76 ± 65.76	42.2%	0.007*	1.84	Very large
CC_4	249.63 ± 40.72	36.8%	0.008*	1.81	Very large

Note: The (*) indicate that the result is statistically significant at $p < 0.05$.

4.2.6. Hand rotation angles

Table 6 provides the total angle of controller rotation. *EC* exhibited a significantly smaller total rotation angle compared to other methods ($p < 0.05$). The effect size of *EC* is Huge compared to both CC_1 and CC_2, and the effect size is Very Large compared to both CC_3 and CC_4. Considering that the object rotation angle in each task was fixed at 30 degrees and that the rotation techniques were the same across the five methods, we believe that the main controller rotation occurred during the intense coarse movement phase. Compared to CC_1, CC_2, CC_3, and CC_4, *EC* required the least controller movement during this phase, resulting in the significantly smallest total rotation angle.

4.3. Discussion

The results in Tables 1–3 support **H1**: The reasons for the efficiency of our method may be as follows: (1) Through the mapping from HHIR to VVIR, users can quickly move the object to a nearby target position during the coarse phase with a single swift continuous handler movement. (2) Based on the analysis of gaze and hand interaction with IHV, VVIR can be updated rapidly and accurately, allowing efficient phase transitions. (3) VVIR updates in real-time based on the user's manipulation, so the mapping changes in line with user expectations, providing a smooth and natural manipulational experience, with additional reset manipulations required only in specific cases. However, while the results in Table 3 indicate that *EC*'s fine translation time is slightly less than CC_{1-4}, the advantage is not significant. This could be attributed to the presence of enhanced fine phases, leading to an additional update of VVIR during the fine phase, as well as a user manipulation. Nevertheless, considering the improvement in manipulation accuracy brought about by the enhanced fine phase, we find this to be acceptable.

The results in Table 4 support **H2**: Our method adjusted the mapping relationship by reducing the VVIR. When the VVIR was stable, the mapping between the controller and the object remained stable as well. Additionally, each stage of the VVIR better reflected the effective range of object movement, facilitating more precise alignment.

The results in Tables 5 and 6 support **H3**: We gauge user fatigue in terms of hand travel distance and hand rotation angles. Compared to CC_{1-4}, *EC* demonstrates significantly smaller hand travel distances and hand rotation angles, indicating that users expend the least effort when using *EC*.

5. User study 2: Multiple objects manipulation

In the second user study, a multiple objects manipulation task was designed to evaluate the objective performance and subjective experience of VVIR-OM.

5.1. Study design

We recruited the same 32 participants from Study 1 to participate in the second study. Each participant was required to use the five manipulation techniques to complete the task of arranging furniture in a living room. The equipment and development platform were the same as those used in user study 1.

Hypothese Our method enables users to efficiently accomplish multiple objects manipulation tasks. Hence, we have formulated the following hypotheses:

H4: Compared to CC_{1-4}, using *EC* allows users to complete the multiple objects manipulation task in less time.

H5: Compared to CC_{1-4}, using *EC* allows users to achieve greater accuracy in the multiple objects manipulation task, with higher translation precision, while rotation and scaling precision are similar.

H6: Compared to CC_{1-4}, using *EC* allows users to complete the multiple object manipulation task with lower fatigue.

H7: The user task load of *EC* is lower than that of CC_{1-4}.

H8: *EC* is easier to use than CC_{1-4}.

5.1.1. Task and procedure

The task required participants to use manipulation techniques to arrange multiple pieces of furniture in the virtual room to their predefined positions (see Figure 6). The target state of the furniture was represented by a red highlight. Unlike close-range manipulation techniques, participants were not allowed to teleport or physically move during the task.

In terms of the procedure, in a single experiment, the user is assigned the manipulation method. Then the user begins furnishing a room. The process of manipulating individual objects is the same as in user study 1. When the last piece of furniture is completed, the system records the data for this experiment.

