2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) | 978-1-6654-0158-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISMAR52148.2021.00020

VR Collaborative Object Manipulation Based on Viewpoint Quality

Lili Wang[1;2;3] * Xiaolong Liu[1] Xiangyu Li[1]

[1]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China
[2]Peng Cheng Laboratory, Shengzhen, China
[3]Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University, Beijing, China.

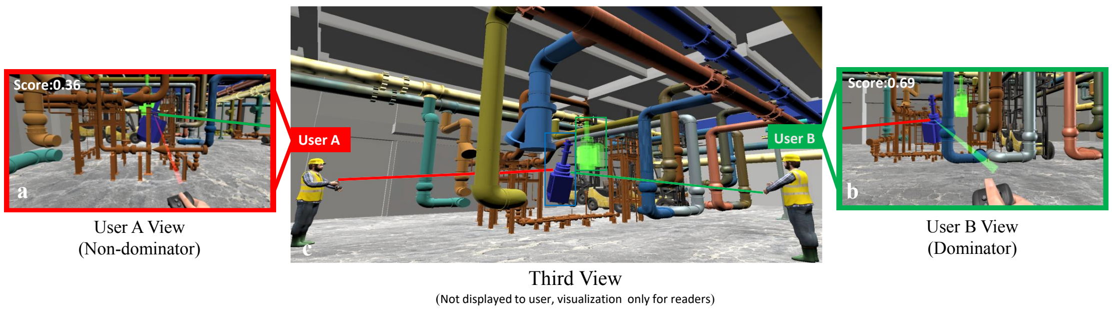

Figure 1: Collaborative manipulation based on view quality in scenarios where the pipeline is manipulated to match the target. Two users are manipulating the pipe collaboratively. (a) and (b) are the views of users A and B, respectively, (c) is the third view used to visualize the positions of users A and B, the blue box indicates the blue pipe being manipulated, and the green box marks the pipe target location. Using our method, the viewpoint quality of user B is higher than that of user A (0.69 vs 0.36), that is, user B is more suitable for manipulating objects in this frame, so user B is set as the dominant manipulator.

Abstract

We introduce a collaborative manipulation method to improve the efficiency and accuracy of object manipulation in virtual reality applications with multiple users. When multiple users manipulate an object in collaboration, a certain user may have a better perspective than other users at a certain moment, and can clearly observe the object to be manipulated and the target position, and it is more efficient and accurate for him to manipulate the object. We construct a viewpoint quality function and evaluate the viewpoints of multiple users by calculating its three components: the visibility of the object need to be manipulated, the visibility of target, the depth and distance combined of the target. By comparing the viewpoint quality of multiple users, the user with the highest viewpoint quality is determined as the dominant manipulator, who can manipulate the object at the moment. A temporal filter is proposed to filter the dominant sequence generated by the previous frames and the current frame, which reduces the dominant manipulator jumping back and forth between multiple users in a short time slice, making the determination of the dominant manipulator more stable. We have designed a user study and tested our method with three multi-user collaborative manipulation tasks. Compared to the previous methods, our method showed significant improvement in task completion time, rotation accuracy, user participation and task load.

Index Terms: Virtual reality—Collaboration—Object manipulation—Viewpoint quality;

1 Introduction

Object manipulation (translation, rotation, and scaling) is one of the most commonly used basic operations in 3D user interaction and can be used in many virtual reality (VR) applications, such as product design, 3D object modeling, and virtual object assembly. The efficiency and accuracy of the manipulation directly affect the effect of the applications. Many techniques have been proposed to improve the accuracy and efficiency of object manipulation, such as mid-air manipulation [15] and PinNPivot manipulation [8].

Collaborative manipulation means that multiple users share a virtual environment and perform collaborative manipulation on the same object, which is necessary for applications such as VR team manipulation training. The advantage of collaborative manipulation is that if a single user cannot efficiently and accurately complete a specific object manipulation task in a virtual environment, usually other users can help him efficiently and accurately complete the manipulation, that is, collaborative manipulation can enhance the team's ability to solve problems. In complex manipulation tasks (such as precise positioning of objects), multi-user collaborative manipulation is superior to single-user manipulation in terms of efficiency and accuracy [9]. For example, when the surrounding environment occludes the target location, when the user uses ray casting technology to place the object far away from its current location, a second user with better visibility can help him efficiently and accurately manipulate the object.

