HPIPainting: A Hand-Pen Interaction for VR Painting

ANG CAI, School of Computer and Communication Engineering, USTB, China
CHAO YAO*, School of Computer and Communication Engineering, USTB, China
HONGJUN LIU, School of Intelligence Science and Technology, USTB, China
CHANGSHENG LI, School of Intelligence Science and Technology, USTB, China
YONGYUE ZHANG, School of Intelligence Science and Technology, USTB, China
YU GUO, School of Intelligence Science and Technology, USTB, China
XIAOKUN WANG, School of Intelligence Science and Technology, USTB, China
XIAOJUAN BAN*, School of Intelligence Science and Technology, USTB, China

Virtual Reality (VR) painting applications allow users to create visual imagery in 3D space. However, existing bare-hand VR painting and sketching systems frequently rely on generic hand gestures, which could lead to significant user misunderstandings. In this paper, we propose HPIPainting, a bare-hand VR painting system that implements a virtual pen interaction mechanism based on the Hand-Pen Interaction(HPI) paradigm. This paradigm leverages microgesture recognition to integrate natural pen-grasping gestures into a precise interaction model, enabling users to sketch in 3D space naturally and immersively. Specifically, we explored the design space for VR painting gestures and filtered out 9 microgestures through subjective evaluations to control various painting functions, such as start painting, edit, brush adjustment, geometric creation, and grid operations. Studies demonstrate that HPIPainting improves the immersion, usability, and expressive freedom of bare-hand VR Painting, it achieves controller-level drawing accuracy with a mean error of 1.21 mm, fast and reliable mode-switching within 222 ms, and significantly higher ratings for ease of use, hand fatigue, and naturalness compared to pinch- and controller-based input.

CCS Concepts: • **Human-centered computing** → **Virtual reality**.

Additional Key Words and Phrases: Bare-hand, Hand-Pen Interaction Paradigms, Microgesture, Virtual Pen

ACM Reference Format:
Ang Cai, Chao Yao, Hongjun Liu, Changsheng Li, Yongyue Zhang, Yu Guo, Xiaokun Wang, and Xiaojuan Ban. 2025. HPIPainting: A Hand-Pen Interaction for VR Painting. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 3, Article 71 (September 2025), 26 pages. https://doi.org/10.1145/3749538

*Corresponding authors.

Authors' Contact Information: Ang Cai, School of Computer and Communication Engineering, USTB, Beijing, China, M202320931@xs.ustb.edu.cn; Chao Yao, School of Computer and Communication Engineering, USTB, Beijing, China, yaochao@ustb.edu.cn; Hongjun Liu, School of Intelligence Science and Technology, USTB, Beijing, China, D202210386@xs.ustb.edu.cn; Changsheng Li, School of Intelligence Science and Technology, USTB, Beijing, China, M202320825@xs.ustb.edu.cn; Yongyue Zhang, School of Intelligence Science and Technology, USTB, Beijing, China, M202421746@xs.ustb.edu.cn; Yu Guo, School of Intelligence Science and Technology, USTB, Beijing, China, guoyu@ustb.edu.cn; Xiaokun Wang, School of Intelligence Science and Technology, USTB, Beijing, China, wangxiaokun@ustb.edu.cn; Xiaojuan Ban, School of Intelligence Science and Technology, USTB, Beijing, China, banxj@ustb.edu.cn.

ACM 2474-9567/2025/9-ART71
https://doi.org/10.1145/3749538

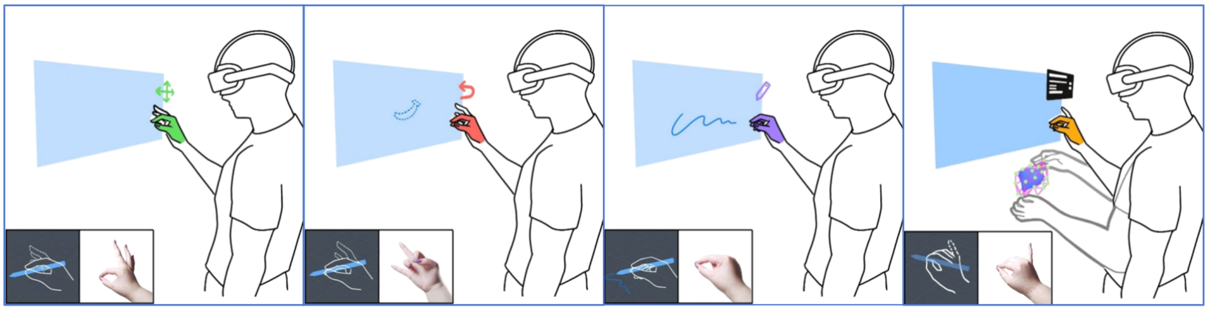


Fig. 1. HPIPainting's design concept: By recognizing microgestures in users' pen-holding postures in real-time to change virtual pen states, the system maps common drawing operations (adjusting brush size/type, undo/redo, start drawing, and call out menu) in 3D space to corresponding virtual pen states. Using the virtual pen as a painting agent, users can create naturally and intuitively in both 2D canvas and 3D space.

1 INTRODUCTION

The rapid development of Virtual Reality (VR) and Augmented Reality (AR) technology has opened up new possibilities for creative expression in various artistic fields such as painting, sketching, and 3D modeling. The immersive environments and natural gesture interactions provided by VR technology have changed the way artists and designers create and share their ideas. Among these creative activities, bare-hand sketching plays a crucial role in the early stages of design, enabling the rapid exploration and communication of concepts [44, 53].

Existing bare-hand VR sketching and painting systems primarily employ hardware-based methods to enable creative interactions, such as handheld controllers with 6 DoF [38], stylus-like tools [24], or VR tablets [1]. These controller-based interactions often necessitate that users monitor the on-screen cursor movement while also assessing the motion of the device beyond their direct line of sight. This disconnection between visual input and physical action demands a heightened level of attention for users to coordinate the two, thereby increasing their cognitive load. These controller-based approaches may not provide the most natural and immersive experience for bare-hand sketching and painting in VR.

Recent research has explored the potential of bare-hand interaction as an alternative to controller-based methods for VR painting and sketching [5, 10, 51]. These approaches typically rely on using fingertips or the entire finger as a brush, recognizing actions such as fingertip pinching or index finger bending to draw lines in 3D space [15, 21, 37]. However, the gestures used in these systems are generally generic bare-hand input gestures rather than being specifically designed for painting tasks. As a result, they tend to be ambiguous and cognitively demanding when applied to creative drawing scenarios. This mismatch increases the risk of gesture misrecognition and hinders the user's ability to perform fine-grained, multi-step operations required in professional VR painting. Prolonged use of such general-purpose gestures can lead to increased cognitive load, thereby reducing usability and making it difficult to accurately interpret user intentions or support complex interaction workflows [2]. To achieve bare-hand interaction in professional VR painting applications where users need to use multiple tools and control methods, developing an interaction paradigm that aligns with user cognition and accurately controls complex painting processes is crucial. Therefore, we propose a bare-hand 3D painting method based on natural painting gestures.

This paper presents HPIPainting, a novel bare-hand painting system in virtual reality that enables 3D drawing through the use of a virtual pen. Inspired by real-world 3D printing pens, HPIPainting introduces a new interaction paradigm called Hand Pen Interaction (HPI), which leverages microgesture recognition to integrate natural pen-grasping gestures with a virtual pen as the primary interaction agent. We explored the design space of VR painting gestures and proposed 21 low-amplitude gestures and evaluated them using 3 subjective criteria such as cognitive load, fatigue, and social adaptability, as well as 2 objective criteria including recognition accuracy and time demand. Based on this evaluation, 9 gestures were selected for their overall performance. These gestures are designed to maintain consistency with the pen-grasping posture, resulting in minimal motion variance and clear functional mapping.

As shown in Fig. 1, in HPIPainting, the selected microgestures are mapped to commonly used painting functions in VR, such as changing brush type, adjusting brush thickness, undo and redo, and call out menu. When the user picks up the pen, the available operations are centred around the virtual pen; when the user puts down the pen, the available operations are centred around the sketches. This design reduces the user learning curve by aligning with familiar painting habits.

To further enhance usability, HPIPainting introduces a smart guidance interface that displays real-time interaction status at the virtual pen tip. When users grasp the pen, the system enters drawing mode, supporting the creation of arbitrary lines and geometric shapes. When the pen is released, the system switches to editing mode, enabling operations such as repositioning, resizing, and deformation of objects. This context-aware design avoids gesture ambiguity. For instance, the same 'Prolong Middle Finger(PLM)' gesture is used to change brush type in drawing mode and to duplicate an object in editing mode. Additionally, a smart guidance interface displays gesture-function mappings on the pen and provides real-time visual fitting of shapes such as circles and straight lines, allowing users to confirm or reject the system's suggestions.

To evaluate the effectiveness and user experience of HPIPainting, we conducted a series of user studies. We compared the HPI paradigm with 2 painting methods, pinch and controller, and compared 6 mode-switching techniques. Experimental results, measured by 4 objective metrics [2, 46, 50] (tracing error, tracing completion time, mode-switching time, and mode-switching error rate) and subjective evaluation dimensions. Results showed that the HPI paradigm achieves comparable performance to the controller in terms of drawing accuracy and mode-switching time, and reduced mental and physical demands. It also received significantly higher ratings in terms of usability, hand fatigue reduction, and VR naturalness. We also explored the potential of HPIPainting system for VR 3D sculpture applications. Subjective feedback from participants indicated that they preferred the gestural interaction of HPIPainting because it enhanced the realism of the bare-hand VR painting experience as if they were painting in the real world.

The main contributions of this work include:

- We have introduced the Hand Pen HPI paradigm for bare-hand painting, which allows users to draw in 3D space using pen-holding gestures that are more consistent with real-world cognition.
- We explored the microgesture design space of HPIPainting and proposed a set of actions and corresponding functional mappings of the VR painting system through user-centred design.
- We have carried out a series of evaluation experiments that show that HPIPainting is a more expressive system for painting with bare hands and has shown potential in VR 3D sculpture.

2 RELATED WORKS

HPIPainting enables users to sketch with bare hands in VR by integrating sketching functions into hand-pen interactions. We draw inspiration from previous work exploring sketching interfaces in VR, creative hand interactions, and physics and hardware-based drawing methods in VR/AR.

2.1 Bare-hand and Hardware-based Interaction in VR

With virtual reality development, humans can manipulate objects and communicate through gestures because hands are highly manipulable and expressive. Freehand interaction has become important research, promoting 3D creation tools that support freehand interaction as input. Two main categories of freehand interaction techniques in VR are bare-hand interaction and hardware-based interaction, each with advantages and limitations. Bare-hand interaction in VR relies on recognition and tracking of users' hand gestures and movements, allowing natural and intuitive interaction with virtual environments without additional input devices [49]. This approach replicates how people interact with the real world, making it easier for users to adapt to VR experiences. Bare-hand interaction can be achieved through vision sensors [54], computer vision algorithms [7], and machine learning techniques [11]. These methods enable systems to interpret users' hand gestures and translate them into corresponding virtual actions. Main advantages include enhanced immersion and presence [31], and higher flexibility and expressiveness compared to hardware-based methods [52].

However, gesture recognition accuracy and reliability can be affected by occlusion, lighting conditions, and individual differences in hand shapes and movements [3]. Additionally, lack of haptic feedback makes it difficult for users to perceive boundaries and properties of virtual objects [6]. Hardware-based interaction relies on physical input devices, such as hand-held controllers, joysticks, or specialized VR accessories like gloves [33] or styluses [9]. These devices provide tangible interfaces to interact with virtual environments, often offering haptic feedback and precise control [35]. For example, controllers like those in HTC Vive or Oculus Rift allow users to manipulate virtual objects, navigate virtual space, and perform actions using buttons, triggers, and touchpads [9, 32]. Main advantages include increased precision and control [40], reduced physical strain and fatigue [28], and potential for widespread adoption. However, additional physical devices can hinder immersion and presence [43], users may experience fatigue during prolonged use [14], and cost and availability of specialized accessories can be barriers [22]. Another limitation is reduced flexibility and naturalness compared to bare-hand interaction [4].

In VR sketching and painting, both bare-hand systems like Air Painting [47] and SwingStroke [42], and hardware-based systems like TiltBrush [13] and Quill [12], have been explored to provide natural and expressive painting experiences or precise control and haptic feedback, respectively. With hand tracking technology advancement, major VR headsets now offer seamless bare-hand interaction. Microsoft Hololens 2 [18], HTC Vive [17], Apple Vision Pro [16], Pico 4 pro[20], and Meta Quest series [19] can track fingers, palms, and wrist joints in real time. This enables users to utilize complex hand dexterity and expressiveness in various activities. In future VR/AR applications, bare-hand interaction modes are expected to become more common, with social interaction as the main theme [26].

