Metaverse (2021) Volume 2 Issue 2, *13 pages*.
doi: 10.54517/met.v2i2.2104

ORIGINAL RESEARCH ARTICLE

Research on object placement method based on trajectory recognition in Metaverse

Ke Lang, Xiaoying Nie, Yongjian Huai*, Yuanyuan Chen

School of Information, Beijing Forestry University, Beijing 100083, China. E-mail: huaiyj@163.com

ABSTRACT

Many studies focus on only one aspect while placing objects in virtual reality environment, such as efficiency, accuracy or interactivity. However, striking a balance between these aspects and taking into account multiple indicators is important as it is the key to improving user experience. Therefore, this paper proposes an efficient and interactive object placement method for recognizing controller trajectory in virtual reality environment. For creating user-friendly feedback, we visualize the intersection of the ray and the scene by linking the controller motion information and the ray. The trajectory is abstracted as point-clouds for matching, and the corresponding object is instantiated at the center of the trajectory. To verify the interactive performance and user satisfaction with this method, we carry out a study on user experience. The results show that both the efficiency and interaction interest are improved by applying our new method, which provides a good idea for the interactive design of virtual reality layout applications.

Keywords: objects placement; trajectory recognition; handheld controller

1. Introduction

With the development of digitization, the Metaverse opens a new digital era for us with a real sense of experience. The continuous progress of interactive technology and the deepening of interactive perception provide a virtual reality experience ladder for the Metaverse[1]. It promotes the innovation and development of virtual reality interaction methods, and also puts forward higher functional requirements for virtual reality applications. Object placement is a basic and important function in virtual reality games and tools. It is important to design an efficient and interactive way to place objects. At present, the placement of virtual objects is mainly based on VR handheld controller and hand tracking without controller. According to a recent article released by RoadToVR[2], the monthly-connected VR headsets on Steam have surpassed 3.4 million in 2022. We can see a 29.46% increase compared to the same period last year, which shows the widespread use and continued growth of VR headsets. The most popular commercial VR headsets, such as Oculus Quest and HTC Vive, all have in common that they use handheld controllers as the standard interaction mode. Therefore, we expand the research of object placement method based on handheld controller.

Now, we have two major ways to place objects in VR by using the controller. Use a ray to target the position or directly grasp the object when the user

ARTICLE INFO
Received: May 10, 2021 | Accepted: June 20, 2021 | Available online: July 6, 2021

CITATION
Lang K, Nie X, Huai Y, et al. Research on object placement method based on trajectory recognition in Metaverse. Metaverse 2021; 2(2): 13 pages.

COPYRIGHT
Copyright © 2021 by author(s). *Metaverse* is published by Asia Pacific Academy of Science Pte. Ltd. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), permitting distribution and reproduction in any medium, provided the original work is cited.

can reach it. Both of these methods need to interact with an object list to select which object needs to be placed. It can be annoying when the user wants to switch between different kinds of objects frequently. And when the types of objects increase, the menu will get more layers. This can make the interaction process inefficient.

On the other hand, human-computer interaction is one of the seven levels of the Metaverse technology framework, interactivity has become a key issue in related technology research[3]. The immersion, space and virtuality in the concept of the Metaverse[4] have spawned the innovation of interaction forms, and people's high requirements for interaction have also promoted the development of the Metaverse[5]. Interactivity is one of the main features of VR devices compared to traditional digital devices. It is helpful to improve user experience by designing reasonable interaction modes that promote user movement and increasing the interaction between users and the environment. This is consistent with the interactivity effects model[6], which revealed positive effects of interactivity on engagement, attitude and behavioral intention. In addition, previous study showed that playfulness would also impact user's sense of interactivity[7], which reflects the degree of enjoyment that users perceive when interacting with VR technology. Traditionally, users use the dropdown list to select different items. That means users need to repeat the same action, which is not fun and easily tires the user.

In order to solve these problems, we propose an object placement method based on trajectory recognition of the controller. By recognizing the movement trajectory of the controller, different objects can be selected (see **Figure 1**). This can avoid frequent interaction with the menu, allowing users to place objects in the VR scene anywhere. The concept of controller trajectory recognition in our method is consistent with many virtual reality games stimulating the user's body movements to mobilize the enthusiasm for interaction. Our method only needs one button, which acts as a switch for starting trajectory recognition. Without operating additional controller buttons, users can focus on the interaction itself, increasing the user experience and interactive fun in VR.

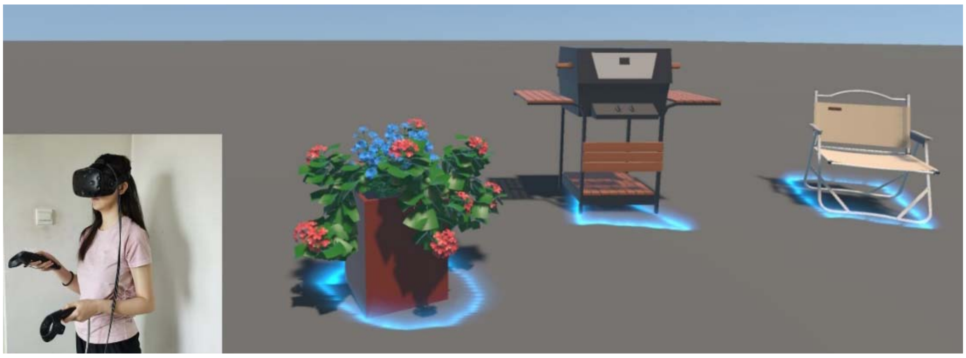

Figure 1. Placement objects are recognized in the center of the trajectories.