The difficulty of collaborative manipulation is how to collaborate among the multiple users. The intuitive methods of collaborative manipulation are first-come-first-manipulate and active switching dominance. First-come-first-manipulate means that when multiple users manipulate the same object in a given frame, the first user detected to be manipulated has the right to manipulate the object, while other users' manipulations are invalid. Active switching dominance means that non-dominant user can actively switch dominance when he think his view is better to the view of the dominance manipulator. These two methods are simple and easy to implement, but they do not consider the quality of the viewpoint of all users, so the efficiency and accuracy still need to be improved. Lages et al. [14] proposed a method of manually specifying the dominant manipulator in the collaborative manipulation process. This method assigns a user to the role of the director, who can observe the entire

*Corresponding Author: wanglily@buaa.edu.cn

978-1-6654-0158-6/21/$31.00 ©2021 IEEE
DOI 10.1109/ISMAR52148.2021.00020

scene and the locations of all collaborative users, and then manually specify who is the dominant manipulator in the current frame. The problem with this method is that the director's experience directly affects the assignment of the dominant manipulator.

In this paper, we introduce a collaborative manipulation method based on viewpoint quality evaluation to improve efficiency and accuracy of object manipulation in VR applications. We construct a viewpoint quality function with three components: the visibility of the object need to be manipulated, the visibility of target and the comprehensive item of the target depth and distance to evaluate the viewpoints of multiple users. Then, we compare the viewpoint quality of multiple users, and select the user with the highest viewpoint quality as the dominant manipulator, who directly manipulates the object at the moment. In order to maintain the stability of dominance, we introduce a temporal filter to filter the dominant sequence generated for the previous frames and the current frame.

We designed a user study and tested our method through three multi-user collaborative manipulation tasks. The results show that, compared with the existing methods for determining the dominant manipulator, our method has a significant improvement in the completion time of the manipulating task and the rotation accuracy. Moreover, our method can balance user participation time and significantly reduce task load without having to reduce presence or increase simulator sickness as the cost. Fig. 1 illustrates our method in the scenario of a pipeline manipulation task (T3).

2 Related Work

Many researchers devoted themselves to study of object manipulation for multi-user collaborative tasks. Ruddle et al. [20] compared two integrated action methods in collaborative manipulation: one used the synchronized component of participants' manipulations, and the other computed the mean manipulation of participants' manipulations. Duval et al. [4] only used the user's translation to manipulate the object, instead of the complex 6-degree-of-freedom motion. Riege et al. [19] proposed a Bent-Pick-Ray method, which is a ray-based collaborative manipulation method in VR. When two users manipulate the same object, the selected ray bends according to the pointing direction and the selected point, which provides continuous visual user feedback.

In the above methods, the roles of multiple users in collaborative manipulations are symmetrical, i.e. the manipulations that these users can perform on objects are the same. The difference between these methods is that the manipulations of multiple users are integrated in different ways to manipulate on the objects.

There are also some collaborative manipulation methods, where the roles of multiple users are different. Pinho et al. [16] proposedb a multi-user collaborative object manipulation method that assigned different manipulating freedoms to different users. For example, during the entire manipulation, user A is only responsible for translating the object, and user B is only responsible for rotating the object. Soares [23] proposed the EGO-EXO technology, which also separated the degree of freedom of object manipulation, and assigned different users with different manipulations according to the distance between the user's initial position in the virtual scene and the manipulated object. Chenechal [2] proposed an asymmetric collaborative manipulation method in a multi-scale environment. In this method, the first user is a giant with a global field of view to translate the object, the second user is an ant inside the manipulated object to scale and rotate object, the third one can help the second user to scale the object using a third view, and the fourth one can switch other's viewpoints to help them to communicate. Grandi [9] described the asymmetric collaborative manipulation in virtual and augmented realities. Two users collaboratively manipulate object, one in VR and one in AR. They also compared this asymmetric collaborative operation with symmetrical collaborative operation, i.e. both users are in VR or both users are in AR. Wang et al. [27] investigated gaze behavior in a face-to-face collaborative assembly task. One user is responsible for the manipulation. His gaze was detected by Tobii Pro Glasses 2 and shared to the other user acting as an assistant to hand over the required parts.

Some research focused on how to determine that one or more users can manipulate objects in the current frame among multiple users. Baron [1] used two conditions to determine the user's right to manipulate the object: 1) at least one other user's permission is obtained before the user can select a handle of the object; 2) after the user manipulates the object, he must obtain the consent of all other users for the manipulation to be effective. Lages [14] proposed a ray-camera action method, in which the director can assign manipulation rights to actors according to the views of the director.

Similar to these methods, our method also need to determine the dominant manipulator in the process of the object manipulation. However, our method is based on the evaluation of viewpoint quality for multiple users and determines the dominant manipulator with the highest viewpoint quality in real-time automatically.