The study used a 5×6 design, with two independent variables: manipulation techniques (CC_1, CC_2, CC_3, CC_4, and *EC*) and furniture. To cover a wide range of manipulation scenarios while minimizing the task load, a total of 6 furniture items were placed in the room, corresponding to three angular distances (small, medium, large) and two depth differences (small, large) combinations. To make the task more realistic and challenging, six furniture items with distinct shapes and sizes were selected. For each manipulation technique, participants were required to repeat the room layout task 6 times.

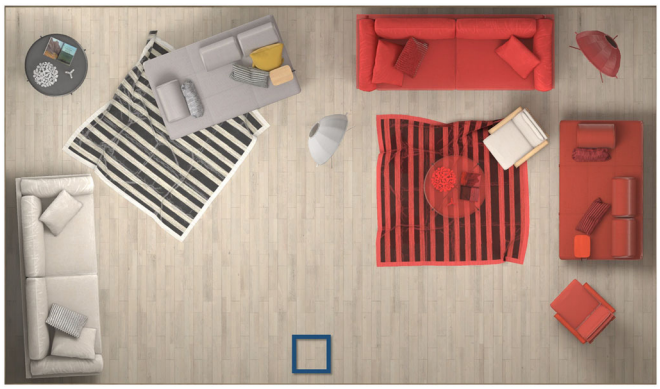

Figure 6. Six pieces of furniture are scattered in the room, and the correct placement of each piece of furniture is indicated by highlighting them in red. The blue rectangle represents the user's position.

5.1.2. Metrics

We evaluate the techniques using three objective metrics: efficiency, accuracy, and fatigue. In the multiple object manipulation task, the efficiency metric is the layout completion time, measured in seconds, from the participant's selection of the first furniture item to the release of the last furniture item, including non-manipulational time. The accuracy metric includes (1) total translation error, measured in millimeters, which is the sum of translation errors for all furniture items; (2) total rotation error, measured in degrees, which is the sum of rotation errors for all furniture items; (3) total scaling error, measured in terms of the target size of the furniture items, which is the sum of scaling errors for all furniture items. The fatigue metric includes (1) hand travel distance, measured in meters; (2) hand rotation angles, measured in degrees.

We also employ three subjective metrics to evaluate participants' VR experience: (1) the standard NASA TLX questionnaire (Hart, 2006; Hart & Staveland, 1988), used to evaluate participants' workload; (2) the usability questionnaire (H. Kim et al., 2015), used to evaluate participants' perception of the technology's usability. The statistical analysis methods used in this study were the same as those in Study 1; (3) the standard SSQ (Kennedy et al., 1993) questionnaire, used to evaluate participants' simulator sickness.

5.2. Results

5.2.1. Layout completion time

Table 7 presents the completion time for the placement task, which includes the durations of coarse translation, fine translation, rotation, scaling, and reset. The average frequency of reset occurrences is 0.6 times for each piece of

Table 7. Layout completion time, in seconds.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	176.55 ± 12.04				
CC_1	280.09 ± 24.78	37.1%	<0.001*	5.31	Huge
CC_2	273.96 ± 17.94	35.6%	<0.001*	6.38	Huge
CC_3	259.01 ± 34.63	31.8%	<0.001*	3.18	Huge
CC_4	229.87 ± 20.87	23.2%	<0.001*	3.13	Huge

Note: The (*) indicate that the result is statistically significant at $p < 0.05$.

Table 8. Total translation error, in millimeters.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	56.46 ± 8.40				
CC_1	108.43 ± 31.30	47.8%	0.006*	2.26	Huge
CC_2	96.76 ± 25.46	41.5%	0.009*	2.12	Huge
CC_3	113.51 ± 21.95	50.2%	<0.001*	3.43	Huge
CC_4	119.30 ± 30.77	52.6%	0.002*	2.78	Huge

Note: The (*) indicate that the result is statistically significant at $p < 0.05$.