2.2 Sketching and Painting Systems in VR

Virtual Reality has provided artists and designers with novel creative environments. One common approach uses handheld controllers as input devices. Commercial software like Tilt Brush [13] and Quill [12] allow users to draw lines and shapes in 3D space using controllers, providing rich brushes and color options. Researchers have developed prototype systems based on handheld devices, such as Cave Painting [25] and Drawing on Air [39], exploring more flexible and expressive painting techniques.

However, controller-based systems have limitations in naturalness and immersiveness, requiring users to adapt to controller operation and making it difficult to achieve fine control similar to real-world painting. Another approach uses specially designed painting tools, such as styluses [38] and tablets [39], to simulate traditional painting experiences. These tools are often used with virtual canvases, allowing users to draw in 2D on canvas and project creations into 3D space. While this method provides more precise and controllable painting experiences, it lacks flexibility and realism in 3D space. Moreover, specialized devices are often costly and not easily accessible to general users. Some research has explored novel interaction techniques, such as gesture-based painting [34] and gaze-based painting [36], aiming to provide more natural and seamless creative experiences. However, these techniques face challenges in robustness, accuracy, and user adaptability.

Dudley et al. [10] explored using index finger bending gestures to control bare-hand 3D drawing in AR environments, but focused primarily on improving line painting and did not address full sketching workflow challenges. Surale et al. [46] compared performance of 11 bare-hand mid-air dominant and non-dominant hand gestures as mode-switching inputs and experimented with 3D line drawing tasks. The recent application AirDraw [27] on Apple Vision Pro provides 3D drawing functionality based on pinch gestures in AR environments, allowing users to draw and sculpt. However, it focuses on casual creativity rather than professional sketching workflows, limiting application in precise technical drawing tasks. Similarly, Hu et al. [15] proposed finger-pinching gestures as input to draw lines controlling terrain changes. Although bare-hand sketching systems proposed by A. W. Ismail et al. [21] and HandPainter system by Jiang et al. [23] can implement freehand sketching in VR, they still require users to wear additional hardware and rely on complex hand gestures. Existing research has mainly focused on bare-hand sketching based on general gestures such as pinching or VR drawing using hardware input, neglecting users' need for natural and precise bare-hand drawing.

3 STUDY 1: HPI MICROGESTURES DESIGN AND RECOGNITION

In order to make grasping more natural, we preset a fixed grasping position on the virtual pen, requiring index finger and thumb pinching as the essential condition, with other fingers' contact serving as input switches. When users approach the virtual pen and perform the pinching motion, it automatically adheres to their hand.

We systematically explored the microgesture design space for virtual pen grasping, developing basic pen-grasping gestures and expanding them into a comprehensive gesture set. Through evaluation of subjective preference, accuracy, and temporal requirements, we refined the gesture space from 21 candidates to 9 easily executable and well-received gestures. The brainstorming sessions of the participants determined the microgesture mappings and established the HPI paradigm.

3.1 Gesture Design

We proposed four basic microgestures based on pinching of the index and thumb: the thumb, index and middle fingers simultaneously reach a preset gesture finger position (M), the thumb, index, and ring fingers simultaneously reach a preset gesture finger position (R), the thumb, index, and pinky fingers simultaneously reach a preset gesture finger position (P), and all the fingers simultaneously reach a preset position (A). Based on this, as shown in Fig. 2, we expanded the basic gestures according to the three variations:

- **Combination:** we combined the microgestures of the individual fingers except the index finger and the thumb two by two to form the first set of variations (M&R, M&P, R&P)
- **Repetition:** we performed double repetitions of the above microgestures in quick succession to form a new set of variations (DM, DR, DP, DA, and DM&R, DM&P, DR&P), with triple repetitions or more considered too complex or inappropriate.
- **Prolonged maintenance:** this involves holding the basic microgestures and the first set of variations for more than one second to form a new set of variations (PLM, PLR, PLP, PLA and PLM&R, PLM&P, PLR&P).

3.2 Gesture Recognition

3.2.1 VR Interaction Headset. In our experiments, we use the Meta Quest 2 as the interaction headset, which is the same device will be used in the subsequent experiments. The headset supports hand tracking, and the hand tracking interface provides a real-time reconstruction of a virtual 3D hand that we will use to interact with objects.

3.2.2 Microgestures Recognition. For virtual pen grasping, we adopt a method of presetting the grasping gesture position and require the index finger and thumb finger to be pinched together as a necessary condition for picking up the virtual pen.

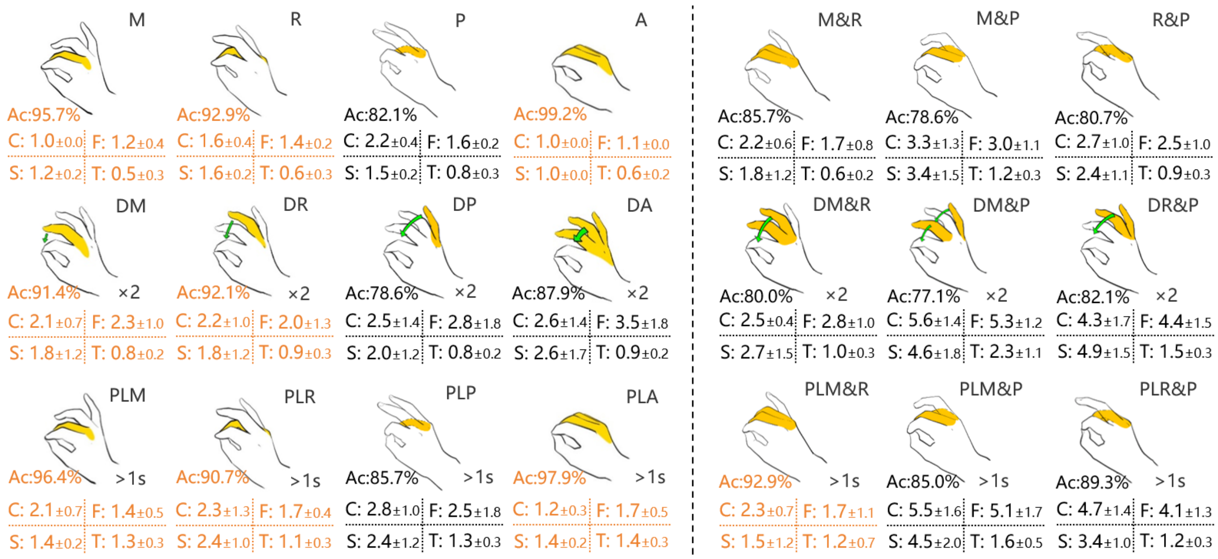


Fig. 2. Each gesture draft in the gesture set: The first row includes four basic gestures and their combination gestures, the second row includes repetitive variations corresponding to the gestures in the first row, and the third row includes variations involving prolonged holding of the gestures in the first row. Below each figure, the average subjective ratings (1–7, lower is better) of users in terms of cognitive load (C), fatigue (F), and social adaptability (S) are displayed, along with objective ratings for completion time (T) and accuracy (Ac). Gestures in the final gesture set are highlighted in orange.

Specifically, as shown in Fig. 3(a), the Oculus integration SDK provides detailed data on the 21 joint positions of the hand skeleton. By analyzing these data, we can accurately map the user's real hand movements to the virtual hand. When the index finger and thumb are detected to be pinched together and the user's hand is close to the virtual pen, the virtual pen is automatically fitted to the user's hand according to the preset hand shape, as shown in (b) and (c) in the Fig. 3. We customized and extended the default Oculus integration SDK implementation to meet the specific needs of HPIPainting, which monitors the subtle changes of each finger. As shown in Fig. 3(d), we define a key angle parameter: the tip angle θ_f. This angle is crucial for accurately capturing subtle hand gestures. For fingers $f \in middle, ring, little$ in the hand model, we calculate the angle difference between the actual finger and the predefined finger:

$$\theta_f = \arccos \frac{\vec{v}_{f_{preset}} \cdot \vec{v}_{f_{actual}}}{\left|\vec{v}_{f_{preset}}\right| \left|\vec{v}_{f_{actual}}\right|}, f \in \{middle, ring, little\} \tag{1}$$

The finger posture state s_f is defined as:

$$s_f = \begin{cases} 1, & \text{if } \theta_f < \tau_f \text{ (preset posture reached)} \\ 0, & \text{if } \theta_f \geq \tau_f \text{ (preset posture not reached)} \end{cases}, f \in \{middle, ring, little\} \tag{2}$$

By setting the threshold value of state for each finger, it is possible to determine whether the finger reaches the preset gesture position, thus achieving the effect of accurately recognising finger gestures. We define a set of pen states $S = \{s_{\text{Thumb}}, s_{\text{Index}}, s_{\text{Middle}}, s_{\text{Ring}}, s_{\text{Pinky}}\}$ where each state $s_f \in \{0, 1\}$ indicates whether the corresponding finger is in the triggered state. The pen states S is determined by the combination of triggered fingers according to the following decision function:

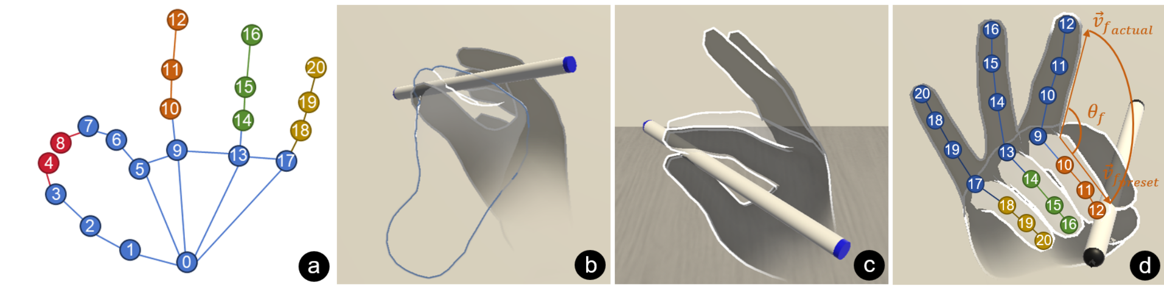


Fig. 3. Schematic diagram of micro gesture recognition algorithm. (a) shows the coordinates of 21 finger joints provided by the head-mounted display; (b) and (c) show the process of grasping a virtual pen using the index finger and thumb; (d) shows the angle between the preset finger and the actual finger.

$$S = \begin{cases} \text{“All”}, & \text{if } s_{\text{Middle}} = 1 \wedge s_{\text{Ring}} = 1 \wedge s_{\text{Pinky}} = 1 \\ \text{“M\&R”}, & \text{if } s_{\text{Middle}} = 1 \wedge s_{\text{Ring}} = 1 \wedge s_{\text{Pinky}} = 0 \\ \text{“M\&P”}, & \text{if } s_{\text{Middle}} = 1 \wedge s_{\text{Ring}} = 0 \wedge s_{\text{Pinky}} = 1 \\ \text{“R\&P”}, & \text{if } s_{\text{Middle}} = 0 \wedge s_{\text{Ring}} = 1 \wedge s_{\text{Pinky}} = 1 \\ \text{“R”}, & \text{if } s_{\text{Middle}} = 0 \wedge s_{\text{Ring}} = 1 \wedge s_{\text{Pinky}} = 0 \\ \text{“M”}, & \text{if } s_{\text{Middle}} = 1 \wedge s_{\text{Ring}} = 0 \wedge s_{\text{Pinky}} = 0 \\ \text{“P”}, & \text{if } s_{\text{Middle}} = 0 \wedge s_{\text{Ring}} = 0 \wedge s_{\text{Pinky}} = 1 \\ \text{“G”}, & \text{otherwise} \end{cases} \tag{3}$$

where $\wedge$ represents the logical operation 'AND'. Each finger's state s_i is determined by comparing the angle difference θ_f with a threshold τ_f as previously defined.

With the above functional extensions, our system recognizes the user's real hand movements and reproduces them in VR as virtual hands, enabling the determination of whether each finger reaches the preset gesture position when the user grasps the virtual pen, allowing the system to perform the corresponding movements.