Some research focused on the evaluation of viewpoint quality. Generally, a good viewpoint means that the scene image rendered from this viewpoint has less occlusion, a larger visible area of the object of interest to the user, and prominent visual features of the object. Kamada et al. [12] proposed the new concept named as a general position of the viewpoint. From this viewpoint, the maximum shape information of the objects can be obtained in the image rendered. Plemenos et al. [17] introduced a new constrain of the general position of the viewpoint, which was if the maximum angular deviation between the angle of view direction and the surface normal of the object was the smallest, it showed a lot of geometric details on the object, so the viewpoint was good. Vázquez et al. [26] proposed viewpoint entropy, which included the projection area and the number of visible faces of the objects, and they this measurement to explore objects or scenes automatically. Sokolov et al. [24] proposed viewpoint quality in the context of global world exploration, which included two factors: the total curvature for meshes and the projection area of the visible region of the objects. After this, they [25] proposed another method to calculate viewpoint quality for automatic 3D scene exploration, in which three factors were used: the size of object bounding box, the observation quality and the fraction of visible area of the object. Freitag et al. [5] proposed a method to normalize the viewpoint quality values according to the viewpoint quality of the whole scene and used the normalized viewpoint quality to adjust the travel speed when traversing large scenes automatically. Then they [6] proposed an interactive assist interface based on automatic analysis of object visibility and viewpoint quality to support exploration, and guide the user to the interesting parts of the scene. The viewpoint quality included the object uniqueness and the visual size of the object.

Most of the existing methods use viewpoint quality to improve the efficiency of a single user's navigation and exploration in a virtual scene. Our method is to calculate and compare the viewpoint quality of multiple users to select the dominant manipulator to improve the efficiency and accuracy of the manipulation, So the viewpoint quality evaluation function in our method is completely new.

3 Method

Our method determines the dominant manipulator for VR collaborative manipulation in real time by evaluating and comparing the quality of multiple users' viewpoints. Assume that there are two users A and B working together in the virtual scene to manipulate the object to the target position. In the first frame, we render images under viewpoints A and B respectively, calculate the quality of the viewpoint, and set the user with a higher viewpoint quality as the dominant manipulator. In the next frame, we set the dominant manipulator based on the results of the two user viewpoint quality calculations in the current frame and the previous frames. For exam-

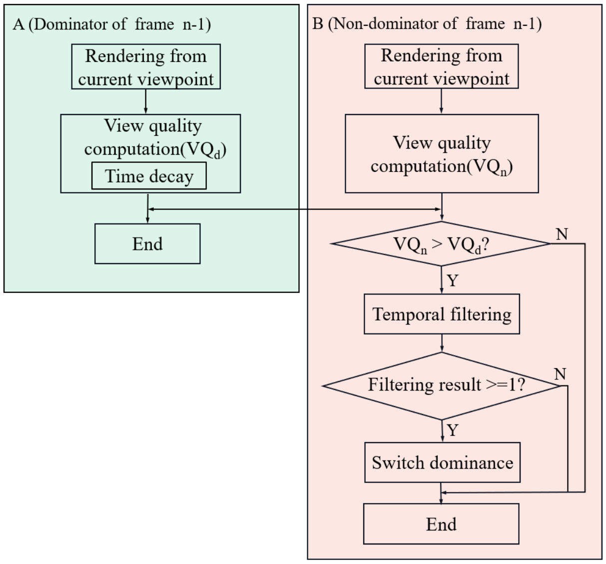

Figure 2: Manipulation dominance determination pipleline based on viewpoint quality of the current frame *n* for both the dominator and the non-dominator of the previous frame *n-1*.

ple, if A is the dominator and B is the non-dominator of frame *n-1*, Fig. 2 shows the pipeline of our method for setting the dominator for A and B of frame *n*.

For dominator A, we first render the view from A's viewpoint, calculate the viewpoint quality of A with the viewpoint quality function, and send the quality score to the non-dominator B. The first two steps of B are the same as those of A. in the third step, we compare B's viewpoint quality with A's viewpoint quality received. If the quality of B's viewpoint is less than that of A, we still keep A as the dominator and B as the non-dominator; otherwise, the dominance sequence generated with considering of the previous frames and the current frame for B is filtered with the temporal filter (Sect. 3.2). If the result of the filtering is '1', we switch the dominance in frame *n*, i.e. A is set as the non-dominantor, and B is set as the dominantor; otherwise we keep A as the dominantor and keep B as the non-dominantor.

This pipeline can easily be extended to the collaborative manipulation for more users. The only difference between the multi-user pipeline of our method and the two user pipeline is that the viewpoint quality scores of all users need to be sent to each non-dominant user of frame *n-1*, and every non-dominant user must judge whether its own viewpoint quality is the highest. According to the judgment result, it enters the same follow-up process as the two-user pipeline, i.e. if the viewpoint quality of an non-dominant user is largest, temporal filtering is conducted, then we set this user as the dominant manipulator if the filtering result is '1'; otherwise, we don't change the dominance. Only the pipeline of the non-dominant user with the highest viewpoint quality will enter the filtering process. When its filtering result is 0, the original dominance is maintained. We did not continue to test other non-dominant users with lower quality viewpoints. This is because according to our method, after a certain period of time, this non-dominant user with the highest viewpoint quality will pass filtering and become a dominant manipulator.