Table 9. Total rotation error, in degrees.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	5.26 ± 3.31				
CC_1	6.19 ± 2.91	15.0%	0.63	0.30	Small
CC_2	5.98 ± 3.74	12.1%	0.74	0.2	Very small
CC_3	6.15 ± 2.84	14.5%	0.64	0.29	Small
CC_4	4.94 ± 2.67	−6.6%	0.86	0.11	Very small

furniture. This is due to the common placement of furniture near walls positioned at the boundary of VVIR. As the coarse phase of manipulation concludes, there is a greater probability that the controller may reside at the boundary of HIHR, subsequently triggering controller resets. *EC* exhibited a significantly shorter total time compared to other methods ($p < 0.05$), with a huge effect size when compared to other methods. CC_1 and CC_2, which lacked improvements in translation, had the longest completion times. CC_4, based on gaze and hand interaction, performed the best among the comparison methods but still had a notable difference from *EC*. This suggests that *EC* significantly improved manipulation efficiency in the context of semantically meaningful placement tasks.

5.2.2. Total translation error

Table 8 presents the total translation error for the placement task. *EC* exhibited a significantly smaller total translation error compared to other methods ($p < 0.05$), with a huge effect size when compared to other methods. There was no significant difference in error between CC_1, CC_2, and CC_4. CC_2, which had technical improvements for fine manipulation, performed the best among the four comparison methods but still had a noticeable difference compared to *EC*.

5.2.3. Total rotation error

Table 9 presents the total rotation error. The rotation error of method *EC* is not significantly different from other methods, and the effect size is below the Small. This indicates that all methods perform similarly in the rotation phase of the task.

Table 10. Total scaling error.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	0.089 ± 0.041				
CC_1	0.101 ± 0.043	12.0%	0.65	0.29	Small
CC_2	0.099 ± 0.028	9.6%	0.66	0.27	Small
CC_3	0.075 ± 0.030	−19.4%	0.41	0.51	Medium
CC_4	0.137 ± 0.084	34.6%	0.31	0.72	Medium

Table 11. Hand travel distance, in meters.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	15.41 ± 2.64				
CC_1	102.42 ± 7.89	85.0%	<0.001*	14.79	Huge
CC_2	99.58 ± 14.38	84.5%	<0.001*	8.14	Huge
CC_3	24.51 ± 4.46	37.1%	0.003*	2.48	Huge
CC_4	40.15 ± 10.71	61.6%	<0.001*	3.17	Huge

Note: The (*) indicate that the result is statistically significant at $p < 0.05$.

Table 12. Hand rotate angle, in degrees.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	1816.19 ± 228.25				
CC_1	3332.57 ± 1168.76	45.5%	0.023*	1.80	Very large
CC_2	3219.63 ± 1010.31	43.6%	0.016*	1.92	Very large
CC_3	2640.95 ± 963.75	31.2%	0.111	1.18	Very large
CC_4	3479.19 ± 750.32	47.8%	0.001*	3.01	Huge

Note: The (*) indicate that the result is statistically significant at $p < 0.05$.

5.2.4. Total scaling error

Table 10 presents the total scaling error. The scaling error of *EC* is not significantly different from other methods, and the effect size is below the median. This indicates that all methods perform similarly in the scaling phase.

5.2.5. Hand travel distance

Table 11 presents the total hand movement distance for the placement task. *EC* demonstrated significantly shorter total movement distance compared to other methods ($p < 0.05$), with a huge effect size in all comparisons. Among the four comparison methods, CC_3 performed the best in terms of total distance. We attribute CC_3's performance to the large mapping ratio used throughout the task. However, the instability of the mapping ratio made it challenging to further reduce the total distance. CC_4 exhibited a slightly higher total distance compared to CC_3, but it was still significantly shorter than CC_1 and CC_2. This suggests that gaze-based manipulation can effectively reduce hand movement distance, but its performance is not optimal in complex environments with varying depths.