3.3 Optimizing Gesture Set

3.3.1 Participants and Procedure. We recruited 14 participants (6 female, 8 male) on campus, aged 18-30 (M=22.43, SD=2.82), who could perform all microgestures. Six participants had used AR/VR devices before. They participated in Study 1 tests and assisted with brainstorming in Section 3.4. Before testing, we demonstrated all microgestures from Section 3.1 to participants. Participants repeated all gestures until they could accurately perform all microgestures. Then they wore the head-mounted display, grabbed the virtual pen in VR, and performed each gesture ten times in sequence. During gesture repetition, we recorded input time for each gesture to be correctly recognized by the system and videotaped the entire process to measure completion time and accuracy for objective assessment. Participants then rated each gesture using a 7-point Likert scale based on 3 subjective criteria. The entire experiment took about 20 minutes. Inspired by NASA-TLX, we developed the following three subjective criteria:

- **Cognitive Load (C):** The complexity and differentiation of the gesture. A lower score indicates less concentration and easier execution.
- **Fatigue (F):** The physiological burden of performing the gesture. A lower score indicates a lower level of fatigue when performing the gesture.

Table 1. The VR painting microgesture function proposed by participants during the brainstorming session.

Microgestures	Applications	Feedbacks
M/PLM	Adjusting Brush Type	This gesture is similar to the natural pen grip, and users have reported that it is ergonomic and suitable for high-frequency operations.
R/PLR	Adjusting Brush Size	This gesture is slightly more complicated than M and is suitable for fine adjustment of parameters as a secondary function. It can be paired with the M gesture.
DM/DR	Undo/Redo	Double-tapping is naturally associated with correction operations, and deliberate execution is required to reduce accidental activation. Repeating actions quickly fits the concept of 'backward' or 'forward'.
A/PLA	Start/Continue drawing	The A gesture is the most commonly used posture for holding a pen in real life and has the highest accuracy rate, making it suitable for indicating the start of drawing.
PLM&R	Call out menu	This combination of hand gestures is non-directional and clearly different from other gestures, and is suitable for accessing menus.

- **Social Adaptability (S):** The acceptability of the gesture in social situations. A lower score indicates a higher level of acceptability of the gesture in social situations.

3.3.2 Results and Discussion. We evaluated error rate, completion time and subjective evaluation of 21 gestures, selecting 9 gestures based on these factors. Gestures M, R and A are highly operable due to natural ergonomic properties. Among rejected gestures, basic gesture P and its variations received high subjective ratings with significant individual differences in execution. About 35.7% of subjects (n=14) reported that knuckle coordination difficulty exceeded expectations and required special training for accurate completion, while other subjects easily met gesture morphological requirements. This differentiation may relate to individual differences in hand anatomy.

We analyzed subjective evaluation to determine users' microgesture preferences, providing reference for functional mapping of the microgesture set in Section 3.4:

- Each basic micro-gesture scored lower than its variant, indicating users prefer simpler operations. We recommend using basic microgestures when possible.
- In first-person perspective, single-finger gestures generally have higher accuracy than combined gestures, and microgestures with more finger occlusions have lower recognition accuracy. Recognition accuracy of A and its variants is highest. However, variant DA has higher subjective scores, possibly due to high fatigue from quickly swinging three fingers.
- Multiple variants were selected, requiring care when mapping functions. Simultaneous use may cause recognition problems.

3.4 Gesture Mapping

In Section 3.3, we have built a set of microgestures, but we still need to determine how to map these gestures to the functions required by the VR painting scene. Therefore, we conducted a brainstorming session to design user-centric microgesture functions.

3.4.1 Procedure. We first showed the participants all the selected microgestures and explained why they were selected, explained the purpose of designing microgestures, and outlined the various operations that could be performed in the VR painting scene. Then, we arranged a 10-minute discussion period, during which we encouraged participants to comment on and suggest improvements to the VR painting operations we had introduced. Participants were free to speak. We then gave the participants 10 minutes to think about the one-to-one mapping between microgestures and painting operations. Each participant then took turns sharing their ideas, which we recorded. Finally, we had a group discussion to evaluate and select the most reasonable mapping scheme from the microgestures to the functions, and all participants reached a consensus on the final implementation scheme for the microgestures. The whole brainstorming session was led by three members of our team: one was responsible for leading the meeting process, while the other two were responsible for recording the participants' ideas and counting the results. The meeting lasted about an hour.

3.4.2 Results and Discussion. Brainstorming results are shown in Table 1. P7 suggested designing microgestures based on real-world painting habits to reduce learning curves, proposing A/PLA for painting functions. While P3 and P8 argued that M/PLM three-finger grip could also intuitively support painting, other participants agreed on A/PLA for painting operations. R/PLR was assigned to the frequently used brush thickness adjustment. P2 noted that M&R lacked directional distinction from other gestures, making it suitable for general functions like menu activation. P11 emphasized prioritizing frequently used operations for gesture mapping while integrating complex operations into menus, with the non-dominant hand simplifying interactions.

4 STUDY 2: HPI PARADIGM PERFORMANCE EVALUATION

To evaluate the performance of HPI paradigm in mode-switching and painting, we conducted a study comparing it with two other painting methods (Pinch and Controller). According to the conclusions of Surale et al. [46], we selected several mode-switching techniques with the best performance and combined them with the three painting methods in common configurations. Based on the conclusions in Section 3, we adopted all fingers simultaneously reaching predefined positions (A/PLA) as the baseline for HPI drawing to reveal the upper limit of mode-switching speed for HPI paradigm. The evaluation was based on four objective metrics and five subjective metrics.

4.1 Participants and Apparatus

We recruited 15 participants (8 females, 7 males) on campus, aged 18 to 33 years (M=25.13, SD=3.10). Seven of the participants had used AR or VR equipment to experience VR painting before. The reward was approximately 12 USD. A Meta Quest 2 head-mounted display (HMD) was used, and a Windows 11 PC (CPU AMD R9 7945HX, GPU GeForce GTX 4070 Laptop) ran the experiment application written in Unity 2023.2.20f1c1.

4.2 Interactive Methods

As shown in Fig. 4, we compared three drawing methods, each with two mode-switching techniques for brush color switching, to evaluate HPI performance in both mode-switching and drawing tasks.

Group 1 - HPI Baseline: Using A/PLA in Study 1 drawing as the HPI baseline:

- **Hand-Pen Interaction M/PLM (HPIM):** mode-switching activates when the middle finger reaches the preset position; drawing activates when all fingers reach preset positions simultaneously (A/PLA).
- **Non-Dominant Fist (ND-FIST):** Mode activates when the non-dominant hand forms a tight fist and deactivates when the hand relaxes.

Group 2 - Pinch Baseline: Using pinch-based drawing (index finger and thumb):

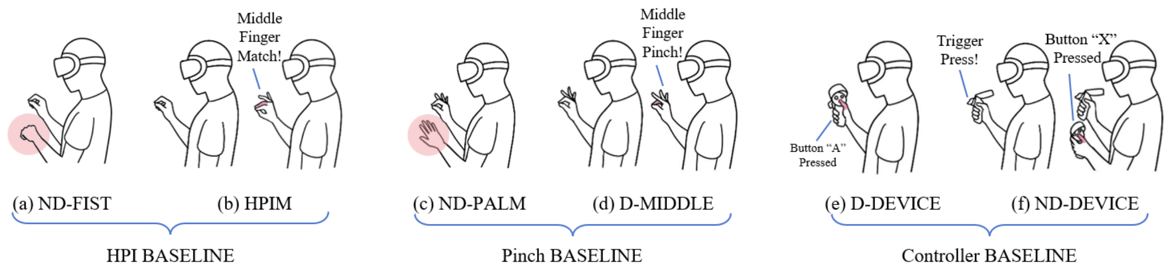


Fig. 4. Selected mode-switching techniques: (a) Non-dominant hand fist (ND-Fist) and (b) HPI M/PLM (HPIM) (BASELINE block: HPI); (c) Non-dominant hand palm (ND-Palm) and (d) Middle finger pinch (D-middle) (BASELINE block: Pinch); (e) Dominant hand device (D-Device) and (f) Non-dominant hand device (ND-Device) (BASELINE block: Controller).

- **Dominant Middle Finger Pinch (D-MIDDLE):** Mode-switching and drawing both activate when middle finger and thumb pinch together.
- **Non-Dominant Palm (ND-PALM):** Mode switches when non-dominant hand extends all fingers with palm facing up.

Group 3 - Controller Baseline: Physical controller input for comparison:

- **Non-Dominant Controller (ND-DEVICE):** Mode activates when pressing the non-dominant hand controller button (requires both hands holding controllers).
- **Dominant Controller (D-DEVICE):** Mode activates when pressing the dominant hand controller button (requires only dominant hand controller).

4.3 Task

In this study, we extended the VR line drawing task described in [46]. Considering that the application scenario of virtual pen microgestures is painting, we used drawing of straight lines, circles, and bows in 3D space as the basic tasks. Microgestures or other dominant/non-dominant hand gestures were used to change the colour of the line (blue/red), which represented different modes. We asked participants to switch modes during drawing. All tasks were completed in a standing position.

4.3.1 Line Drawing. Each trial presented five translucent sphere pairs (diameter = 30mm) in a virtual environment, with each pair representing start and end points for drawing line segments. Three line types were tested: straight lines (80mm), circles (radius = 40mm), and bows (20mm width, 70mm length), as shown in Fig. 5. Sphere positions were calibrated so participants could reach all targets without stepping, with the topmost pair at chin level.

The drawing process involved four steps: (1) place the 'ink' anchor in the starting sphere (opacity change indicates palm center inside sphere), (2) activate the technique trigger to begin drawing, (3) trace the line contour until reaching the end sphere, and (4) release the trigger to stop. Audio feedback signaled line start and end. Engagement or disengagement outside spheres triggered an error requiring redrawing. This process repeated for all five sphere pairs, with the current target highlighted and drawing direction indicated by transparency gradients. Completed pairs were hidden and saved for accuracy analysis.

4.3.2 Baseline Block and Compound Block. We designed two task modules using the subtraction method [8, 30, 41, 45]. The baseline module presented five blue sphere pairs where participants drew only blue lines using the default mode. The compound module alternated blue and red sphere pairs (always starting and ending with blue), requiring participants to use drawing techniques for blue pairs and mode-switching techniques for red pairs.

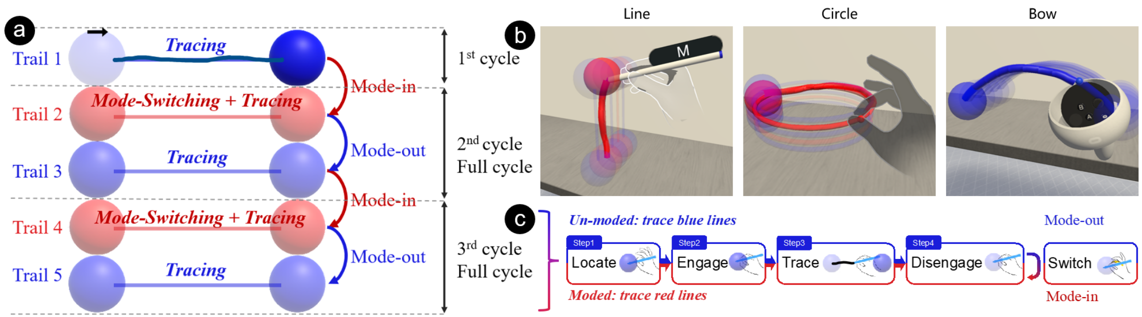


Fig. 5. Experiment setup. (a) Three cycle examples of line drawing tests in compound blocks. A complete cycle includes a complete mode-switching process: switching modes to draw moded lines, switching modes to draw un-moded lines. (b) Three line drawing tasks: The technique being used in the Line composite task is HPIM, the technique being used in the Circle compound task is D-Middle, and the technique being used in the Bow baseline task is D-Device. (c) The line-tracing process in a trial. The upper part (in blue) shows an un-moded line-drawing process, and the lower part (in red) shows a moded line-tracing process.

Mode-switching required the specified gesture before drawing, with visual feedback showing red lines for switched mode and blue for default mode. A colored indicator on the controller/fingertip displayed current mode status. Failed mode switches triggered an error, alarm sound, and required redrawing the line.

4.4 Design and Procedure

Previous studies on mode-switching typically used sphere-connecting tasks in VR. Since virtual pen microgestures are primarily used for sketching, painting, and 3D modeling, we replaced the baseline sphere-connecting task with drawing line segments (straight lines, circles, and curves in Fig. 5 (b)) to better evaluate HPIM's drawing performance. To ensure drawing accuracy measurement didn't interfere with mode-switching speed, we conducted a pre-experiment with four participants comparing separate versus simultaneous testing. Finding no significant differences, we proceeded with the simultaneous approach using a BCCCCB block design.