3.1 Viewpoint quality computation

We construct the viewpoint quality evaluation function with 3 factors. The first one is the visibility of the object need to be manipulated, the second one is the visibility of the target, and the third is a comprehensive metric of the depth and distance of the target from the current viewpoint. These three factors are weighted and summed with the weights α and β (0.2 and 0.4 in our implementation). The initial values of α and β are set as 0.33. We use some images that are very easy to subjectively distinguish quality to test these weights, and adjust the weights to make the results as reasonable as possible.

User viewpoint quality is quantified with this evaluation function in Algorithm 1. The inputs of the algorithm are the object need to be manipulated o, the target t, the user viewpoint of the current frame V and the image I rendered from V in the first step. Image I has 3 channels: $I.id1$ stores the object ID of the first depth layer, $I.id2$ stores the object ID of the second depth layer, and $I.z$ stores the z values in 3D world coordinate system of the first depth layer. The output of the algorithm is the user viewpoint quality q.

Algorithm 1 Viewpoint quality evaluation

Input: object o, target t, viewpoint V, view I
Output: Viewpoint quality q of V
1: $A_o = \text{Area}(I, o)$;
2: $A_t = \text{Area}(I, t)$;
3: $A_{ot} = \text{Area}(I, o, t)$;
4: $V' = \text{ConstructLargeFov}(V, \text{LARGEANG})$;
5: $I_o = \text{RenderObject}(V, V', o)$;
6: $I_t = \text{RenderObject}(V, V', t)$;
7: $A'_o = \text{Area}(I_o, o)$;
8: $A'_t = \text{Area}(I_t, t)$;
9: $dq = \text{DepthQuality}(I, t)$;
10: $d = \text{Distance}(V, t)$;
11: $A''_t = A'_t - A_t - A_{ot}$;
12: $q_1 = A_o / A'_o$;
13: $q_2 = (A_t / A'_t) e^{-A''_t / A'_t}$;
14: $q_3 = dq / \sqrt{d^2 + 1}$;
15: $q = \alpha q_1 + \beta q_2 + (1 - \alpha - \beta) q_3$;

Algorithm 2 Depth quality computation

Input: view I, target t
Output: Depth quality dq
1: $b = \text{AABB}(t)$;
2: $l = \text{DiagonalLength}(b)$;
3: $z = \text{ZBuffer}(I, t)$;
4: $var = \text{Variance}(z)$;
5: $dq = 4 * var / l^2$;

In this algorithm, first, we calculate the visible area of the object to be manipulated, the visible area of the target, and the area of t occluded by o (lines 1-3). Then we construct a new viewpoint V', which has the same position and orientation of V, but has a large FOV angle, e.g. 170 degree in horizontal direction (line 4). After that, we use V' to render image I_o with a single object o and image I_t with t, calculate the ratio of the FOV angles of V and V', and normalize the image size of I_o and I_t (lines 5-6). The area A'_o of o in I_o and the area A'_t of t in I_t are computed (lines 7-8). The area A'_o and A'_t are larger than the area A_o and A_t due to two reasons. The first one is that we only render a single object o or t with V', so there is no occlusion in the image; the second one is that the FOV angles of V is large, so in the most cases o and t are entirely in the field of view. Next, we compute depth quality of t in I with the function $DepthQuality$ (line 9), which is described in Algorithm 2. The distance from V to the center of t is calculated in line 10. A''_t in line 11 represents the invisible area of t from the viewpoint V and not occluded by o. Then, the three factors q_1, q_2 and q_3 are calculated with the equations in lines 12-14 and weighted summed in line 15 to get the final viewpoint quality q.

Algorithm 2 takes the image I and target t as inputs, and outputs the depth quality dq of t. We construct the AABB b of t (line 1), and obtain the diagonal length of b (line 2). Then we get z values of t's footprint in I (line 3), compute the variance of these z values (line 4), and calculate dq in line 5. The greater dq, the greater the change in z value of the visible area of the target t.

For the dominant manipulator, we add an extra time decay factor to its viewpoint quality function. For a period of time, if user A always dominates the object manipulation, and the viewpoint quality of user A and user B are very similar, we add a time decay term to reduce A's viewpoint quality, and give B the dominance. This is because in the preliminary experiment, we found when the object approached the target, and only minor adjustments were needed. At that time, if one user manipulated the object for a long time, the matching error cannot be effectively decreased. Therefore, it is better to switch the dominant position between the two users and perform alternate manipulations on the objects. The time decay factor q_{fd} is computed in Equation 1, where q_d is the viewpoint quality for the dominator and q_{nd} is the viewpoint quality for the non-dominator.

$$q_{fd} = -2 * |q_d - q_{nd}| \tag{1}$$

3.2 Temporal Filtering

In order to avoid the high-frequency switching of dominance among multiple users from affecting the continuity of the manipulation, we propose a temporal filtering algorithm to filter the view quality comparison sequence of the user with the highest viewpoint quality in the current frame, and determine whether to switch the dominance according to the filtering result. For each user, if it has the highest viewpoint quality in a frame, we set its flag in this frame to 1, otherwise the flag is set to 0. After the non-dominant user with the highest viewpoint quality is determined in frame n, we get its flags of frame n-i+1 to n and generate a view quality comparison sequence. In our implementation, i is 100.