Finally, we compared our method with another existing method and performed statistical analysis. The quantitative data and user feedback demonstrated the effectiveness and usability of our method. We also investigated user satisfaction by constructing an experience evaluation system. The main contributions of this paper can be summarized as follows:

(1) We propose an object placement method based on trajectory recognition of handheld controller.
(2) Apply 2D gesture recognition method to VR and allow users to customize layout trajectory.
(3) To explore the performance of our approach both unilaterally and comprehensively, we conduct a comparative user study and construct a user experience evaluation system by combining Analytic Hierarchy Process (AHP) with fuzzy comprehensive evaluation.

2. Related works

2.1. Trajectory analysis for gesture recognition

Gesture recognition by analyzing trajectories was first studied in 2D. The $-family recognizer is a typical representative, which is based on simple approximate spatial registration. It is very popular for 2D touch-based gestural interface design as it allows rapid prototyping of specific recognizers[8]. For example, $1 and Protractor can handle single-stroke gestures[9,10]. $N and $N Protractor can recognize multi-stroke gestures[11,12]. But considering the stroke order and direction, the memory usage will be too high due to the arrangement and combination of stored gestures. The appearance of $P[13] fills the defects of these algorithms. $P+[14] is an improvement to $P, which considers the connection relationship of consecutive points and improves the accessibility of touch screens for people with low vision. $Q[15] further improves the recognition speed on the basis of $P.

As more and more 3D input devices enable users to acquire gestures in space, many researches are gradually transferred to gesture recognition in 3D space. Sven et al.[16] extended and modified $1[17] to recognize 3D gestures, using a scoring heuristic to reduce false positives. The uWave recognizer, Protractor3D, and the 3¢ have also been proposed successively[18–20]. Mehdi et al.[21] not only proposed $F and FreeHandUni, but also extended four 2D stroke recognizers to 3D.

At present, many virtual reality games and applications also adopt the concept of trajectory recognition. *MageVR* allows players to use spells in an immersive way by recognizing VR actions. The function of drawing glyphs for the backpack adds to the immersive experience. *VRIK Player Avatar* designs a gesture input system that allows users to bind actions to each hand, reducing the need to open menus. Many VR gesture recognition plugins have also been developed in Unity Engine. *VR infinite gesture* uses a neural network to train in detecting gestures done while interacting with the system with an HTC Vive controller. *VR Magic Gesture AI* by Raving Bots, takes the gesture performed by the user as input to a neural network (MLP) and uses backpropagation training algorithm to recognize gestures. *VR Gesture and Signature* by AirSig receives top marks for accuracy compared to the competition.

2.2. Object placement and arrangement

Creation and placement of scene objects are two important steps to author an immersive virtual environment (IVE). Many previous studies have explored 3D modeling in VR. In this study, we focus on the placement and arrangement of objects.

Sevinc et al.[22] proposed a free-hand 3D interaction technique for efficient placement and selection of virtual objects on a 2D panel. The right index finger emits a ray perpendicular to the panel. The ray is drawn in three colors according to the distance from the fingertip to the panel, corresponding to Select, Add, and Idle mode. In Add mode, touch the cursor to add objects to the virtual environment. Mine[23] proposed a snap-to-grid object placement method in ISAAC, a scene synthesis application for interactive virtual world construction. Objects are placed on evenly spaced grid intersections, and the snap point is defined as the center of the object bounding box. The user can adjust the grid size independently using the slider in the control panel. Burns et al.[24] proposed a gaze-based object placement method within a VR environment, casting the gaze point ray outward from the view position. Virtual objects are placed at the intersection between the projected gaze radius and the VR environment. Mapes et al.[25] adopted a two-hand interaction technique realized by tracking data gloves to build a network virtual environment, which allows placing and arranging objects in VR.

Procedural generation methods have also been used for object placement in virtual scenes. Peter et al.[26] proposed a new method to rapidly generate furniture arrangements in indoor scenes. The selection and arrangement of furniture objects in a room are optimized according to aesthetic and functional rules. Next, the procedural methods are locally applied in a stochastic fashion to generate important scene details. Interactive methods[27–29] also realize the filling of furniture in virtual scenes by pre-defining furniture configuration rules. Hao et al.[30] designed a new mixed-reality (MR) shop system that combines virtual in-store characteristics and real environments. A new spatial understanding algorithm and layout mechanism was developed to support responsive spatial layout.

Another approach to object placement is to employ multimodal interaction techniques. *VR Designer*[31], allows users to query the repository through voice and display the query results on the menu. Fine-grained adjustments can be made to individual objects in the properties panel when objects are placed, including modifying the location. *Wonderland Builder*, presented by Barot et al.[32], couples speech and tracks hand input commands to create and manipulate objects in VR.

3. Methods

The key to the object placement method based on trajectory recognition of controller is how to recognize the trajectory and determine the position of the object in the virtual space. $P Point-Cloud Recognizer takes a template recognition approach for gesture recognition based on approximate spatial registration. $P has the characteristics of low cost, fast recognition speed and high recognition accuracy, which is enough to meet the target requirements of our method. Furthermore, it provides an interface for rapid prototyping of single-stroke and multi-stroke recognition. Therefore, we conduct research on object placement methods based on $P Point-Cloud Recognizer. In this method, the position coordinate of input device is used as the basis of trajectory recognition. When this method is extended to VR, the input is converted from the 2D stroke of the mouse to the 3D motion information of the controller. According to whether the controller moves and whether the user triggers the controller button, the trajectory layout process is divided into three phases. Next, we will introduce the main tasks of three phases in detail and show the entire object placement process (see **Figure 2**).