The formal experiment consisted of three phases: (1) introduction and demographic collection, (2) testing each technique in turn, and (3) post-experiment questionnaires. We used a within-subjects design with drawing technique as the independent variable, balanced using Latin square ordering. Each technique required five blocks (baseline-compound alternating pattern). Due to experiment duration, participants drew in only three directions: horizontal right, vertical down, and outward depth. Participants rested between techniques and completed subjective ratings after each technique, plus a final preference ranking.

This design yielded 6 techniques × 5 blocks × 6 line tests × 3 directions = 540 trials per participant.

4.5 Evaluation Metrics

The dependent variables are the mode-switching time/error rate in the compound blocks and the tracing error rate/completion time. The calculation methods for the tracing error rate, tracing completion time, mode-switching time and error types are explained in the following subsections. In addition, we used a questionnaire to collect subjective feedback from the participants, including ratings for each head gesture and a ranking of the most liked gestures. To ensure the fairness of the experiment, the same type and colour of brush was used for all experimental conditions.

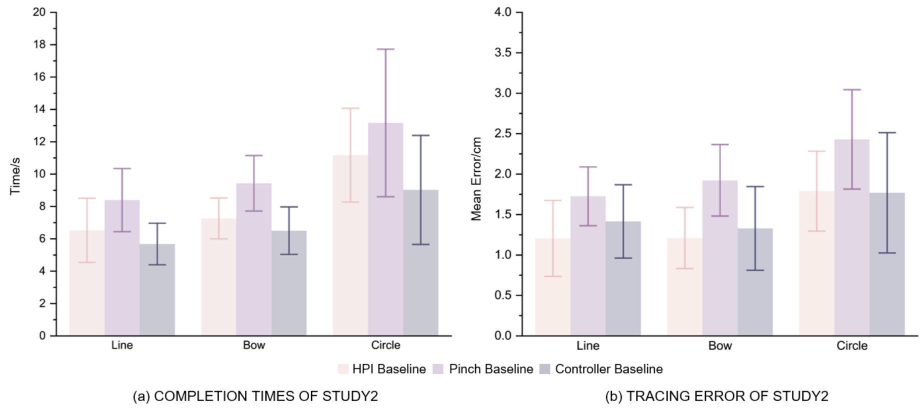


Fig. 6. (a) Bar chart showing the completion time for each BASELINE techniques in each line tracing task. (b) Bar chart showing the error rate for each BASELINE techniques in each line tracing task.

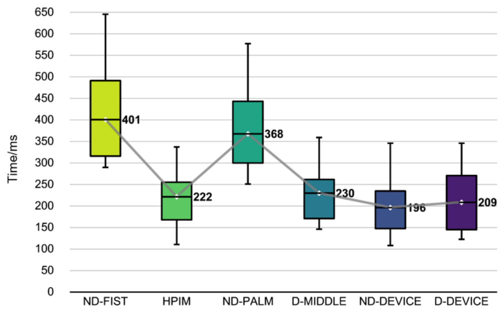
4.5.1 Tracing Error. The error of our measurement is defined as the average distance from the user-drawn point to the nearest point on the target stroke. For circle or horizontal line tasks, this can be simplified to the distance from the point to the origin of the axis or the X-axis. For other tasks, the distance from the nearest discrete point on the target stroke is used.

4.5.2 Tracing Complete Time. The time taken to draw a line graph is the time taken for the line graph in the starting sphere to finish in the finishing sphere.

4.5.3 Mode-switching Time. As shown in Fig. 5, each tile comprises three cycles. The first cycle begins when the sphere appears and ends when the trigger releases after drawing lines for the first sphere group. The second cycle immediately follows, ending when the trigger releases after drawing lines for the third sphere group. The third cycle starts immediately afterward and concludes when the trigger releases after drawing lines for the fifth sphere group. The second and third cycles constitute complete cycles [30, 46]. Each complete cycle encompasses a full mode-switching sequence: activating the specified mode-switching technique, completing the color-coded line drawing task, reverting to baseline mode, and executing the unencoded line drawing task. The initial cycle ensures the unencoded drawing task is completed before entering the complete cycle phase.

4.5.4 Mode-switching Error. Following previous research [46], we defined four error types. Start errors occur when participants initiate line drawing outside the starting sphere. End errors happen when participants release the trigger outside the designated ending sphere, failing to reach the target position. Mode entry errors result from participants' inability to draw lines on red spheres using the specified gesture, indicating failed mode activation. Mode exit errors occur when participants draw lines on blue spheres after mode-switching, indicating failure to return to baseline mode. Each error type is counted upon occurrence. Within each block, the error rate represents the ratio of errors to total trials, while the total error rate constitutes the sum of all four individual error rates.

4.5.5 Subjective Feedback. After completing each technique, participants rated the head gestures on six dimensions: ease of learning, ease of use, accuracy, speed, fatigue, and naturalness. Ratings used a 5-point continuous scale where 1 represented the most negative evaluation and 5 the most positive. These assessment criteria were adapted from previous mode-switching studies [30, 41]. Upon experiment completion, participants ranked their preferred head gestures and provided reasoning based on their VR mode-switching experience.



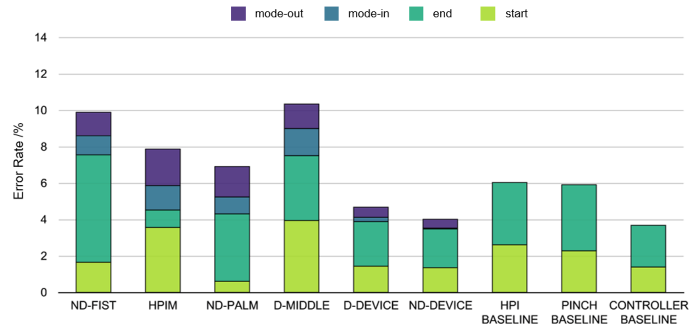


Fig. 7. (a) Box plot of mode-switching time (ms) for technique. The white diamonds represent the average time for each technique. (b) The proportion of various error rates (%, classified by techniques) is shown in a bar chart. The sum of the four error rates in a bar chart represents the overall error rate for that technique.

4.6 Results and Discussion

4.6.1 Data Pre-processing. We used Excel (MS Office 2019) and Origin 2025 Pro (version 10.2) for data processing and analysis. Before analyzing the data, we removed 1.19% of outliers from the tracing experiments. The Shapiro-Wilk test indicated that tracing error, complete time, mode-switching time, and overall mode-switching error rate were normally distributed ($P > .05$). When conducting these measurements, we used non-parametric Friedman tests with technique as the factor. If significant main effects were detected, we performed post-hoc analyses using the Wilcoxon signed-rank test and applied Bonferroni correction.

4.6.2 Tracing Error. Friedman ANOVA analysis ($\chi^2(2) = 24.13, p < .01$) and post-hoc tests revealed that HPI BASELINE achieved the lowest average error rate ($M = 1.21, SE = .05$) across all drawing tasks, compared to CONTROLLER BASELINE ($M = 1.32, SE = .07$) and PINCH BASELINE ($M = 1.92, SE = .06$). HPI demonstrated particularly superior performance in circle drawing tasks requiring high precision, significantly reducing error rates. Repeated measures ANOVA ($F_{2,21} = 13.26, p < .01$) confirmed HPI's significant advantage over PINCH technique ($p < .05$). Fig. 6 (b) presents the tracing error results for each baseline.

4.6.3 Tracing Complete Time. Analysis revealed significant performance differences across input techniques. Repeated measures ANOVA indicated that while Controller achieved the shortest completion time ($M = 6.49s, SE = .19$), HPI ($M = 7.25s, SE = .16$) demonstrated superior accuracy-speed balance. Friedman ANOVA analysis ($\chi^2(2) = 24.13, p < .01$) confirmed significant differences among techniques, with post-hoc tests revealing significant differences between HPI and Pinch ($M = 9.42s$, $SE = .22$) ($p < .01$). Although HPI showed slightly longer completion times than Controller, its significantly improved operational accuracy, particularly in linear drawing tasks (with only approximately 1.5s time difference), positions HPI as a more balanced solution for interactive scenarios requiring precise input. Fig. 6 (a) presents the tracing time results for each baseline.

4.6.4 Mode-switching Time. The main effect of techniques on mode-switching time was highly significant ($F_{5,56} = 19.14$, $p < .01$, $\eta^2G = .58$). Descriptive statistics revealed that ND-FIST ($M = 401ms$, $SE = 25.98$) and ND-PALM ($M = 368ms$, $SE = 23.64$) had significantly longer switching times than other modes, while ND-DEVICE achieved the shortest switching time ($M = 196ms$, $SE = 18.04$). No significant differences emerged between HPIM and D-MIDDLE. Mode-switching times ranked from fastest to slowest: ND-DEVICE (196 ms), D-DEVICE (209 ms), HPIM (222 ms), D-MIDDLE (230 ms), ND-PALM (368 ms), and ND-FIST (401 ms). These results demonstrate

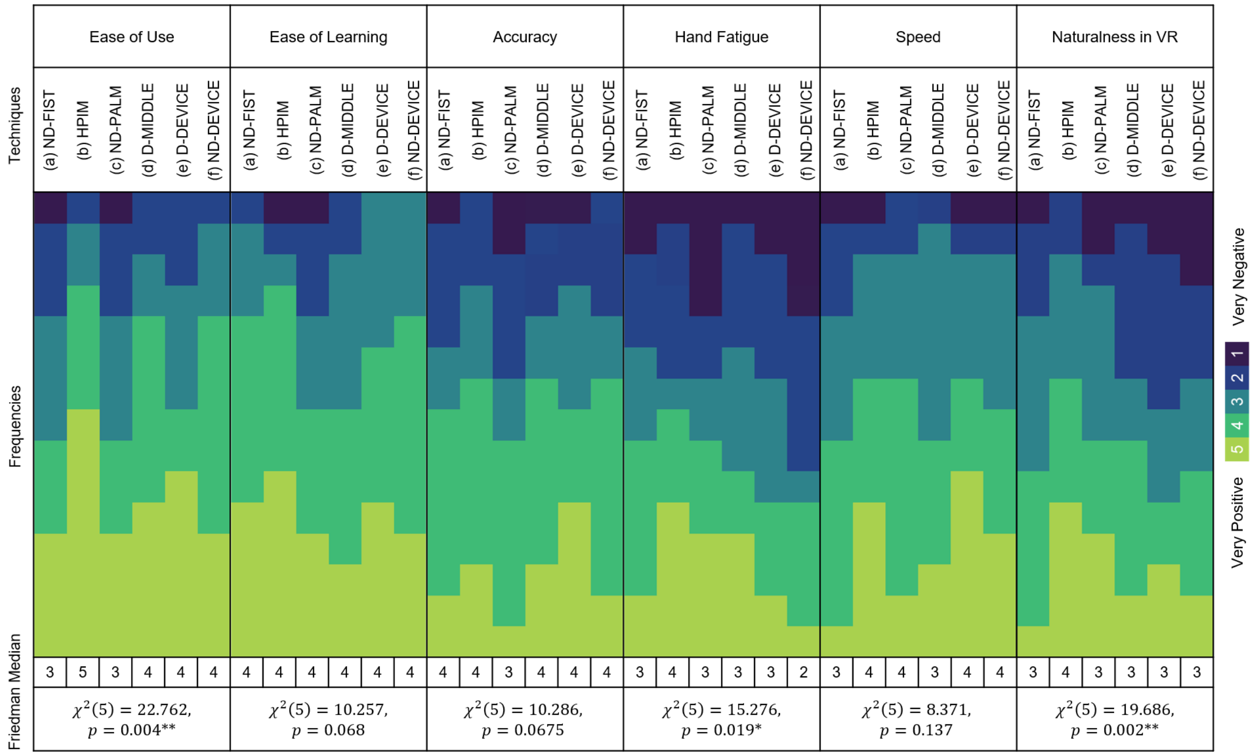


Fig. 8. Subjective ratings and Friedman test results indicate that there are qualitative differences between different techniques (* and ** represent significant results at 0.05 and 0.01, respectively).

that HPIM represents an effective dominant-hand technique and ranks among the fastest methods. Fig. 7 (a) presents the mode-switching time results for each technique.