Two filters are applied to the view quality comparison sequence. The first filter is $[1/m, 1/m, ..., 1/m]$, whose length is i. If the filtering result is greater than 1, it means that this user has the highest viewpoint quality in at least m frames in consecutive i frames, and we set it to the dominant manipulator. m is set as 60 in our implementation. Otherwise, we relax the filtering with the second filter $[0, 0, ...0, 1/k, 1/k, ..., 1/k]$. The number of '0' in this filter is $i-k$, the number of '1/k' is k, and $k < m$ ($k = 30$ in our implementation). If the result of the second filter is 1, it means this user starts to have a best viewpoint quality for k frames, so we can set him as the dominant manipulator.

4 User Study

4.1 Pilot User Study

We first designed a pilot user study to evaluate the effectiveness of our viewpoint quality evaluation function.

Participants. We have recruited 12 participants, 9 males and 3 females, between 20 and 30 years old. Ten of our participants had used HMD VR applications before. Participants had normal and corrected vision, and none reported vision or balance disorders.

Hardware and software setup. We used two sets of HTC Cosmos VR systems with two hand-held controllers, allowing two users to point virtual lasers at the virtual environment (VE). Each HMD was connected to its own workstation with a 3.6GHz Intel(R) Core(TM) i7-9900KF CPU, 16GB of RAM, an d an NVIDIA GeForce GTX 1080 graphics card. The tracked physical space hosting the VR applications is 4m × 4m. We used Unity 2019.1 to implement our VR collaborative manipulation tasks. The virtual environments were rendered at 90fps for each eye.

Manipulation implementation. We use object manipulation method in [9]. When the user keeps holding the "on" button, the translation and rotation of the handle are mapped to the virtual space according to 1:1. The user can repeat the action to move the object away or close or reach a total rotation angle beyond the wrist limit, e.g. keep press 'on' button, move, release the button, place the hand, and repeat). In our implementation, we use the right hand to move the object, and use the left hand to rotate object. The up and down direction of the joystick on the handle are used to modify the object's scale uniformly.

Task. The task of pilot user research is to require participants to manipulate the object to the target position. Two people form a group to manipulate the object collaboratively. Twelve people formed a total of six groups: 2 groups with $Table$ scene, 1 for $Bunny$ scene, and 3 for $Pipe$ scene.

Procedure. We random set one participant as the dominant manipulator in the first frame. Then the non-dominant user can press 'view' button on the handle to look at the view of the dominant manipulator, and get the dominance by pressing 'switch' button on the handle if he thinks his own view is better. This process is repeated until the users finish the manipulation. The views of two users and their scores are recorded when the non-dominant user presses 'switch' button on the handle, and the view of the non-dominant user is set as 'better view' of each view pair.

Metric. We measure the effectiveness of our viewpoint quality evaluation function with a metric named as $AccuracyRate$. After the participants perform and complete the manipulation tasks, we check all view pairs generated during the whole process and their viewpoint qualities. We count the number of views n_{bl} with better viewpoint qualities in the view pair that are better views, and calculate the ratio of n_{bl} to the number of pairs n_p to get $AccuracyRate$.

Results. In our pilot user study, n_p is 134, n_{bl} is 132, therefore $AccuracyRate$ is 98.5%, which means that most people subjectively believe that the views that are better for manipulating object at a certain moment are consistent with the views that have higher scores calculated by using our viewpoint quality evaluation function. Figure 3 shows some view pairs of user A (line 1) and user B (line 2) when the non-dominator pressed 'switch' button. The image marked with a check mark is the better view user select at that moment, and the image marked in the green frame is the view with a higher score calculated using our viewpoint quality evaluation function. In most cases, the images with high viewpoint quality selected by our method are basically the same as the better views subjectively perceived by the user. The image pair in column 5 gives an example of inconsistent results. The view point quality of the top image calculated by our viewpoint quality evaluation function is higher, because the area of the visible parts of the object and the target is larger, and the target is closer to the user. However, the user feels that the simpler the surrounding background of the manipulated object and the target, the easier it is to manipulate and match the object, so the user selects the bottom image. In summary, our viewpoint quality evaluation function is effective in the most cases.

4.2 User Study Design

We designed a user study with three tasks to evaluate efficiency, accuracy, task load, and presence of our methods. The hardware settings and manipulation implementation used in the user study are the same as those used in the pilot user study.