5.2.6. Hand rotation angles

Table 12 presents the total hand rotation angle for the placement task. *EC* demonstrated a significantly shorter total rotation angle compared to CC_1, CC_2, and CC_4 ($p < 0.05$). There was no significant difference in total rotation angle between *EC* and CC_3 ($p = 0.11$), but *EC* still had the shortest rotation angle among the five methods. CC_3 performed well due to overall less hand movement required. CC_4 exhibited the poorest performance among all methods. We believe this is mainly due to the degradation of manipulation when dealing with large depth differences, as well as the increased

Table 13. NASA Task load index data.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	18.67 ± 4.39				
CC_1	29.44 ± 3.27	36.6%	<0.001*	2.78	Huge
CC_2	25.67 ± 4.76	27.3%	0.007*	1.53	Very large
CC_3	31.78 ± 6.64	41.3%	<0.001*	2.33	Huge
CC_4	34.22 ± 7.02	45.5%	<0.001*	2.66	Huge

Note: The (*) indicate that the result is statistically significant at $p < 0.05$.

Table 14. Usability questionnaire data.

Condition	Avg ± std. dev.	$(CC_i\text{-}EC)/CC_i$	p	Cohen's d	Effect size
EC	44.67 ± 5.79				
CC_1	33.89 ± 5.68	31.8%	0.002*	1.88	Very large
CC_2	34.22 ± 10.23	30.5%	0.023*	1.26	Very large
CC_3	32.44 ± 8.06	37.7%	0.003*	1.74	Very large
CC_4	36.32 ± 8.43	22.9%	0.034*	1.15	Very large

Note: The (*) indicate that the result is statistically significant at $p < 0.05$.

burden on hand manipulation caused by frequent eye movement during the manipulation.

We conducted a user workload assessment using the NASA TLX questionnaire, and the results are presented in Table 13. Overall, in tasks involving semantic complexity, the user workload of *EC* was significantly lower compared to CC_1, CC_2, CC_3, and CC_4. Among the comparative methods, CC_2 had the lowest user workload. The user workload during indoor arrangement tasks was more significantly influenced by fine manipulations.

The results of the technology usability assessment are shown in Table 14. *EC* demonstrated significantly higher usability compared to CC_1, CC_2, CC_3, and CC_4, with a Very Large effect size for all comparisons. CC_1 and CC_4 also received favorable ratings in terms of usability. However, the instability of the mapping ratio resulted in the poorest usability evaluation for CC_3.

Participants completed the SSQ before and after the task. We analyzed the SSQ data using the Total Severity score (TS) (Kennedy et al., 1993). The scores before and after the tasks of EC are 3.71 ± 2.37 and 4.43 ± 1.05, which are slightly smaller than those of other methods.

5.3. Discussion

The results in Table 7 support **H4**: The layout completion time in *EC* is significantly less than CC_{1-4}. This is because (1) The cross-viewport manipulation algorithm allows users to quickly manipulate objects outside their field of view by moving their head, bringing the objects and target positions into the same viewport, thereby simplifying the manipulation task rapidly. (2) The dynamically updated VVIR facilitates both rapid coarse-grained movements and stable fine-grained adjustments, reducing errors in user manipulation.

The results in Tables 8–10 support **H5**: The total translation error in *EC* is significantly smaller than in CC_{1-4}. This is attributed to the enhanced fine phase, allowing users to make finer position adjustments. In contrast, there is no significant difference in rotation and scaling compared to CC_{1-4}, as this aligns with our expectations, since we did not make improvements specifically for rotation and scaling precision.

The results in Tables 11 and 12 support **H6**: The hand travel distance and hand rotate angle in *EC* are both significantly smaller than in CC_{1-4}. This is because users can complete the long distance coarse phase manipulation with little hand movement. Additionally, in the fine phase, a stable spatial mapping reduces manipulation errors, which means users expend less effort.

The results in Table 13 support **H7**: Compared to CC_{1-4}, our method significantly reduces the task load. This is because in our method, users experience shorter hand travel distances and smaller rotation angles, thus reducing the physical load. The mapping is adaptively adjusted, relieving users from the need to actively monitor it, thereby reducing the mental load.

The results in Table 14 support **H8**: In our method, users do not need to pay attention to changes in mapping relationships or multiple modal inputs. They simply need to naturally look at the target area and move the hand while manipulating. This makes it user-friendly. Furthermore, in our method, mapping relationships remain stable, making users feel that their manipulations are more efficient and accurate, ultimately enhancing the user experience.