4.6.5 Mode-switching Error Rates. Error rates for HPI, pinch, and controller baseline tasks ranged from 3.5%to 6.1%, while mode-switching techniques ranged from 4% to 10.2%. Most techniques showed comparable error rates, except ND-FIST, which exhibited higher end-error rates than dominant-hand gestures. Technique type (TECHNIQUE) showed no significant effect on most error types, except end error rate. Repeated measures ANOVA revealed a significant main effect of technique type on end error rate ($F_{5,56} = 4.27$, $p < .01$, $\eta^2 G = .90$). HPIM demonstrated strong overall performance with an end mode-exit rate (3.83%) slightly above average but within acceptable ranges compared to mainstream techniques, and a combined error rate (7.77%) lower than D-MIDDLE (10.24%) and ND-FIST (9.89%). These results indicate that HPIM, as a newly proposed mode-switching technique, achieves error rates comparable to existing methods while demonstrating potential advantages in specific error types, particularly for interactive scenarios requiring frequent and precise mode-switching. Fig. 7 (b) presents the mode-switching error rates for each technique.

4.6.6 Subjective Feedback. Fig.8 presents subjective rating results. Friedman tests revealed significant differences among techniques for usability ($\chi^2(5) = 22.75, p = .004$), hand fatigue ($\chi^2(5) = 15.28, p = .019$), and VR naturalness ($\chi^2(5) = 19.69, p = .002$), while no significant differences emerged for accuracy ($p = .0675$), ease of learning ($p = .068$), and speed ($p = .137$). Post-hoc tests showed that HPIM (median = 5) significantly

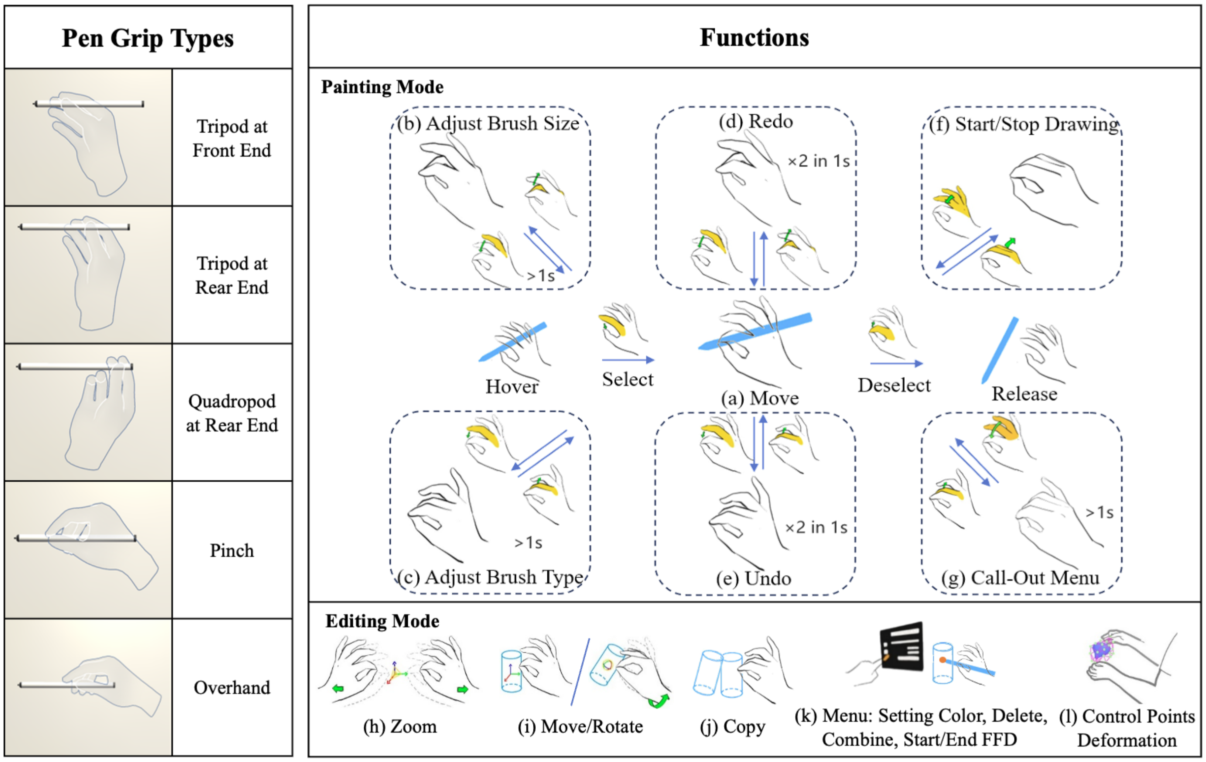


Fig. 9. Design concept of the HPIPainting system: The first column refers to the research by Li et al. [29] and sets up multiple pen grip types to meet the needs of different users, including Tripod at Front End, Tripod at Rear End, Tripod at Rear End, Pinch, and Overhand. The second column is the function column. We divide the painting functions into two operating modes. When the user picks up the virtual pen, it is in painting mode, and all user interactions are centred around the virtual pen. When the user puts down the virtual pen, it is in editing mode, and all user interactions are centred around the sketches.

outperformed ND-FIST in usability ($p < .01$); HPIM (median = 4) significantly outperformed ND-DEVICE in hand fatigue ($p < .01$); and HPIM (median = 4) excelled in VR naturalness, showing significant differences compared to ND-DEVICE ($p < .01$), D-DEVICE ($p < .01$), and ND-FIST ($p < .05$). These results indicate that while techniques performed similarly in speed and ease of learning, HPIM provided superior user experience in key metrics, particularly usability and natural VR interaction, demonstrating clear overall advantages. Fig. 8 shows the evaluation results for each technique across all metrics.

5 HPIPainting SYSTEM DESIGN

In Study 2, we evaluated HPI paradigm performance in mode-switching and simple line drawing, but utilized only 2 of the 9 designed microgestures. This inadequately represents HPIPainting VR usage scenarios. Based on Section 3 conclusions, we propose a novel 3D painting system that integrates primary painting interactions into the dominant hand's virtual pen, rather than using the dominant hand for painting assistance. Beyond these core features, our system incorporates a smart guidance interface to enhance usability and implements simplified single-point Free-Form Deformation (FFD) technology to support direct hand-object interaction and clay sculpting-like modeling.

Considering diverse user pen-holding postures, we referenced Li et al.'s experiments [29] and predefined multiple gripping gestures, as shown in the Pen Grip Types on the right side of Fig. 9. We use 'Tripod at Front End' as the default gesture:

- **Tripod at Front End:** Common three-point grip at pen's front end for precise writing/drawing;
- **Tripod at Rear End:** Three-point grip at pen's rear end, commonly used for pointing tasks.
- **Quadropod at Rear End:** Four-point support using index finger, middle finger, thumb, and ring finger for stability;
- **Pinch:** Grip between thumb and fingers with pen held parallel to writing surface;
- **Overhand:** Relaxed grip enabling versatile pen control including rolling, sliding, and flipping movements.

5.1 Painting Mode

Based on Study 1 findings, we implemented painting mode functions corresponding to microgestures, as shown in Fig. 9. When grasping the virtual pen, users can perform the following painting operations in the right column: **(a) Grab virtual pen:** Pinching with thumb and index finger; **(b) Adjusting brush size:** Ring finger reaching preset gesture position for more than one second; **(c) Adjusting brush type:** Ring finger reaching predefined gesture position for more than one second; **(d) Redo:** Ring finger reaching predefined gesture position twice within one second; **(e) Undo:** Middle finger reaching predefined gesture position twice within one second; **(f) Start/Continue drawing:** All fingers reaching predefined gesture position; **(g) Call out menu:** Middle finger and ring finger simultaneously reaching preset gesture position for more than one second.

5.1.1 Brush Type Design. To create realistic VR handwriting, we implemented an extrusion method similar to 3D printing pens (Fig. 10(a)). The system samples pen tip coordinates each frame and extrudes content when the distance between consecutive points exceeds a threshold. We explored several rendering approaches: Unity's Linerenderer with sprite materials produced distorted curves without mesh (Fig. 10(b)(1)); manual 2D line segment meshes became invisible at certain angles (Fig. 10(b)(2)); ultimately, we developed 3D tubular line segments with complete grids, vertices, and triangular faces that remain visible from all angles and support mesh deformation for complex shapes (Fig. 10(b)(3)). We also added a geometry painting brush (Fig. 10(c)).

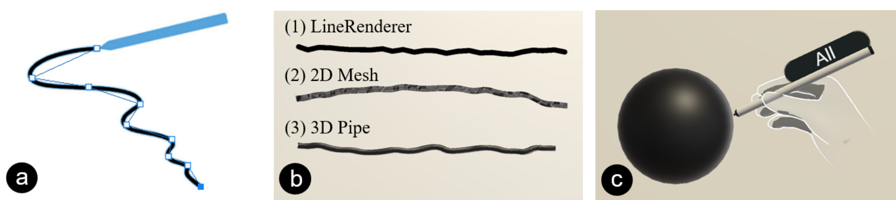


Fig. 10. Handwriting generation: (a) shows how we generate handwriting points in VR. (b) shows the three handwriting effects we explored; where (1) was generated for Linerenderer, (2) for self-generated 2D mesh grid line segments, and (3) for generated 3D pipe-like handwriting. (c) Demonstrates how to draw geometric shapes, including spheres, rounded cubes, cubes, cylinders, and capsules. The shapes can be drawn proportionally or non-proportionally.

5.1.2 Jittering Solution. Due to the inevitable hand jitter, the drawn line segments will produce obvious fluctuation phenomenon. In order to solve the jitter problem, various handwriting recognition methods and denoising fitting algorithms have been explored, where the RANSAC(Random Sample Consensus) algorithm has been used for the fitting of circles, the Principal Component Analysis (PCA) algorithm has been used to optimize straight lines, and in other cases Bezier curves have been used for the smoothing of drawn handwriting. The optimized results are shown in the Fig. 11 below.

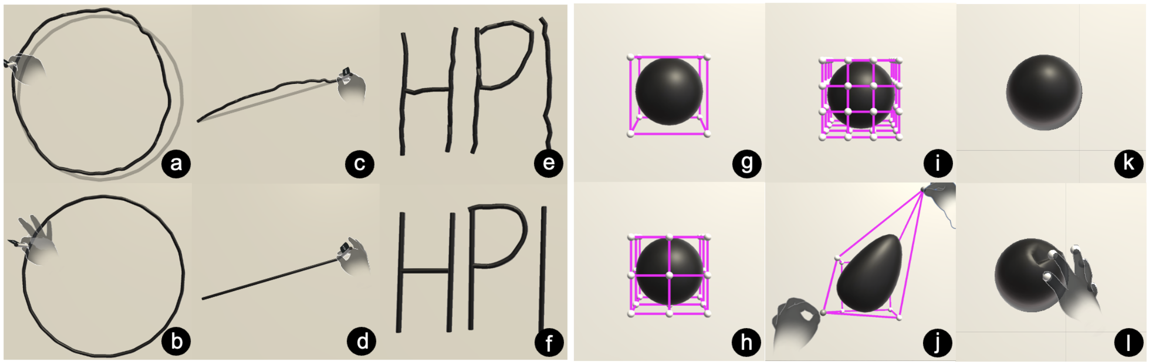


Fig. 11. Handwriting fitting and editing, a) b) is the comparison of handwriting after applying circle fitting algorithm, c) d) is the comparison of handwriting after applying straight line fitting, e) f) is the comparison of handwriting before and after applying Bezier curve. g)-i) are FFD2×2×2, FFD3×3×3, FFD4×4×4, in that order; j) is the FFD2×2×2 deformation effect; k) l) shows the before and after effects of pinch modify.

CircleFit. We apply a RANSAC-based algorithm to detect circular patterns in handwritten point clouds. The algorithm randomly samples 3 points to determine circle parameters, then evaluates model fit by calculating point-to-circle distances. After multiple iterations, we select the most consistent solution. Our detection considers four factors: F1) proximity between start and end points; F2) overall fitting error; F3) sufficient point distribution range relative to circle diameter; and F4) minimum point count threshold (typically 8 points). This approach efficiently distinguishes intentional circles from other curved patterns.

LineFit. We use PCA to detect if user-drawn points approximate a straight line. After calculating the center of mass and covariance matrix, we find eigenvalues and eigenvectors. The principal direction is the eigenvector with the largest eigenvalue. When the ratio of largest eigenvalue to eigenvalue sum exceeds our threshold, we identify the pattern as a line. We then define this line using center of mass and principal direction, calculate point-to-line distances, and compute root mean square error. If this error is below threshold, we replace points with the fitted line. Our implementation uses power iteration to find the principal eigenvector efficiently.

Bezier curves. Bezier curves are used in our system to smooth curves. A degree n curve uses $n + 1$ control points with parametric equation $P(t) = \sum_{i=0}^{n} B_i^n(t)P_i$, where $B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$ for $t \in [0, 1]$. We implement cubic Bezier curves ($n = 3$) with four control points and use De Casteljau's algorithm for evaluation. Connected Bezier segments create complex shapes with smooth strokes that maintain quality from any viewing angle.