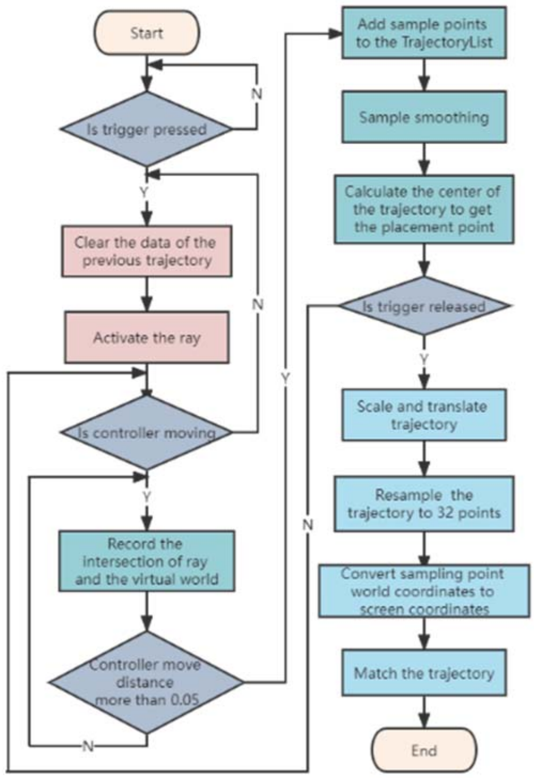

Figure 2. Object placement algorithm process based on trajectory recognition.

3.1. Preparation phase

The preparation phase is marked by pressing down the button, but the controller position has not changed. The main goal of this phase is to clear the input data of the previous trajectory and prepare for the input of the current trajectory. We define a button on the controller as a switch for trajectory recognition. To avoid user operation by mistake, we set an input threshold to monitor button triggering. When the user triggers the button above the threshold, the controller ray is activated. The coordinate information of the collision point between the ray and the scene is recorded in real time through the ray detection callback, so as to prepare for the determination of the placement point in the next phase.

3.2. Placement point determination phase

The placement point is determined by the user moving the controller and obtaining its position coordinates. We calculated the displacement of the controller in the two frames, that is, whether the point is recognized as a sampling point through space-time sampling. Points that change less than a set threshold are considered inadvertent jitter when the user interacts with the controller in hand. These points can be ignored to reduce the number of trajectory sampling points (see **Figure 3**).

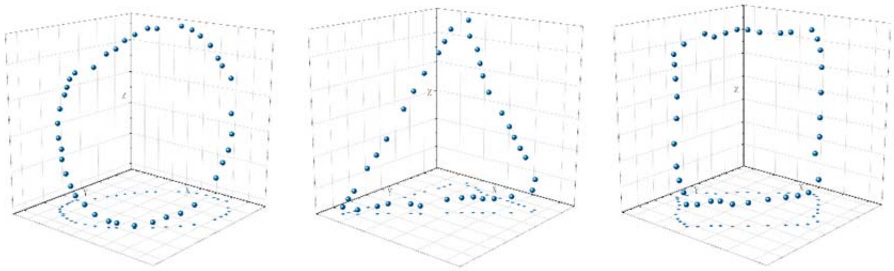

Figure 3. The visualization of three initial trajectory sampling.

The placement point can be determined after trajectory sampling is complete. We set the placement point as the center of the trajectory and calculate the average value of all points sampled on the trajectory. However, when the user activates the ray, the coordinates of the point where the ray collides with the scene are recorded. It is easy to generate unnecessary outliers and cause the offset of the center of the trajectory. To improve the accuracy of target placement points, we use statistical outlier removal method for smoothing. Sparse outlier removal is based on the calculation of the distance distribution from the point to its neighbors in the input data. The k-nearest neighbor method is used to calculate the average distance from each point to all its neighbors. Points whose average distance is outside the standard range (defined by the global mean distance and standard deviation) can be defined as outliers and removed from the sample set to improve the quality of the collected data.

3.3. Trajectory recognition phase

The user releases the controller button to indicate the end of trajectory input. $P Recognizer for trajectory preprocessing based on processing chains. The input trajectory is first scaled to the reference size and the shape remains [0..1]x[0..1]. After scale normalization is completed, the center of the trajectory is then calculated and shifted to the origin of the coordinates. For easy comparison with template, the trajectory is resampled by a fixed number of points at equal intervals. The trajectory is abstracted into 32 sampling points by calculating the interval length and adding interpolation points.

It is time to match the template trajectory. Both the input trajectory and the template trajectory are resampled to the same number of points. Then select the starting point randomly and set the step size to calculate the Euclidean distance in multiple groups. For each point in the input point cloud, find the nearest point that has not yet been matched from the template point cloud. The order of match points determines the contribution of the point, because the matching is selected from the entire point set. As the algorithm progresses, few options remain for the points in later order from the set when searching their closest pair. Therefore, it is necessary to set a weight for each matching point. The weights encode the confidence of the selected points according to the matching order. The first point has the largest weight, while the last point is the least trusted. Since the world coordinates obtained are 3D coordinates, the $P can only match the information of X and Y dimensions. To solve the problem, we convert the world coordinates of these points to screen coordinates. Although the $P can also be extended to 3D, one more dimension greatly increases the difficulty of matching and the 2D recognition method has been tested to meet the layout requirements. Finally, the different trajectories are recognized and the corresponding objects are instantiated at the placement points.