The effectiveness of VVIR-OM indicates that keeping hand movements within the comfort region significantly reduces manipulation fatigue. Therefore, the HIHR can serve as a reference for other manipulation methods to improve user load and cybersickness. Additionally, our method focuses on the dynamically changing user's region of interest in manipulation, represented through adaptively updated VVIR. Hence, the introduced VVIR and IHV in our method can be used as inspiration for other manipulation methods to improve efficiency and accuracy. Lastly, our method introduces various modalities to assist users in manipulation without increasing the learning and manipulation burden, which can also be borrowed by other manipulation methods to further enhance efficiency and improve the user experience.

6. Conclusions, limitations and future work

We have proposed a new object manipulation method based on the variable virtual interaction region to improve manipulation efficiency and accuracy and reduce user fatigue. We have introduced two concepts: the hand interaction hemisphere region and the variable virtual interaction region. We also have proposed an interaction heat volume based updating method to automatically update the variable virtual interaction region in the manipulation process. We used two user studies to evaluate the performance of VVIR-OM with two tasks, single object manipulation and multiple object manipulation. The results indicated that VVIR-OM achieved significant improvements in efficiency, accuracy, and fatigue compared to the state-of-the-art methods. Additionally, VVIR-OM demonstrated significant advantages over other methods in terms of task load and usability without exacerbating cybersickness.

VVIR-OM has some limitations. Similar to most existing manipulation methods, VVIR-OM only considers manipulating virtual objects when the user keeps static in VR environment. If the user walks while performing manipulation, HIHR may shake with the user's motion, resulting in an unstable mapping between the position of the hand and the position in the virtual space. Our VVIR update algorithm focuses on the process of shrinking the virtual interaction region, and this is because for a manipulation process, there is a higher probability of moving from the coarse phase to the fine phase and then to the enhanced fine phase. If the object is still far from the target location when entering the fine phase, the smaller virtual interaction region may not cover the target location, which then requires the user to use manual interaction by pressing a button and reverting to the coarser phase, but this manual method is simple and fast.

One future work is to adapt VVIR-OM to scenarios where users need to walk and interact simultaneously. This can be achieved by designing algorithms to improve the stability of mapping. Another is to investigate a stable and fully automatic method for adaptive updating of VVIR, including VVIR getting smaller and regression getting larger, eliminating the need for users to adjust the virtual interaction region manually. We will also explore rotation and scaling methods based on interaction region mapping and interaction heat to form an integrated system of manipulation methods.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

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

Abtahi, P., Hough, S. Q., Landay, J. A., & Follmer, S. (2022). *Beyond being real: A sensorimotor control perspective on interactions in virtual reality* [Paper presentation]. Proceedings of the 2022 Chi Conference on Human Factors in Computing Systems (pp. 1–17). https://doi.org/10.1145/3491102.3517706

Aguerreche, L., Duval, T., & Lécuyer, A. (2009). Short paper: 3-Hand manipulation of virtual objects. In *Joint Virtual Reality Conference of EGVE - ICAT – Euro VR 2009* (pp. 153–156). https://doi.org/10.2312/EGVE/JVRC09/153-156

Bachynskyi, M., Palmas, G., Oulasvirta, A., & Weinkauf, T. (2015). Informing the design of novel input methods with muscle coactivation clustering. *ACM Transactions on Computer-Human Interaction*, *21*(6), 1–25. https://doi.org/10.1145/2687921

Bowman, D. A., & Hodges, L. F. (1997). *An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments* [Paper presentation]. Proceedings of the 1997 Symposium on Interactive 3D Graphics (pp. 35–ff). https://doi.org/10.1145/253284.253301

Cheung, Y.-W., & Lai, K. S. (1995). Lag order and critical values of the augmented dickey–fuller test. *Journal of Business & Economic Statistics*, *13*(3), 277–280. https://doi.org/10.2307/1392187

Cohen, J. (2013). *Statistical power analysis for the behavioral sciences*. Academic press.