Participants. We have recruited 36 participants, 30 males and 6 females, between 20 and 30 years old. 24 of our participants had VR experience before. Participants had normal and corrected vision, and none reported vision or balance disorders. Two people form a group to manipulate the object collaboratively. There are 5 control conditions and an experimental condition with our method. All groups are required to participate in each task of the experimental conditions and the five control conditions.

Conditions CC_1, CC_3 and CC_5 are for two intuitive methods and the method in [14]. CC_1 is for the first-come-first-manipulate, CC_3 is for active switching dominance, CC_5 is for the method with a third director to assign the dominance. CC_2 is last-come-first-manipulate,

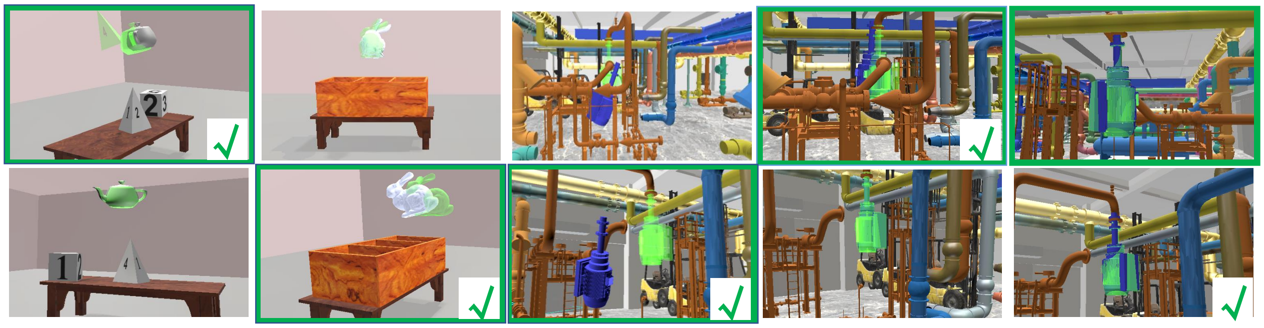

Figure 3: Image pair comparison in the pilot user study. Column 1 is for scene *Objects*, column 2 is for scene *Bunny* and Columns 3-5 are for scene *Pipe*. The images with green check mark are the better views users select at that moment, and the images in the green frames are the views with higher viewpoint qualities evaluated with our method.

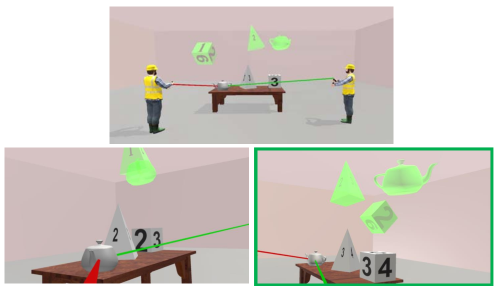

Figure 4: The first task $T1$ of our user study. Two users are manipulating the teapot to the green target position (top). The bottom images show the views seen from two viewpoints of two users. The viewpoint quality of the image with the green frame is higher.

which is a variant of the method of CC_1. Both CC_1 and CC_2 used a competitive collaboratively approach. For each frame, the dominance to manipulate the object is determined automatically. The difference between CC_1 and CC_2 is the strategy for determining who gets dominance. In CC_1, the user who manipulates the object first in the frame get the dominance, while in CC_2, the user who manipulate the object at last in the frame get the dominance. CC_4 is for active switching dominance after viewing the dominantor's view, which is a variant of the method of CC_3. Both CC_3 and CC_4 let the non-dominant user get the dominance manually. In CC_3, the non-dominant user can get the dominance by pressing 'switch' button on the handle when he thinks his view is good. In CC_4, the non-dominant user need to have a look at the dominant manipulator's view by pressing 'view' button on the handle, and press 'switch' button to get the dominance if he thinks his own view is better than the dominator's view. The experimental condition (*EC*) is with our method.

Task 1. In the first task (T1), two participants are required to manipulate the cube, triangular pyramid, and teapot on the table (*Table* scene) to the predefined target position collaboratively (Figure 4). These three objects and corresponding target positions appear in the scene in the order of cube, triangular pyramid and teapot. After the user completes manipulations on an object, the next object and the target location will be displayed immediately. The size of *Table* scene is 5m x 5m, and two users are placed at a random locations of the scene in the initialization. After the user manipulate the teapot to the desired position, the task is complete.

Task 2. In the second task (T2), two participants are required to manipulate *Bunny* to the target position collaboratively. At the beginning, Bunny has been placed in a wooden box, which provides more occlusions from two user's viewpoint (Figure 5). The size of *Bunny* scene is 5m×5m, and two users are placed at a random locations of the scene in the initialization. After users manipulate *Bunny* to the desired location, the task is complete.