3.4. User-defined trajectory

Human-centered immersive interaction design focuses on the human experience. In order to enhance the fun of interaction and the autonomy of scene creation, we provide the function of user-defined trajectory. Users customize trajectory through interaction with the 2D panel and draw different trajectories in the scene to represent different virtual objects. When the creation mode is enabled, the trajectory can be modified in real time (see **Figure 4**). User-defined trajectories are saved as template trajectories for matching with input trajectories during recognition. The user-defined trajectory is saved as the template for matching with the input trajectory during the recognition phase. For user-friendly visual feedback, we visualize collision points with particles rendering model. visualization helps users understand the trajectory of their design. The particles appear for three seconds and then disappear in the order they were drawn.

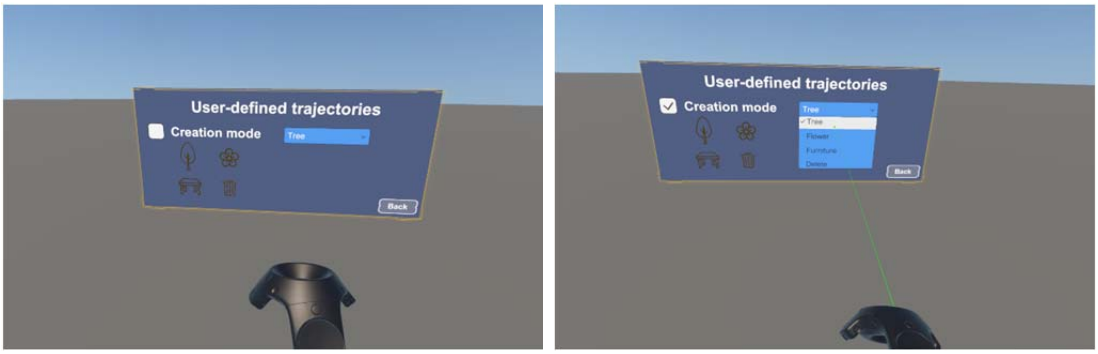

Figure 4. User-defined trajectory 2D menu: Recognition mode (left) and Creation mode (right).

4. User study

Our goal is to improve placement efficiency and enhance interaction playfulness, and the precise location of placement points is necessary for scene authoring. Therefore, we think accuracy, efficiency, and playfulness are the three main related aspects. We compared our method with the existing method of selecting objects from a menu and assigning them to the intersection of ray and scene. This existing method is called the direct placement method, and our method is called the indirect placement method. The purpose of the comparative study is to investigate whether our method improves interactive performance and provides a better user experience.

Therefore, we proposed the following hypothesis:

H1: Indirect and direct placement method have similar high accuracy.

H2: Indirect placement method is more efficient than direct placement method.

H3: Indirect placement method is more interesting than direct placement method.

4.1. Study design and tasks

In order to verify **H1** to **H3**, we created a digital garden as a task scene and selected common landscape elements in gardens as placement objects. For this, we carried out three main tasks, two focused on individual features and one on environment creation. As much as possible, we kept the object placement method as the only variable and minimized the influence of other variables.

Task 1 for accuracy (see **H1**). The user activated the controller ray and placed 4 chairs that need to be instantiated into 4 designated locations marked with the “X” icon. The two designated points were farther from the user, and the other two were closer. We recorded the distance between four chairs placed by participants and the marker in either way. The step of selecting a chair from the menu using the direct placement method was set up before the task, because Task 1 was concerned with the accuracy of object placement rather than menu interaction.

Task 2 for efficiency (see **H2**). The difference from Task 1 is that the user needs to place four different objects. One of the main concerns of the indirect placement method is whether the efficiency is improved when the placement object changes. The complete time of the two methods reflect the placement efficiency.

Task 3 for playfulness (see **H3**). For this, participants were asked to replicate an outdoor scene with seven landscape elements. This task was a synthesis of the first two tasks and was used to explore the applicability of our method as the number of the layout objects increase. In addition, we think that a trajectory should represent a class of objects. For example, the circle represents the flower. Therefore, we need to explore how to design a trajectory to recognize different objects under the same class. To solve this problem, two ways were designed for users to choose. One was to generate a menu of this class of objects in the center of the trajectory after recognizing trajectory, and the user selected again. The other was to distinguish different objects by numbers (see **Figure 5**).

In this experiment, each participant was asked to perform these three tasks in turn (see **Figure 6**). Although the focus of each task was different, we all recorded the time participants took to complete each task. It is helpful for us to better analyze the effectiveness of our method. Combining objective data such as task completion time with subjective user perceptions, we can gain some insights into the usability and intuitiveness of two methods.

Figure 5. Two forms of trajectory recognition of a class of objects: menu selection (left) and distinguish by numbers (right).

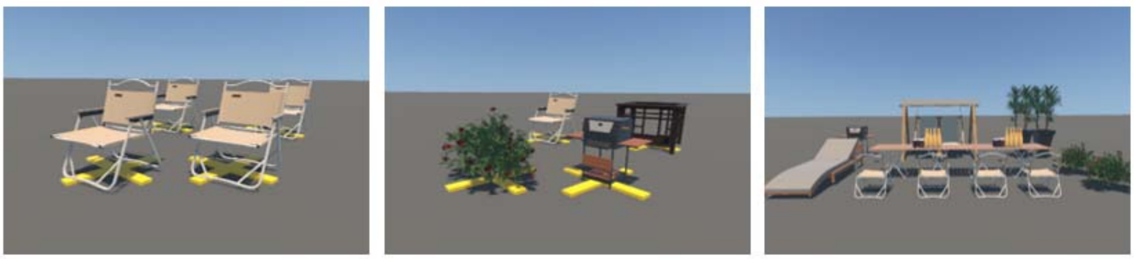

Figure 6. Task 1(left), Task 2 (middle) and Task 3 (right).