Feuchtner, T., Müller, J. (2018). *Ownershift: Facilitating overhead interaction in virtual reality with an ownership-preserving hand space shift* [Paper presentation]. Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (pp. 31–43).

Frees, S., & Kessler, G. D. (2005). *Precise and rapid interaction through scaled manipulation in immersive virtual environments* [Paper presentation]. IEEE Proceedings. VR 2005 (pp. 99–106).

Frees, S., Kessler, G. D., & Kay, E. (2007). Prism interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction*, *14*(1), 2–es. https://doi.org/10.1145/1229855.1229857

Gelman, A. (2005). Analysis of variance. *Quality Control Applied Statistics*, *20*(1), 295–300. https://doi.org/10.1214/009053604000001048

Gloumeau, P. C., Stuerzlinger, W., & Han, J. (2020). Pinnpivot: Object manipulation using pins in immersive virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, *27*(4), 2488–2494. https://doi.org/10.1109/TVCG.2020.2987834

Grandi, J. G., Debarba, H. G., & Maciel, A. (2019). *Characterizing asymmetric collaborative interactions in virtual and augmented realities* [Paper presentation]. 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR) (pp. 127–135). https://doi.org/10.1109/VR.2019.8798080

Greenhouse, S. W., & Geisser, S. (1959). On methods in the analysis of profile data. *Psychometrika*, *24*(2), 95–112. https://doi.org/10.1007/BF02289823

Hart, S. G. (2006). Nasa-task load index (NASA-TLX); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *50*(9), 904–908. https://doi.org/10.1177/154193120605000909

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (task load index): Results of empirical and theoretical research. In *Advances in psychology* (Vol. 52, pp. 139–183). Elsevier.

Hincapié-Ramos, J. D., Guo, X., Moghadasian, P., Irani, P. (2014). *Consumed endurance: A metric to quantify arm fatigue of mid-air interactions* [Paper presentation]. Proceedings of the Sigchi Conference on Human Factors in Computing Systems (pp. 1063–1072).

Kendall, M. G. (1948). *Rank correlation methods*. Griffin.

Kennedy, R. S., Lane, N. E., Berbaum, K. S., & Lilienthal, M. G. (1993). Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology*, *3*(3), 203–220. https://doi.org/10.1207/s15327108ijap0303_3

Kim, H., Lee, G., & Billinghurst, M. (2015). *A non-linear mapping technique for bare-hand interaction in large virtual environments* [Paper presentation]. Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction (pp. 53–61). https://doi.org/10.1145/2838739.2838774

Kim, M. G., Ryu, J., Son, J., & Han, J. (2022). Virtual object sizes for efficient and convenient mid-air manipulation. *The Visual Computer*, *38*(9–10), 3463–3474. https://doi.org/10.1007/s00371-022-02555-6

Lim, C., Kim, J., & Kim, M. J. (2022). *Thumble: One-handed 3D object manipulation using a thimble-shaped wearable device in virtual reality* [Paper presentation]. Adjunct Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (pp. 1–3). https://doi.org/10.1145/3526114.3558703

Lisle, L., Lu, F., Davari, S., Tahmid, I. A., Giovannelli, A., Llo, C., Pavanatto, L., Zhang, L., Schlueter, & Bowman, D. A. (2022). *Clean the ocean: An immersive VR experience proposing new modifications to Go-Go and WiM techniques* [Paper presentation]. 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW) (pp. 920–921). https://doi.org/10.1109/VRW55335.2022.00311

Liu, C., Plopski, A., & Orlosky, J. (2020). Orthogaze: Gaze-based three-dimensional object manipulation using orthogonal planes. *Computers & Graphics*, *89*, 1–10. https://doi.org/10.1016/j.cag.2020.04.005

Liu, X., Wang, L., & Luan, S. (2023). Manipulation guidance field for collaborative object manipulation in VR. *International Journal of*