Task 3. In the third task (T3), two participants are required to manipulate a piece of blue pipe to the target position collaboratively. There are many occlusions in the scene, so the visibility of views rendered at different viewpoints varies greatly. The size of *Pipe* scene is 16m×12m (Figure 1), and two users are placed at a random locations of the scene in the initialization. When the user finish to manipulate the blue pipe to the target position, the task is completed.

Procedure. For CC_1,CC_2,CC_3,CC_4, two participants form a group for collaborative manipulation. For CC_5, in addition to the two collaborative manipulators, we randomly designated one participant as the director among the remaining 34 participants. All groups performed the three co-manipulation tasks with all conditions in random order. The minimum interval between the tasks is one day and the maximum interval is three days. For each task, participants practice for 1 minute before the task starts. When both users point to the object need to be manipulated, our system starts recording time and other objective metrics. We tell the participants that we will record and evaluate the task completion time, which indirectly encourages them to complete the task as soon as possible.

Metrics. Task performance was measured with the following objective metrics: (1) Task completion time, in seconds, represents the time from when both collaborators point to the object until they both press 'end' button to confirm the end of the manipulation; (2) rotation error, in degrees, indicates the angle difference between the local coordinate system of the manipulated object and the target coordinate system when the last collaborator pressed 'end' button. If the angle difference of the three coordinate axes is α, β, γ, the rotation error is $\sqrt{\alpha^2+\beta^2+\gamma^2}$; (3) position error, in millimeters, represents the distance from the center of the manipulated object to the center of the target position when the last collaborator presses 'end' button; (4) scale error, in times, represents the absolute value of the ratio difference between the diagonal length of the bounding box of the manipulated object and the diagonal length of the target's bounding box when the last collaborator presses 'end' button; (5) participation, which is estimated based on the equality of the manipulation time of each participant [9]. We also evaluated the perception with three subjective metrics: user task load, measured with the standard NASA TLX questionnaire [10, 11], user sense of presence in the VE, measured with the standard Igroup Presence Questionnaire (IPQ) [21], and user simulator sickness, measured with the standard simulator sickness questionnaire (SSQ) [13].

Statistical analysis. For each metric, the values of *EC* were compared to those of CC_1, CC_2, CC_3, CC_4 and CC_5 respectively. First, the normality of the data was assessed using the Shapiro-Wilk test [22]. Then the comparison was performed with a repeated measures ANOVA [7] if the values showed a normal distribution. When values did not follow a normal distribution, the comparison was performed using a Wilcoxon signed-rank test [18]. In addition to the

p value of the statistical test, we also estimated the size of the effect using Cohen's d [3].

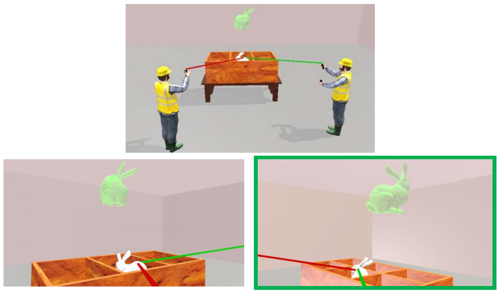

Figure 5: The second task $T2$ of our user study. Two users are manipulating Bunny to the green target position (top). The bottom images show the views seen from two viewpoints of two users. The viewpoint quality of the image with the green frame is higher.

4.3 Results

The user study results for evaluate task performance (Section 4.3.1) and perception (Section 4.3.2) were reported and discussed.

4.3.1 Task performance

Task completion time. Table 1 gives the task completion time. The third column gives the average and standard deviation, and the fourth column gives the relative time cost reduction from EC to CC. The fifth to seventh columns provide statistical information about the difference between the EC and CC. Statistical significance is indicated by an asterisk.

Compared with all control conditions of all three tasks, our method significantly improves the task time performance, and the effect size ranges from "Large" to "Huge". The reasons why our method is more efficient may be: (1) the visibility of the object and the target in the view observed by the user who actually manipulates the object is good, and the angle and distance between the manipulator and the target are more suitable for manipulation; (2) the assignment of the entire dominance is automatic and does not require the user to determine when to switch the dominance.

Generally, CC_1 and CC_2 are more efficient than CC_3 and CC_4, because CC_1 and CC_2 automatically switch dominance, while CC_3 and CC_4 require users to manually switch dominance. CC_4 is much slower than CC_3 because CC_3 switches dominance in one step, while CC_4 switches dominance in two steps, that is, non-dominant users first observe the view of the dominant manipulator and then decide whether to switch dominance, which takes time. For the task in the simple scenes (T1 and T2), the efficiency of CC_5 is similar to CC_3 and CC_4, but if the scene has many occlusions, the time cost of CC_5 is very large.