4.2. Apparatus and participants

This study was conducted in our laboratory with HTC Vive HMDs and handheld controllers, which tracked with two tripod-mounted Lighthouse 1.0 base stations. The HMD was connected to a computer (3.20 GHz AMD Ryzen 7 5800H with NVIDIA GeForce RTX 3060 GPU) and the experimental platform was developed in Unity2020.4.37.

The experiment counted with 8 participants, 2 women and 6 men. Half of the participants had previous VR experience, while the other half did not. All participants were right-handed with normal or corrected-to-normal vision. Each participant arrived at the experiment site at a predetermined time.

4.3. Procedure

Each participants went through the same experiment process, which was divided into three phases (see **Figure 7**). The whole process lasted an average of 45 minutes.

Phase 1: preparation (10 min.). A member of our team explained to the participants the purpose and nature of the experiment and how their information would be collected and processed. Each participant filled out a pre-study questionnaire about previous VR experience. For those new to VR, given 5 minutes to get used to the VR headset and learn about the button operation of the controller. Once the participants became accustomed and comfortable with their virtual presence, we proceeded to the next phase.

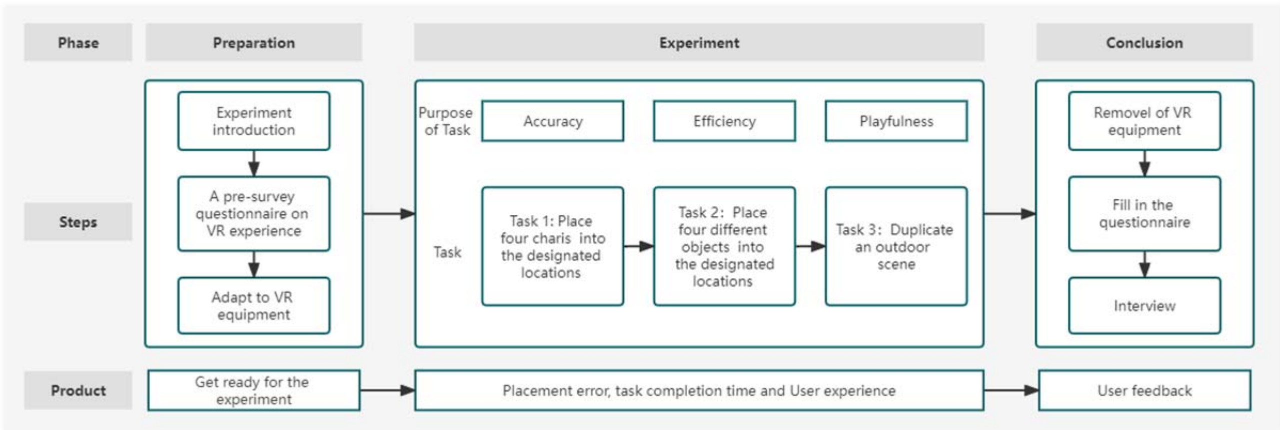

Figure 7. The experiment procedure was followed by each participant.

Table 1. Subjective questionnaire of object placement methods

Q1: "I will use the method of track recognition to place objects in VR."
Q2: "I think the method of trajectory recognition improves the layout efficiency."
Q3: "I think the method of trajectory recognition can place the object accurately on the target point."
Q4: "I think the method of trajectory recognition is more interesting."
Q5: "The act of opening the menu takes me out of the experience process."
Q6: "I think it is another way to move objects by deleting objects in the original position and generating objects in the new position."
Q7: "I think the function of user-defined trajectory adds flexibility and playfulness."
Q8: "For the selection of a class of objects, I prefer the way of drawing trajectory and number."
Q9: "I think the method of trajectory recognition provides a reference for VR layout interaction."

Phase 2: experiment (20 min.). We introduced the task content to the participants through playing teaching video and oral explanation. They were given 5 minutes to explore the method freely. The experiment officially began after the exploration. The participants performed the tasks in sequence. The data obtained with the two object placement methods were collected separately, including execution time and placement point error. We also closely observed how participants responded during the tasks, as this tends to be the most authentic user feedback.

Phase 3: conclusion (15 min.). Team members helped participants remove the VR equipment and instructed them to answer a 7-point Likert scale subjective questionnaire related to our placement methods (shown in Table 1). We also conducted a more open and subjective interview where participants could express their opinions and aspects of our method that needed improvement. This was of great significance for our future work.

5. Results

In this section, we analyzed the data obtained in three experiment tasks and the statistical results from the subjective questionnaire. To discuss the comprehensive performance of our method, a user experience evaluation system was also constructed in Section 5.3.

5.1. Time and precision

To examine the placement efficiency of the two methods, we recorded the duration of each participant completing the three tasks. The results are shown in **Figure 8**. Since there is no need to change layout objects in Task 1, the completion time of the two methods is similar. The advantages of our method begin to show when change objects frequently in Task 2. Task 3 was designed to simulate the intended use cases for our method. We recorded the total time for duplicating the scene. As stated in our first hypothesis **H2**, our method is superior to the direct placement method in terms of placement efficiency. This advantage will continue to expand with the increase of the number of layout objects.