Human–Computer Interaction. Advance online publication. https://doi.org/10.1080/10447318.2023.2250941

McAtamney, L., & Corlett, E. N. (1993). Rula: A survey method for the investigation of work-related upper limb disorders. *Applied Ergonomics*, *24*(2), 91–99. https://doi.org/10.1016/0003-6870(93)90080-s

Mendes, D., Caputo, F. M., Giachetti, A., Ferreira, A., & Jorge, J. (2019). A survey on 3D virtual object manipulation: From the desktop to immersive virtual environments. *Computer Graphics Forum*, *38*(1), 21–45. https://doi.org/10.1111/cgf.13390

Montano Murillo, R. A., Subramanian, S., Martinez Plasencia, D. (2017). *Erg-o: Ergonomic optimization of immersive virtual environments* [Paper presentation]. Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (pp. 759–771).

Monteiro, P., Goncalves, G., Coelho, H., Melo, M., & Bessa, M. (2021). Hands-free interaction in immersive virtual reality: A systematic review. *IEEE Transactions on Visualization and Computer Graphics*, *27*(5), 2702–2713. https://doi.org/10.1109/TVCG.2021.3067687

Nguyen, T. T. H., & Duval, T. (2013). *Poster: 3-point++: A new technique for 3d manipulation of virtual objects* [Paper presentation]. 2013 IEEE Symposium on 3d User Interfaces (3DUI) (pp. 165–166). https://doi.org/10.1109/3DUI.2013.6550230

Pfeuffer, K., Mayer, B., Mardanbegi, D., Gellersen, H. (2017). *Gaze + pinch interaction in virtual reality* [Paper presentation]. Proceedings of the 5th Symposium on Spatial User Interaction (pp. 99–108).

Pierce, J. S., Stearns, B. C., Pausch, R. (1999). *Voodoo dolls: Seamless interaction at multiple scales in virtual environments* [Paper presentation]. Proceedings of the 1999 Symposium on Interactive 3D Graphics (pp. 141–145).

Pohl, H., Lilija, K., McIntosh, J., & Hornbæk, K. (2021). *Poros: Configurable proxies for distant interactions in VR* [Paper presentation]. Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (pp. 1–12). https://doi.org/10.1145/3411764.3445685

Poupyrev, I., Billinghurst, M., Weghorst, S., & Ichikawa, T. (1996). *The go-go interaction technique: Non-linear mapping for direct manipulation in VR* [Paper presentation]. Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology (pp. 79–80). https://doi.org/10.1145/237091.237102

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, *52*(3–4), 591–611. https://doi.org/10.1093/biomet/52.3-4.591

Slambekova, D., Bailey, R., & Geigel, J. (2012). *Gaze and gesture based object manipulation in virtual worlds* [Paper presentation]. Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology (pp. 203–204). https://doi.org/10.1145/2407336.2407380

Turner, J., Alexander, J., Bulling, A., Gellersen, H. (2015). *Gaze + rst: Integrating gaze and multitouch for remote rotate-scale-translate tasks* [Paper presentation]. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (pp. 4179–4188).

Yu, D., Lu, X., Shi, R., Liang, H.-N., Dingler, T., Velloso, E., & Goncalves, J. (2021). *Gaze-supported 3d object manipulation in virtual reality* [Paper presentation]. Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (pp. 1–13). https://doi.org/10.1145/3411764.3445343

About the authors

Qinwen Zheng is a master student in the School of Computer Science and Engineering of Beihang University, China. His current research focuses on virtual reality, augmented reality, and HCI.

Lili Wang received her PhD degree from 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, real-time rendering and HCI.

Wei Ke received his PhD degree from Beihang University, Beijing, China. He is currently a professor at the Faculty of Applied Sciences, Macao Polytechnic University. His research interests include programming languages, image processing, computer vision, and tool support for component-based engineering and systems.

Sio Kei Im received his PhD degree from Queen Mary University of London, United Kingdom. He is a professor at the Faculty of Applied Sciences, Macau Polytechnic University. His research interests include video coding, image processing, machine learning for NLP and multimedia.