5.2 Editing Mode

In addition to these painting mode functions, we have added some editing mode functions to make HPIPainting more comprehensive. When users release the virtual pen, the following operations in Fig. 9(right) can be performed: **(h) Zoom:** Pinch gesture control; **(i) Move/Rotate:** Pinch gesture control; **(j) Copy:** Pinch with the middle finger and thumb while moving an object; **(k) Menu: Setting color, delete, combine, start/end FFD:** Select functions by clicking the menu with the non-dominant hand, then tap the object with the pen tip to apply the function. **(l) Control FFD points for deformation:** Pinch gesture to select points.

To enable HPIPainting to create complex 3D sketches, we incorporated various Free-Form Deformation (FFD) models into the VR interaction environment. Additionally, we developed a simplified, single-point-based FFD method for direct hand-object interaction, which produces intuitive deformation effects similar to clay modeling. We named this interaction technique the "Pinch Modifier."

5.2.1 Pinch Modifier. To achieve intuitive hand interaction and realistic deformation effects in Fig. 11(f), we implement a sphere collider-based approach for detecting hand-mesh contact. Small sphere colliders are attached to each finger joint position of the virtual hand. During each frame, we monitor these sphere colliders for intersections with the target mesh. When a collision occurs, the intersection point serves as input to our Pinch Modifier algorithm, which applies a localized deformation around that point.

5.2.2 Free Form Deformation. We employ the Free-Form Deformation (FFD) algorithm to achieve intuitive deformation of 3D mesh models. This technique establishes an n×n×n grid of control points around the target mesh (e.g., 2×2×2, 3×3×3, or 4×4×4), as shown in Fig. 11(g-i). Users can directly manipulate these control points via hand interaction to deform the mesh. The algorithm first creates a three-dimensional control grid and calculates the relative coordinates of each vertex within the grid. In the shader, trilinear interpolation is used to determine the new position of each vertex based on the stored local coordinates and the current control point positions. After calculating the new vertex positions, we recalculate the normals and bind the updated vertex data to the mesh material to render the deformation effect.

5.3 Smart Guidance Interface Design

Beyond implementing HPIPainting functions, we consider user interface design to guide novice users and enable quick onboarding. We designed a smart guidance feedback interface for this purpose.

5.3.1 Sketch Management and Menu Features. With virtual pen microgesture mapping, users can draw straight lines, curves, circles, and geometries in 2D and 3D space. Drawing processes require frequent sketch combination and position adjustment, necessitating sketch capture capabilities and secondary menu functions. Users can mitigate 3D drawing deviations by snapping sketches to canvas using both hands.

Users do not require constant menu visibility. We designed toggleable menu display where users interact through fingertip clicking or dragging, manipulating buttons, drop-down menus, sliders, and interface elements. HPIPainting provides buttonless interaction for basic drawing with immediate sketch capture and instant feedback when fingertips touch menu buttons. We consolidated drawing influence operations into a drop-down menu where users select influence type, then click the affectable toggle, causing a trigger ball to appear at the pen tip. When the trigger ball contacts target sketches, corresponding operations execute. Common editing functions including color changes, copying, deletion, and merging are integrated into this module.

5.3.2 Visual and Auditory Feedback. Since we propose a bare-hand painting approach, we cannot provide haptic vibration feedback. As shown in Fig. 12(a-c), when users hold the virtual pen, a status bar appears at the pen's end indicating current pen state. This bar disappears and produces audio tones when the virtual pen is released or grabbed.

When users enable automatic fitting, the system detects handwriting in real time and displays fitted results for circles or straight lines, as shown in Fig. 11(a,c).

In Fig. 12(g), when drawing on 2D canvas, the pen remains on the surface while users hear friction sounds. Stroke size increases with hand penetration distance, enabling pressure-sensitive drawing with visual and auditory feedback. Fig. 12(h) shows color switching via the palette menu, with selected colors immediately displayed on the pen tip and cap. Fig. 12(a-c) shows the visual feedback of sketch grabbing process.

6 STUDY 3: HPIPainting PERFORMANCE EVALUATION

In this study, users had the opportunity to freely create and edit meshes in virtual space using HPIPainting. The user's performance was then compared with the sketches drawn using Gravity sketch. The main focus of this study was to explore the user experience of constructing 3D objects in VR using HPIPainting. As Gravity sketch

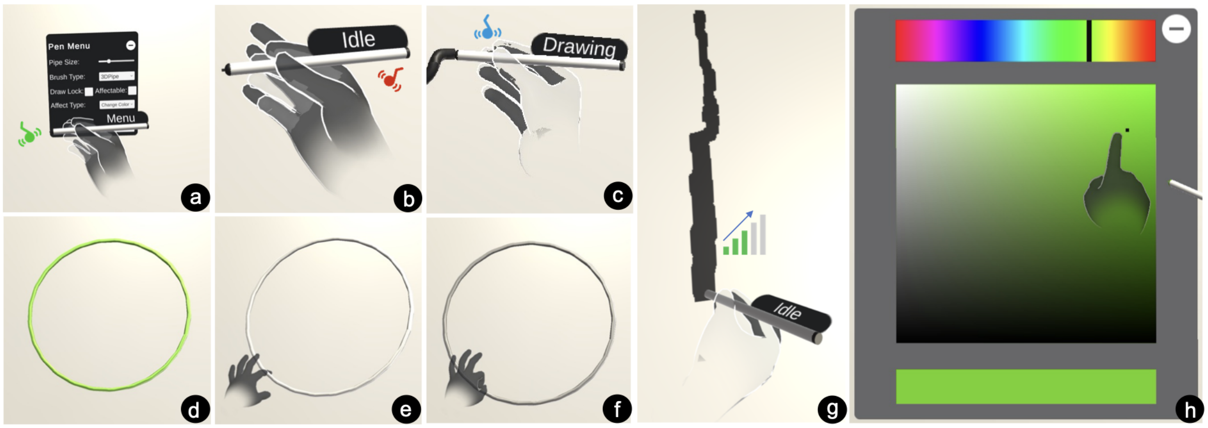


Fig. 12. Visual and auditory feedback: The virtual pen emits different tones when switching states: (a) calling menu, (b) releasing/grabbing pen, (c) starting to draw. Color feedback indicates object interaction: (d) original color, (e) hover color, (f) selected color. (g) The pen maintains surface contact with the blackboard while sensing hand pressure to adjust line width; (h) pen color changes match the color palette selection.

has not released any open source version, we were unable to unify the stroke settings. The device used in this study was the same as in Study 2.

6.1 Participants

We recruited 12 participants (5 females, 7 males) aged between 19 and 32 years (M=25.58, SD=3.58) on campus. Seven participants had little or no VR experience, while five had experience with VR painting software, and one of them was a professional modeler.

6.2 Task

In this study, participants freely created objects of varying complexity. We presented target patterns and asked participants to draw these patterns shown in Fig. 13 using HPIPainting and Gravity Sketch, following a method similar to Tan et al. [48]. This approach proves more beneficial than simply asking participants to reproduce 3D objects, as it enables observation of their ability to independently use virtual pens for three-dimensional construction while providing insights into their drawing planning process. We selected three common drawing examples involving everyday objects from different categories with moderate complexity, suitable for the study's time constraints. These objects comprise simple shapes and lines, representing many common drawing targets. Fig. 13 shows the target objects, which feature clear structures and are easily observable.

6.3 Procedure

We first familiarised participants with the standard drawing features of HPIPainting and Gravity Sketch, followed by several practice rounds to prepare them for the formal task. The experiment required participants to complete all tasks using HPIPainting and Gravity Sketch, alternating between the two technologies to maintain balance. A ten-minute break was provided between each technology switch, with the entire process averaging approximately 1.5 hours per participant. Timing began when participants pressed any key to indicate readiness and ended when they actively pressed the 'Submit' button. After completing the task, participants were asked to complete the NASA-TLX and rate the following statements using a 5-point Likert scale (1: strongly disagree; 5: strongly agree):

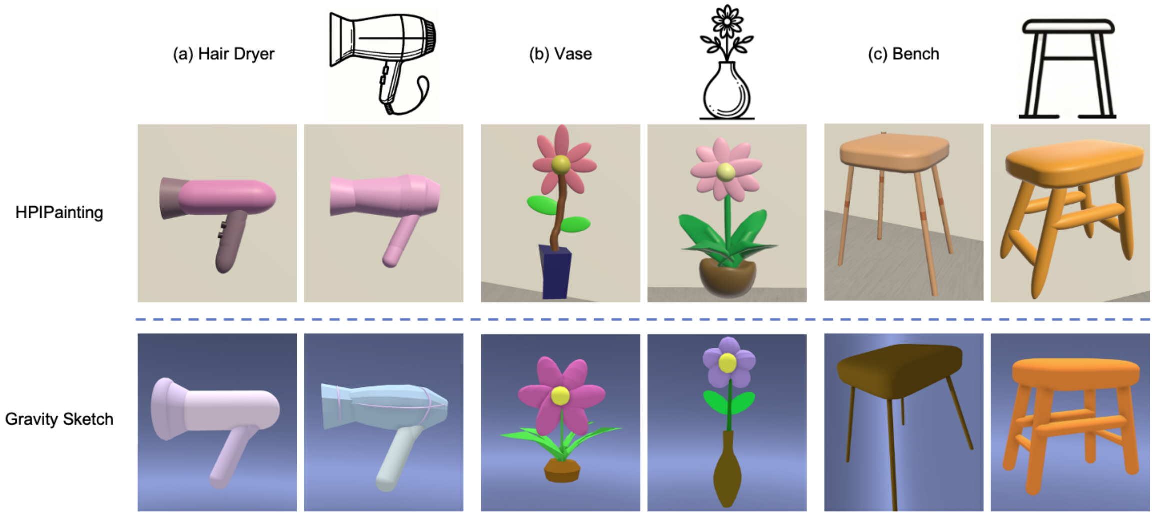


Fig. 13. Top is a sketch of the objects to be drawn (hair dryer, vase and bench), and bottom is a painting example created using HPIPainting and Gravity Sketch in Study 3.

1) The interaction method was natural and intuitive; 2) The system helped me express the shape I wanted correctly; 3) The system was not tiring to use; 4) The drawing function was easy to use; 5) The drawing function was easy to control. Participants were also asked to evaluate the user interface design. Finally, they were interviewed about their experience using HPIPainting and asked to provide further feedback on their evaluations. The focus of the interviews was on their expectations for further improvements.

6.4 Results

We analyzed completion times of 12 participants using two systems, HPIPainting and Gravity Sketch, to draw three common 3D objects (bench, hairdryer, and vase in Fig. 13). For simple bench modeling tasks, HPIPainting (M = 1.83 min, SD = 0.81 min) and Gravity Sketch (M = 1.68 min, SD = 0.74 min) performed similarly, with no significant difference (p = 0.289). However, when modeling slightly more complex objects, differences between systems became more pronounced. In hairdryer and vase modeling tasks, HPIPainting (M = 4.62 min, SD = 1.63 min) took significantly longer than Gravity Sketch (M = 3.27 min, SD = 1.23 min) (t = 6.77, $p < 0.01$). The completion time comparison for each task between the two systems is shown in Fig. 14(b).

These results are expected, as HPIPainting uses bare-hand control input and remains in the demo stage, requiring more control input and detailed steps to define shape features. In contrast, the commercially mature Gravity Sketch has been optimized over many years, with controller inputs and a refined interaction system that enables participants to draw complex shapes more smoothly, reducing completion time by an average of approximately 1.2 minutes (equivalent to 27%-29% of HPIPainting time).