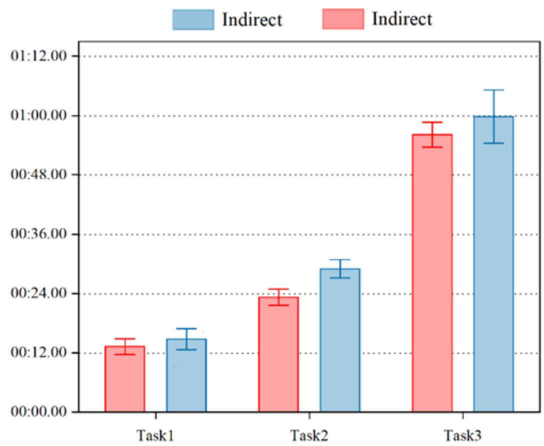

Figure 8. Three tasks completion time. The error bar represents the standard error of the average.

In terms of placement accuracy, we analyzed the data of Task 1 and calculated the placement error of each participant. The error is represented by the distance between the actual placement point and the target point. A large distance means a large error. The high accuracy of the direct method is determined by its own principle. It can be seen from the **Table 2** that our method of taking the trajectory center as the placement point also has similar high accuracy. Since our main goal is to improve placement efficiency and playfulness, we accept more errors than the indirect method. The experimental results are consistent with the statement in hypothesis **H1**.

Table 2. Mean and standard deviation of distance between the actual object placement point and the target

	Target point1(far)	Target point2(far)	Target point3(near)	Target point4(near)
Indirect	1.84 (0.2478)	1.97 (0.53)	1.50 (0.26)	1.43 (0.29)
Direct	1.32 (0.5112)	1.32 (0.43)	1.02 (0.23)	0.98 (0.23)

5.2. User feedback

Participants were asked to score subjective questionnaire with 7-point Likert scale items. The 7-point Likert scale items are ranged from very strongly disagree (1) to very strongly agree (7). The results are shown in **Figure 9**.

The purpose of this questionnaire is to explore the usability of the method and the user's subjective perception of the layout interaction. Most of participants thought our method is easy to learn and intuitive to use (Q1). One participant stated, "I can use this method to place objects easily and quickly." Two participants with no previous VR experience stated that they often forgot the menu button when placing objects with the direct method.

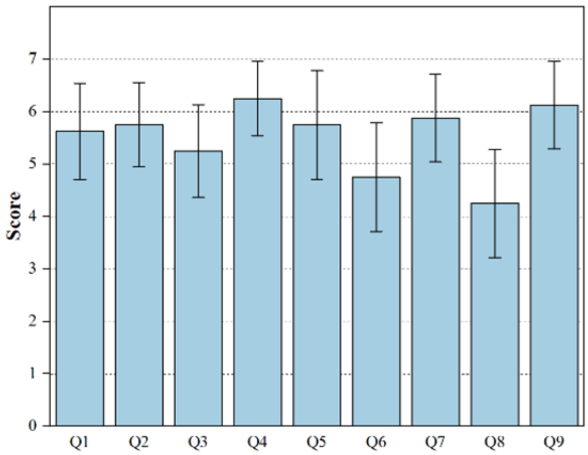

Figure 9. Subjective questionnaire score. The error bar represents the standard error of the average.

The feedback we got on placement efficiency, placement accuracy, and playfulness (Q2, Q3, Q4) was positive. Most of participants stated that our method has obvious advantages in improving placement efficiency when there are many layout objects and objects need to be changed frequently. The playfulness got the highest score in user evaluation, and the participants all stated our method is interesting. The score of subjective question Q4 can support our hypothesis **H3**.

Regarding the interaction with the menu (Q5), many participants expressed a desire to focus on the spatial layout of objects and reduce other interactions when creating a virtual scene. However, the act of opening a menu interrupts the creation process and takes oneself out of creation. In addition, two other participants stated that opening the menu frequently would hinder the user's perspective. Some participants favored Q6, but stated that this way of moving objects would reduce the object's connection to the environment.

The function of user-defined trajectories (Q7) was well received among participants. The system designs four initial trajectories and allows participants to modify according to their preferences. One participant stated: "Creating my own trajectory and combining painting and space design in a virtual environment makes me become a painter and a space designer". The score of Q8 shows that distinguishing a class of objects with numbers increases the trajectory drawing time. The trajectory should not be too complicated. In general, the participants liked our method and thought that the concept of trajectory recognition was also inspirational for the design of other interactive functions (Q9).

In future work, the user thought need to improve or the lack of features were: (1) Add the function of scaling and rotating through trajectory identification. The user can zoom and rotate after the object is placed at the target position. It is possible to define the size of rotation and scaling based on the size of the trajectory. (2) Expand the application scope of this method. It is our next work to extend this method to free-hand interaction and study the combination of multimodal interaction technology based on gesture and speech and this method.

5.3. Construction of user experience evaluation system

Although the results of the user study in Sections 5.1 and 5.2 confirmed the good performance of individual aspects, the overall performance of our method has not been evaluated. For a comprehensive and accurate evaluation, we explored the degree of unilateral influence on the overall evaluation. Combining Analytic Hierarchy Process (AHP)[33] with fuzzy comprehensive evaluation, we constructed a user experience evaluation system. The weight of evaluation index was determined by AHP (Section 5.3.1) and the final evaluation result of the method was obtained by fuzzy comprehensive evaluation (Section 5.3.2). Based on our research, we took placement efficiency, playfulness and placement accuracy as three evaluation indexes.

Determination of evaluation indexes

We invited two experts in the field of virtual reality and four people with rich VR experience to evaluate the importance of three indexes based on the absolute judgment scale. The steps for analyzing evaluation data are as follows: (1) Calculate the third-order comparison matrix by comparing the importance of relevant elements in pairs. (2) Calculate the maximum eigenvalue and weight vector. (3) Calculate consistency index (CI). (4) Query the random consistency index table, RI = 0.52 is obtained by n = 3. (5) Calculate CR. CR <0.1 means that the judgment matrix satisfies the consistency test.

The weights of three indexes of six experts were obtained through the above steps (see **Table 3**). The data did not pass the consistency test were not included in the calculation (Expert 4). We took the mean weight of each index and got the weight vector A. As can be seen from **Table 3**, efficiency has the greatest influence while precision has the least.

$$A = [0.4238 \quad 0.3836 \quad 0.1925] \tag{1}$$

Table 3. Weight results of three indexes

Index	Expert1	Expert2	Expert 3	Expert 4	Expert 5	Expert 6	Wight
Efficiency	52.468%	63.335%	58.126%	60.700%	19.762%	18.223%	42.38%
Playfulness	33.377%	26.050%	30.915%	30.334%	31.190%	70.284%	38.36%
Accuracy	14.156%	10.616%	10.959%	8.965%	49.048	11.493%	19.25%
CR	0.052	0.037	0.004	0.132	0.052	0.052	

Fuzzy comprehensive evaluation

We constructed a comment set V={v1, v2, v3, v4}, where v1 = “very satisfied”, v2 = “relatively satisfied”, v3 = “not very satisfied”, v4 = “dissatisfied”, corresponding to the evaluation score of 4, 3, 2, 1 respectively. 8 participants were required to evaluate each index. We counted the number of evaluators in four class of the three indexes, as shown in the **Table 4**.

Table 4. The number of evaluators in four class of the three indexes

Index	Quite satisfied	Satisfied	A little unsatisfied	Unsatisfied
Efficiency	3	4	1	0
Playfulness	4	3	1	0
Accuracy	2	3	2	1

According to the statistical results, we calculated the comprehensive score with 4 specific steps. (1) Calculate the single-factor evaluation ector for each index by the proportion of the four evaluations. (2) Calculate the evaluation matrix B by combining the single factor evaluation vector of three indexes. (3) Calculate and normalize the membership matrix C: C = A*B. (4) Finally, Calculate the comprehensive score according to the linear relationship (see **Table 5**).

$$B = \begin{bmatrix} 3/8 & 1/2 & 1/8 & 0 \\ 1/2 & 3/8 & 1/8 & 0 \\ 1/4 & 3/8 & 1/4 & 1/8 \end{bmatrix} \tag{2}$$

$$C = [0.40 \quad 0.43 \quad 0.15 \quad 0.02] \tag{3}$$

The score of fuzzy comprehensive evaluation was 3.21, which between 3 and 4. According to the maximum membership principle, it could be concluded that participants were satisfied with the overall evaluation of our method.

Table 5. The comprehensive score

Evaluation	Coefficient	Score
Quite satisfied	0.40	4
Satisfied	0.43	3
A little unsatisfied	0.15	2
Unsatisfied	0.02	1
Result: 0.40*4 + 0.43*3 + 0.15*2 + 0.02*1		3.21

6. Conclusions

We propose an object placement method based on trajectory recognition of handheld controller. Our method extends 2D gesture recognition to VR and allows users to customize the trajectory. The novelty and flexibility bring users a good interactive experience. In order to evaluate the performance of our interactive method, we conducted a comparative user study. Three experimental tasks were designed to study the accuracy, efficiency and playfulness of object placement respectively. The results showed that our method is faster and more interesting than the direct object placement method of. To further discuss the comprehensive performance of our method, a user experience evaluation system was constructed by combining of Analytic Hierarchy Process with fuzzy comprehensive evaluation. The high comprehensive evaluation score indicated that users were satisfied with our object placement method. We also gathered a lot of user feedback, which will provide important reference value for our future work.

Conflict of interest

The authors declare no conflict of interest.

References

1. Zheng C. Key technologies of the metaverse and their similarities and differences with digital twin. Network Security Technology and Application 2022; (09): 124–126.
2. Lang B. Analysis: Monthly-connected VR headsets on steam blast through 3 million milestone [Internet]. 2021. Available from: https://www.roadtovr.com/monthly-connected-vr-headsets-steam-survey-january-2022/
3. Zhang L, Pan H. Research on VR reading user interaction behavior under empowerment. Library Forum 2022; 1–12.
4. Peña Arcila JB. Master of Ibero American virtual environment education. Metaverse 2020; 1(1): 11.
5. Fang L, Shen H. Conceptualizing metaverse: A perspective from technology and civilization. Metaverse 2021; 2(2): 18.
6. Shin DH. User experience in social commerce: In friends we trust. Behaviour & Information Technology 2013; 32(1): 52–67.
7. Park M, Yoo J. Effects of perceived interactivity of augmented reality on consumer responses: A mental imagery perspective. Journal of Retailing and Consumer Services 2020; 52:101–912.
8. Caputo FM, Prebianca P, Carcangiu A, et al. Comparing 3d trajectories for simple mid-air gesture recognition. Computers & Graphics 2018; 73: 17–25.
9. Li Y (editor). Protractor: A fast and accurate gesture recognizer. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems; 2010 Apr 10-15; Atlanta Georgia. New York: ACM Press; 2010. p. 2169–2172.
10. Wobbrock JO, Wilson AD, Li Y (editors). Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology; 2007 Oct 7-10; Newport. New York: Association for Computing Machinery; 2007. p. 159–168.
11. Anthony L, Wobbrock JO. A lightweight multistroke recognizer for user interface prototypes. Proceedings of Graphics Interface; 2010 May 31- June 2; Ottawa. Toronto: Canadian Information Processing Society; 2010. p. 245–252.
12. Anthony L, Wobbrock JO. $ N-protractor: A fast and accurate multistroke recognizer. Proceedings of Graphics Interface; 2012 May 28-30; Toronto. Toronto: Canadian Information Processing Society; 2012. p. 117–120.
13. Vatavu RD, Anthony L, Wobbrock JO (editors). Gestures as point clouds: A $ p recognizer for user interface prototypes. Proceedings of the 14th ACM International Conference on Multimodal Interaction; 2012 Oct 22-26; Santa Monica. New York: Association for Computing Machinery; 2012. p. 273–280.
14. Vatavu RD (editor). Improving gesture recognition accuracy on touch screens for users with low vision.

Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems; 2017 May 6-11; Denver. New York: Association for Computing Machinery; 2017. p. 4667–4679.

15. Vatavu RD, Anthony L, Wobbrock JO (editors). $ q: A super-quick, articulation-invariant stroke-gesture recognizer for low-resource devices. Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services; 2018 Sep 3-6; Barcelona. New York: Association for Computing Machinery; 2018. p. 1–12.
16. Kratz S, Rohs M. A $3 gesture recognizer: Simple gesture recognition for devices equipped with 3d acceleration sensors. Proceedings of the 15th International Conference on Intelligent User Interfaces; 2010 Feb 7-10; Hong Kong. New York: Association for Computing Machinery; 2010. p. 341–344.
17. Wobbrock JO, Wilson AD, Li Y. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. Proceedings of the 20th annual ACM symposium on User interface software and technology; 2007 Oct 7-10; Newport. New York: Association for Computing Machinery; 2007. p. 159–168.
18. Caputo FM, Prebianca P, Carcangiu A, et al. A 3 cent recognizer: Simple and effective retrieval and classification of mid-air gestures from single 3d traces. In: Giachetti A, Pingi P, Stanco F (editors). STAG 2017 - Smart Tools and Applications in Graphics; 2017 Sep 11-12; Catania. Reims: Eurographics Association; 2017. p. 9–15.
19. Kratz S, Rohs M (editors). Protractor 3d: A closed-form solution to rotation invariant 3d gestures. Proceedings of the 16th International Conference on Intelligent User Interfaces; 2011 Feb 13-16; Palo Alto. New York: Association for Computing Machinery; 2011. p. 371–374.
20. Liu J, Zhong L, Wickramasuriya J, et al. uWave: Accelerometer-based personalized gesture recognition and its applications. Pervasive and Mobile Computing 2009; 5(6): 657–675.
21. Ousmer M, Sluyters A, Magrofuoco N, et al. (editors). Recognizing 3d trajectories as 2d multi-stroke gestures. Proceedings of the ACM on Human-Computer Interaction 2020; 4(ISS): 1–21.
22. Eroglu S, Stefan F, Chevalier A, et al. (editors). Design and evaluation of a free-hand VR-based authoring environment for automated vehicle testing. 2021 IEEE Virtual Reality and 3D User Interfaces (VR); 2021 Mar 27-Apr 1; Lisboa. New York: IEEE; 2021. p. 1–10.
23. Mine M. Isaac: A virtual environment tool for the interactive construction of virtual worlds [Internet]. 1995 [updated 1995 May 5]. Available from: https://www.semanticscholar.org/paper/ISAAC%3A-A-Virtual-Environment-Tool-for-the-of-Worlds-Mine/e0e2b7972b6504d17c60d33ad87f856fda376b66
24. Burns A, Sugden B, Massey L, et al. Gaze-based object placement within a virtual reality environment. U.S. Patent. 10,416,760. 2019 Sept 17.
25. Mapes DP, Moshell JM. A two-handed interface for object manipulation in virtual environments. Presence Teleoperators & Virtual Environments 1995; 4(4): 403–416.
26. Kan P, Kaufmann H (editors). Automatic furniture arrangement using greedy cost minimization. 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR); 2018 Mar 18-22; Reutlingen. New York: IEEE; 2018. p. 491–498.
27. Gal R, Shapira L, Ofek E, et al. (editors). Flare: Fast layout for augmented reality applications. 2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR); 2014 Sept 10-12; Munich. New York: IEEE; 2014. p. 207–212.
28. Germer T, Schwarz M. Procedural arrangement of furniture for real-time walkthroughs. Computer Graphics Forum 2009; 28: 2068–2078.
29. Merrell P, Schkufza E, Li Z, et al. Interactive furniture layout using interior design guidelines. ACM Transactions on Graphics (TOG) 2011; 30(4): 1–10.
30. Dou H, Tanaka J. A mixed-reality shop system using spatial recognition to provide responsive store layout. International Conference on Human-Computer Interaction; 2020 Jul 19-24; Copenhagen. New York: Springer, Cham; 2020. p. 18–36.
31. Ferreira J, Mendes D, Nobrega R, et al. (editors). Immersive multimodal and procedurally-assisted creation of VR environments. 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW); 2021 Mar 31-Apr 1; Lisbon. New York: IEEE; 2021. p. 30–37.
32. Barot C, Carpentier K, Collet M, et al. (editors). The wonderland builder: Using storytelling to guide dream-like interaction. 2013 IEEE Symposium on 3D User Interfaces (3DUI); 2013 Mar 16-17; Orlando. New York: IEEE; 2013. p. 201–202.
33. Saaty TL. What is the analytic hierarchy process? In: Mitra G, Greenberg HJ, Lootsma FA, et al. (editors). Mathematical models for decision support. Berlin, Heidelberg: Springer; 1988. p. 109–121.