6.4.1 User Generated Paintings. Our research data, as shown in Fig. 13, indicates that HPIPainting performs comparably to the commercially mature Gravity Sketch in modeling accuracy. This finding is encouraging, considering that Gravity Sketch, as mature commercial modeling software, has been optimized over many years, making it remarkable that HPIPainting, as a research prototype, achieves similar accuracy levels. Sample analysis reveals that both systems support participants in creating structurally sound and proportionally balanced 3D models, though they differ in workflow and interaction methods. Gravity Sketch users completed tasks more

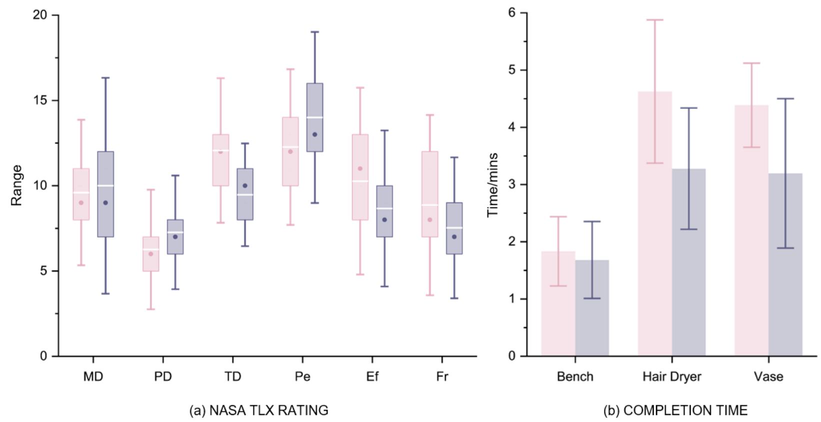


Fig. 14. (a) Overall task rating for TLX. (b) Bar chart showing completion times for three objects for both systems.

quickly, while HPIPainting users reported more intuitive control during certain fine-grained operations. These results demonstrate HPIPainting's competitiveness in core modeling functionality and establish groundwork for future optimizations that balance precision with enhanced user experience.

6.4.2 Likert-scale Ratings. We used the NASA Task Load Index (NASA-TLX) to assess the user experience differences between the HPIPainting prototype system and the commercially mature Gravity Sketch across six dimensions. Twelve participants completed the same task and rated both systems, with results analysed using paired t-tests.

As shown in Fig. 14(a), in mental demand (MD), effort (Ef), and frustration (Fr) dimensions, HPIPainting ($MD : M = 9.6, SD = 2.13; Ef : M = 8.47, SD = 2.85; Fr : M = 8.87, SD = 2.64$) and Gravity Sketch ($MD : M = 10.0, SD = 3.16; Ef : M = 8.67, SD = 2.29; Fr : M = 7.53, SD = 2.01$) showed no significant differences ($MD : t = -0.38, p = 0.71; Ef : t = 1.48, p = 0.16; Fr : t = 1.65, p = 0.12$). This indicates that although HPIPainting remains in the prototype stage, it has achieved levels comparable to commercial software in cognitive load, required effort, and user frustration.

For physical demands (PD), Gravity Sketch ($M = 7.27, SD = 1.67$) was significantly higher than HPIPainting ($M = 6.21, SD = 1.75$), with a significant difference ($t = -2.18, p < 0.05$). This finding suggests that HPIPainting has an advantage in reducing users' physical strain, likely due to its more ergonomic interaction input.

For time demands (TD), data shows that HPIPainting ($M = 12.07, SD = 2.12$) is significantly higher than Gravity Sketch ($M = 9.47, SD = 1.51$), with a significant difference ($t = 3.05, p < 0.01$). This indicates that users perceive greater time pressure when using HPIPainting, reflecting room for improvement in the prototype system's workflow and interaction design optimization.

For performance evaluation (Pe), HPIPainting ($M = 12.27, SD = 2.28$) was significantly lower than Gravity Sketch ($M = 14.0, SD = 2.51$), with a significant difference ($t = -2.15, p < 0.05$). This indicates that participants rated their task performance lower when using HPIPainting, which may be related to the system's insufficient maturity.

6.4.3 User Feedback. Participants' feedback provided additional insights. Several users noted: "HPIPainting offers precise bare-hand control, allowing me to accomplish painting tasks with both hands, which is more intuitive and immersive, though the workflow is slightly cumbersome" (P3), and "When creating geometric shapes, HPIPainting performs similarly to Gravity Sketch, but I prefer how HPIPainting interacts with my hands" (P5). P7 commented:

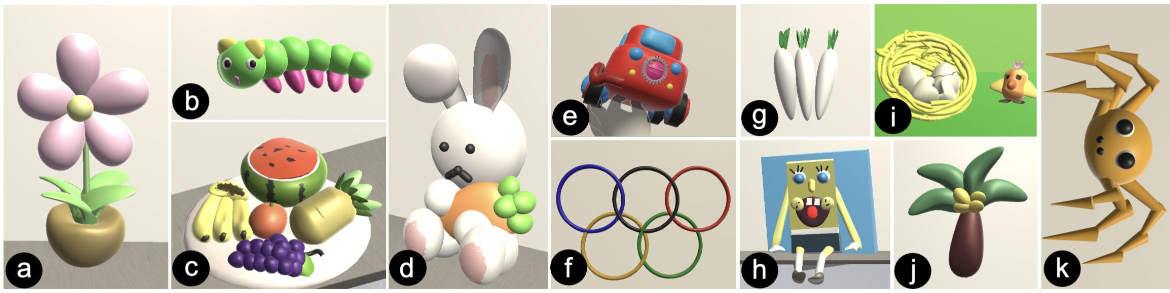


Fig. 15. A portfolio of freely created works using HPIPainting: (a) 23 min, 67 steps; (b) 18 min, 44 steps; (c) 33 min, 102 steps; (d) 30 min, 98 steps; (e) 35 min, 101 steps; (f) 5 min, 15 steps; (g) 10 min, 33 steps; (h) 28 min, 73 steps; (i) 29 min, 89 steps; (j) 15 min, 39 steps; (k) 31 min, 67 steps; where (h) is a well-known cartoon character created by the youngest volunteer child, P1, aged 11.

"I was impressed with the ability to manipulate FFD by hand in VR" and expressed hope for future VR modeling work. However, participants highlighted HPIPainting limitations. P8 commented: "HPIPainting offers immersive workflow for both hands when handling models, but has functional flaws, while Gravity Sketch offers smoother workflow." P2 added: "Gravity Sketch is more mature and feature-rich, saving unnecessary steps. For example, Gravity Sketch allows manipulating mesh faces, which HPIPainting does not." P10 mentioned, "Gravity Sketch has a steeper learning curve, requiring memorization of complex key mappings, but once mastered, the precision is impressive." Regarding time differences, P4 explained: "HPIPainting requires more steps to complete the same task, which can be frustrating, especially when creativity is flowing." P9 noted: "Gravity Sketch allows me to work faster, but HPIPainting lets me control the model directly with my hands—it's a trade-off."

Results indicate that as a research prototype, HPIPainting is on par with commercial software in key experiential dimensions, particularly in reducing physical requirements. However, it lags in interactive feature complexity and fluidity. Future iterations should prioritize optimizing time efficiency and enhancing user performance while maintaining natural interaction advantages to enhance overall competitiveness.

6.4.4 User Experience Comparison. We analyzed performance differences between VR painting experienced and novice users:

Usability: Experienced users showed shorter adaptation periods (19 vs. 32 minutes) and higher completion quality in complex tasks, though no differences existed for basic shapes. Video analysis revealed experienced users made fewer gesture corrections (12.3 vs. 16.7) and used quick gestures more frequently.

Interaction satisfaction: Experienced users scored lower on NASA-TLX, particularly in psychological needs (2.3 points lower) and frustration (1.7 points lower). They better identified system differences from commercial tools, while novices focused on learning difficulty. Both groups praised reduced physical fatigue compared to commercial software.

Generated paintings: Experienced users produced models with better spatial utilization and structural complexity, while novices used simple geometric assemblies.

6.5 Painting Activity: Exploring Creative Freedom with HPIPainting

In this experiment, six participants (3 males, 3 females, average age 19.5, including two teenagers aged 11 and 14) experienced HPIPainting's creative potential over five days. All had VR experience and were encouraged to create freely without restrictions.

We collected VR artworks and creation process recordings shown in Fig. 15. P3 commented: "HPIPainting's flexible gesture interaction made my creation process smooth and enjoyable. I created a cartoon rabbit using geometric modeling, then added dynamic gestures with mesh deformation tools." Participants explored diverse styles: P6 created cartoon fruit plates demonstrating fine modeling capabilities, P5 created a hatching chicken, and child P1 created familiar cartoon characters.

The creations demonstrated HPIPainting's flexibility in supporting diverse artistic expressions from cartoon to realistic styles. Participants noted that intuitive bare-handed interaction allows focus on artistic expression without cumbersome tool operations, proving HPIPainting's potential as a promising VR creation platform.

7 CONCLUSION AND FUTURE WORK

In this paper, we present HPIPainting, a bare-hand VR painting system built upon a HPI paradigm that replaces physical controllers with a virtual pen driven by microgestures. HPIPainting features hand-pen interaction where 9 low-amplitude microgestures were selected from 21 candidates based on cognitive load, fatigue, social adaptability, recognition accuracy, and time demand. These microgestures map to VR painting functions including brush switching, thickness adjustment, mode toggling, geometric primitive creation, and mesh manipulation. HPIPainting integrates microgesture recognition with smart guidance interface, mirroring real-world painting workflows: when users "grasp" the virtual pen, the system enters Painting Mode enabling bare-hand creation of lines and geometric shapes; when users "release" the pen, it switches to Editing Mode for intuitive object pose, scale, and deformation adjustment. By aligning digital interaction with familiar painting processes, HPIPainting reduces users' cognitive load significantly.

We validated HPIPainting through three user studies. Study 1 confirmed optimal gesture set and function mapping. Study 2 showed HPIPainting achieves controller-level drawing accuracy (mean error 1.21 mm), fast mode-switching within 222 ms, and significantly higher ratings for usability, hand fatigue, and immersion versus pinch- and controller-based input. Study 3 demonstrates that although HPIPainting is 27-29% slower than Gravity Sketch for complex tasks, it achieves comparable modeling precision, lower physical demand, and natural bare-hand interaction experience. These findings show HPIPainting meets professional precision requirements while lowering cognitive and physiological barriers and delivering controller-free creative experience in VR.

Despite these strengths, HPIPainting exhibits limitations. Commercial software offers extensive toolsets, layered workflows, and advanced mesh editing capabilities that our prototype lacks. HPIPainting currently supports only core painting and editing operations without features like color mixing palettes, brush libraries, undo history management, or complex scene import/export.

Looking ahead, we plan to extend HPIPainting in three directions. First, expand the toolset and optimize workflow to match commercial VR painting suites, adding layered canvases, advanced mesh sculpting, color palettes, and scene saving. Second, enrich feedback with audio and granular visual affordances to increase immersion. Third, explore adaptive gesture models, extend microgesture operations to objects, and collaborative multi-user scenarios. These enhancements will bring HPIPainting closer to a full-featured, accessible, and engaging VR creative platform.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2022ZD0118001, the National Natural Science Foundation of China under Grant 62332017 and U22A2022. We thank all participants for their valuable time in our user studies. The computing work is partly supported by USTB MatCom of Beijing Advanced Innovation Center for Materials Genome Engineering and this research was conducted under the ethical approval of the Research Ethics Committee of University of Science and Technology Beijing (Protocol A9700671).

References

- [1] Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh. 2018. SymbiosisSketch: Combining 2D & 3D Sketching for Designing Detailed 3D Objects in Situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3173574.3173759
- [2] Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George Fitzmaurice. 2017. Experimental Evaluation of Sketching on Surfaces in VR. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 5643–5654. doi:10.1145/3025453.3025474
- [3] Ilhan Aslan, Andreas Uhl, Alexander Meschtscherjakov, and Manfred Tscheligi. 2014. Mid-air Authentication Gestures: An Exploration of Authentication Based on Palm and Finger Motions. In *Proceedings of the 16th International Conference on Multimodal Interaction* (Istanbul, Turkey) *(ICMI '14)*. Association for Computing Machinery, New York, NY, USA, 311–318. doi:10.1145/2663204.2663246
- [4] Doug A. Bowman, Donald B. Johnson, and Larry F. Hodges. 1999. Testbed evaluation of virtual environment interaction techniques. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (London, United Kingdom) *(VRST '99)*. Association for Computing Machinery, New York, NY, USA, 26–33. doi:10.1145/323663.323667
- [5] Nicola Capece, Carola Gatto, Gilda Manfredi, Gabriele Gilio, Benito Luigi Nuzzo, Lucio Tommaso De Paolis, and Ugo Erra. 2023. Enhancing Art Therapy with Virtual Reality and Hand Gesture Recognition: A Case Study in Pottery Modeling. In *Extended Reality*, Lucio Tommaso De Paolis, Pasquale Arpaia, and Marco Sacco (Eds.). Springer Nature Switzerland, Cham, 210–226.
- [6] Li-Wei Chan, Hui-Shan Kao, Mike Y. Chen, Ming-Sui Lee, Jane Hsu, and Yi-Ping Hung. 2010. Touching the void: direct-touch interaction for intangible displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 2625–2634. doi:10.1145/1753326.1753725
- [7] Andrea Colaço, Ahmed Kirmani, Hye Soo Yang, Nan-Wei Gong, Chris Schmandt, and Vivek K. Goyal. 2013. Mime: compact, low power 3D gesture sensing for interaction with head mounted displays. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 227–236. doi:10.1145/2501988.2502042
- [8] R. F. Dillon, Jeff D. Edey, and Jo W. Tombaugh. 1990. Measuring the true cost of command selection: techniques and results. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '90)*. Association for Computing Machinery, New York, NY, USA, 19–26. doi:10.1145/97243.97247
- [9] Tobias Drey, Jessica Janek, Josef Lang, Dietmar Puschmann, Michael Rietzler, and Enrico Rukzio. 2022. SpARklingPaper: Enhancing Common Pen- And Paper-Based Handwriting Training for Children by Digitally Augmenting Papers Using a Tablet Screen. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 3, Article 113 (Sept. 2022), 29 pages. doi:10.1145/3550337
- [10] John J. Dudley, Hendrik Schuff, and Per Ola Kristensson. 2018. Bare-Handed 3D Drawing in Augmented Reality. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. Association for Computing Machinery, New York, NY, USA, 241–252.
- [11] Sergio Escalera, Jordi Gonzàlez, Xavier Baró, Miguel Reyes, Oscar Lopes, Isabelle Guyon, Vassilis Athitsos, and Hugo Escalante. 2013. Multi-modal gesture recognition challenge 2013: dataset and results. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction* (Sydney, Australia) *(ICMI '13)*. Association for Computing Machinery, New York, NY, USA, 445–452. doi:10.1145/2522848.2532595
- [12] Facebook. 2016. Quill. https://quill.fb.com/.
- [13] Google VR. 2020. Tilt Brush README. https://github.com/googlevr/tilt-brush. original-date: 2020-04-13T18:55:08Z.
- [14] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1063–1072. doi:10.1145/2556288.2557130
- [15] Yushen Hu, Keru Wang, Yuli Shao, Jan Plass, Zhu Wang, and Ken Perlin. 2024. Generative Terrain Authoring with Mid-air Hand Sketching in Virtual Reality. In *Proceedings of the 30th ACM Symposium on Virtual Reality Software and Technology* (Trier, Germany) *(VRST '24)*. Association for Computing Machinery, New York, NY, USA, Article 26, 10 pages. doi:10.1145/3641825.3687736
- [16] Apple Inc. 2025. Apple Vision Pro. https://www.apple.com/apple-vision-pro/.
- [17] HTC Inc. 2025. Vive Pro 2. https://www.vive.com/cn/.
- [18] Microsoft Inc. 2025. Hololens 2. https://learn.microsoft.com/zh-cn/hololens/.
- [19] Meta Inc. 2025. Quest. https://about.meta.com/technologies/meta-quest.
- [20] PICO Inc. 2025. PICO 4. https://www.picoxr.com/global/products/pico4.
- [21] Ajune Wanis Ismail, Fazliaty Edora Fadzli, and Muhammad Syahmirulfitri Mohd Faizal. 2021. Augmented Reality Real-time Drawing Application with a Hand Gesture on a Handheld Interface. In *2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA)*. 418–423. doi:10.1109/ICCCA52192.2021.9666439
- [22] Jason Jerald, Joseph J. LaViola, and Richard Marks. 2017. VR interactions. In *ACM SIGGRAPH 2017 Courses* (Los Angeles, California) *(SIGGRAPH '17)*. Association for Computing Machinery, New York, NY, USA, Article 19, 105 pages. doi:10.1145/3084873.3084900

[23] Ying Jiang, Congyi Zhang, Hongbo Fu, Alberto Cannavò, Fabrizio Lamberti, Henry Y K Lau, and Wenping Wang. 2021. HandPainter -3D Sketching in VR with Hand-based Physical Proxy. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 412, 13 pages. doi:10.1145/3411764.3445302

[24] Daniel Keefe, Robert Zeleznik, and David Laidlaw. 2007. Drawing on Air: Input Techniques for Controlled 3D Line Illustration. *IEEE Transactions on Visualization and Computer Graphics* 13, 5 (2007), 1067–1081. doi:10.1109/TVCG.2007.1060

[25] Daniel F. Keefe, Daniel Acevedo Feliz, Tomer Moscovich, David H. Laidlaw, and Joseph J. LaViola. 2001. CavePainting: a fully immersive 3D artistic medium and interactive experience. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics (I3D '01)*. Association for Computing Machinery, New York, NY, USA, 85–93. doi:10.1145/364338.364370

[26] Taejun Kim, Youngbo Aram Shim, Youngin Kim, Sunbum Kim, Jaeyeon Lee, and Geehyuk Lee. 2024. QuadStretcher: A Forearm-Worn Skin Stretch Display for Bare-Hand Interaction in AR/VR. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 409, 15 pages. doi:10.1145/3613904.3642067

[27] Laan Labs. 2024. AirDraw - Finger Paint. https://apps.apple.com/us/app/airdraw-finger-paint/id6477749286. Version 1.0.4.

[28] Gierad Laput, Robert Xiao, and Chris Harrison. 2016. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 321–333. doi:10.1145/2984511.2984582

[29] Nianlong Li, Teng Han, Feng Tian, Jin Huang, Minghui Sun, Pourang Irani, and Jason Alexander. 2020. Get a Grip: Evaluating Grip Gestures for VR Input using a Lightweight Pen. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3313831.3376698

[30] Yang Li, Ken Hinckley, Zhiwei Guan, and James A. Landay. 2005. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. Association for Computing Machinery, New York, NY, USA, 461–470. doi:10.1145/1054972.1055036

[31] Jenny Lin, Xingwen Guo, Jingyu Shao, Chenfanfu Jiang, Yixin Zhu, and Song-Chun Zhu. 2016. A virtual reality platform for dynamic human-scene interaction. In *SIGGRAPH ASIA 2016 Virtual Reality Meets Physical Reality: Modelling and Simulating Virtual Humans and Environments* (Macau) *(SA '16)*. Association for Computing Machinery, New York, NY, USA, Article 11, 4 pages. doi:10.1145/2992138.2992144

[32] Sally A. Linkenauger, Heinrich H. Bülthoff, and Betty J. Mohler. 2015. Virtual arms reach influences perceived distances but only after experience reaching. *Neuropsychologia* 70 (2015), 393–401. https://api.semanticscholar.org/CorpusID:27839922

[33] Yang Liu, Chengdong Lin, and Zhenjiang Li. 2021. WR-Hand: Wearable Armband Can Track User's Hand. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3, Article 118 (Sept. 2021), 27 pages. doi:10.1145/3478112

[34] Mayra D. Barrera Machuca, Paul Asente, Wolfgang Stuerzlinger, Jingwan Lu, and Byungmoon Kim. 2018. Multiplanes: Assisted Freehand VR Sketching. In *Proceedings of the 2018 ACM Symposium on Spatial User Interaction* (Berlin, Germany) *(SUI '18)*. Association for Computing Machinery, New York, NY, USA, 36–47. doi:10.1145/3267782.3267786

[35] Mark R. Mine. 1995. *Virtual Environment Interaction Techniques*. Technical Report. USA.

[36] Mu Mu and Murtada Dohan. 2021. Community Generated VR Painting using Eye Gaze. In *Proceedings of the 29th ACM International Conference on Multimedia* (Virtual Event, China) *(MM '21)*. Association for Computing Machinery, New York, NY, USA, 2765–2767. doi:10.1145/3474085.3478552

[37] Jing Qian, Tongyu Zhou, Meredith Young-Ng, Jiaju Ma, Angel Cheung, Xiangyu Li, Ian Gonsher, and Jeff Huang. 2021. Portalware: Exploring Free-Hand AR Drawing with a Dual-Display Smartphone-Wearable Paradigm. In *Proceedings of the 2021 ACM Designing Interactive Systems Conference* (Virtual Event, USA) *(DIS '21)*. Association for Computing Machinery, New York, NY, USA, 205–219. doi:10.1145/3461778.3462098

[38] E. Sachs, A. Roberts, and D. Stoops. 1991. 3-Draw: a tool for designing 3D shapes. *IEEE Computer Graphics and Applications* 11, 6 (1991), 18–26. doi:10.1109/38.103389

[39] Steven Schkolne, Michael Pruett, and Peter Schröder. 2001. Surface drawing: creating organic 3D shapes with the hand and tangible tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '01)*. Association for Computing Machinery, New York, NY, USA, 261–268. doi:10.1145/365024.365114

[40] Holger Schultheis and Anthony Jameson. 2004. Assessing Cognitive Load in Adaptive Hypermedia Systems: Physiological and Behavioral Methods. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*. https://api.semanticscholar.org/CorpusID:6514356

[41] Rongkai Shi, Nan Zhu, Hai-Ning Liang, and Shengdong Zhao. 2021. Exploring Head-based Mode-Switching in Virtual Reality. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 118–127. doi:10.1109/ISMAR52148.2021.00026

[42] Yinghan Shi, Lizhi Zhao, Xuequan Lu, Thuong Hoang, and Meili Wang. 2023. Grasping 3D Objects With Virtual Hand in VR Environment. In *Proceedings of the 18th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry* (Guangzhou, China) *(VRCAI '22)*. Association for Computing Machinery, New York, NY, USA, Article 8, 8 pages. doi:10.1145/3574131.3574428

[43] Alexa F. Siu, Mike Sinclair, Robert Kovacs, Eyal Ofek, Christian Holz, and Edward Cutrell. 2020. Virtual Reality Without Vision: A Haptic and Auditory White Cane to Navigate Complex Virtual Worlds. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3313831.3376353

[44] Jungah Son and Misha Sra. 2021. Exploring Emotion Brushes for a Virtual Reality Painting Tool. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology* (Osaka, Japan) *(VRST '21)*. Association for Computing Machinery, New York, NY, USA, Article 69, 3 pages. doi:10.1145/3489849.3489925

[45] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 2017. Experimental Analysis of Mode Switching Techniques in Touch-based User Interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3267–3280. doi:10.1145/3025453.3025865

[46] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 2019. Experimental Analysis of Barehand Mid-air Mode-Switching Techniques in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3290605.3300426

[47] Jeremy Sutton. 2013. Air painting with Corel Painter Freestyle and the leap motion controller: a revolutionary new way to paint!. In *ACM SIGGRAPH 2013 Studio Talks* (Anaheim, California) *(SIGGRAPH '13)*. Association for Computing Machinery, New York, NY, USA, Article 21, 1 pages. doi:10.1145/2503673.2503694

[48] Xiaohui Tan, Zhenxuan He, Can Liu, Mingming Fan, Tianren Luo, Zitao Liu, Mi Tian, Teng Han, and Feng Tian. 2024. WieldingCanvas: Interactive Sketch Canvases for Freehand Drawing in VR. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 725, 16 pages.

[49] Tijana Vuletic, Alex Duffy, Laura Hay, Chris McTeague, Gerard Campbell, and Madeleine Grealy. 2019. Systematic literature review of hand gestures used in human computer interaction interfaces. *Int. J. Hum.-Comput. Stud.* 129, C (sep 2019), 74–94. doi:10.1016/j.ijhcs.2019.03.011

[50] E. Wiese, J. H. Israel, A. Meyer, and S. Bongartz. 2010. Investigating the learnability of immersive free-hand sketching. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium* (Annecy, France) *(SBIM '10)*. Eurographics Association, Goslar, DEU, 135–142.

[51] Jiaqi Xie, Hao Zheng, Junxian Lin, and Xiaoying Tang. 2024. Digital Gamification Design of Chinese Landscape Painting Based on Gesture Interaction. In *HCI in Games*, Xiaowen Fang (Ed.). Springer Nature Switzerland, Cham, 127–142.

[52] Renqiang Xie and Juncheng Cao. 2016. Accelerometer-Based Hand Gesture Recognition by Neural Network and Similarity Matching. *IEEE Sensors Journal* 16, 11 (2016), 4537–4545. doi:10.1109/JSEN.2016.2546942

[53] Xue Yu, Stephen DiVerdi, Akshay Sharma, and Yotam Gingold. 2021. ScaffoldSketch: Accurate Industrial Design Drawing in VR. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 372–384. doi:10.1145/3472749.3474756

[54] Nabil Zerrouki, Fouzi Harrou, Amrane Houacine, Riadh Bouarroudj, Mohammed Yazid Cherifi, Ait-Djafer Amina Zouina, and Ying Sun. 2024. Deep Learning for Hand Gesture Recognition in Virtual Museum Using Wearable Vision Sensors. *IEEE Sensors Journal* 24, 6 (2024), 8857–8869. doi:10.1109/JSEN.2024.3354784