Rotation error. Table 2 gives the rotation error of all conditions for these three tasks. Compared with all control conditions of all three tasks, our method reduced rotation error significantly, and the effect size ranges from "Large" to "Huge". The main reason may be that the angle and distance between the manipulator and the target are more convenient to observe the angle difference between the manipulated object and the target, and then the manipulator can reduce the angle difference through your own manipulation easily. After the tests, we interviewed the participants with the question 'Is the rotation error related to the viewpoint?', and all participants agreed that rotation error is highly related with what the user observes during manipulation.

Position error and scale error. Table 3 and Table 4 show the position errors and the scale errors of all conditions for these three tasks. Compared to the control conditions, our method reduced the position error in most cases, except for vs CC_3 and CC_5 for T1, vs CC_3 and CC_4 for T2, vs CC_1 for T3. However, the difference between our method and the control condition is not significant , and the effective size ranges from 'very Small' to 'Small'. Compared with all control conditions of T1 and T2, our method reduced scale errors, but the reductions are not significant. This may be because the position and scale errors of the control conditions are already very small, and it is very difficult to further reduce these two errors. For T1, Table 3 and Table 4 give the sum of the position errors and scale errors of the three objects. From the table result, it can be seen that the position error of all single objects is less than 1 cm, and the scale error is less than 0.05.

Participation. Fig. 6 shows the user participation of all conditions for these three tasks, and a higher score always means better. Significant differences are noted with a asterisk. For T1, T2, and T3, compared with CC_1, CC_3 and CC_4, the user participation of EC is significantly improved.

CC_1 is first come first manipulating. Therefore, the dominator keeps translating or rotating the object by holding 'on' button or keeps scaling the object by pushing or pulling the joystick, so the non-dominator only can get the dominance when the dominator releases the button or stops pushing or pulling. After trying several

Table 1: The completion time, in seconds.

Task	Condi -tion	Avg $\pm$ std. dev.	$(CC_i\text{-}EC)$ / CC_i	p	Cohen's d	Effect size
T1	EC	172.0 ± 35.74				
	CC_1	255.0 ± 73.87	32.5%	$< 0.001^*$	1.43	Very Large
	CC_2	228.0 ± 70.72	24.6%	0.003^*	1.00	Large
	CC_3	252.0 ± 53.20	31.7%	$< 0.001^*$	1.77	Very Large
	CC_4	318.8 ± 52.16	46.0%	$< 0.001^*$	3.28	Huge
	CC_5	276.7 ± 65.74	37.8%	$< 0.001^*$	1.98	Very Large
T2	EC	77.4 ± 10.48				
	CC_1	121.75 ± 39.40	37.8%	$< 0.001^*$	1.54	Very Large
	CC_2	119.80 ± 40.08	35.4%	$< 0.001^*$	1.45	Very Large
	CC_3	142.00 ± 65.53	45.5%	$< 0.001^*$	1.38	Very Large
	CC_4	129.60 ± 56.65	40.3%	$< 0.001^*$	1.28	Large
	CC_5	125.75 ± 20.84	38.4%	$< 0.001^*$	2.93	Very Large
T3	EC	92.8 ± 11.65				
	CC_1	141.0 ± 42.37	34.2%	$< 0.001^*$	1.55	Very Large
	CC_2	144.0 ± 59.31	35.6%	$< 0.001^*$	1.20	Large
	CC_3	166.0 ± 55.81	44.1%	$< 0.001^*$	1.82	Very Large
	CC_4	166.0 ± 17.25	44.1%	$< 0.001^*$	4.97	Huge
	CC_5	177.9 ± 47.83	47.9%	$< 0.001^*$	2.45	Huge

Table 2: The rotation error, in degrees.

Task	Condi -tion	Avg $\pm$ std. dev.	$(CC_i\text{-}EC)$ / CC_i	p	Cohen's d	Effect size
T1	EC	7.29 ± 1.38				
	CC_1	9.65 ± 2.70	24.5%	0.001^*	1.1	Large
	CC_2	11.73 ± 4.84	37.9%	$< 0.001^*$	1.25	Large
	CC_3	8.82 ± 1.97	17.4%	0.008^*	0.90	Large
	CC_4	9.97 ± 4.56	26.9%	0.01^*	0.83	Large
	CC_5	8.60 ± 1.79	15.2%	0.01^*	0.82	Large
T2	EC	3.17 ± 0.47				
	CC_1	5.97 ± 2.84	46.8%	$< 0.001^*$	1.37	Very Large
	CC_2	5.20 ± 2.66	39.0%	0.002^*	1.06	large
	CC_3	5.91 ± 1.79	46.3%	$< 0.001^*$	2.09	Huge
	CC_4	4.13 ± 1.13	23.1%	0.001^*	1.1	Large
	CC_5	4.21 ± 1.28	24.6%	0.002^*	1.07	Large
T3	EC	1.25 ± 0.40				
	CC_1	2.57 ± 0.81	51.4%	$< 0.001^*$	2.06	Very Large
	CC_2	2.86 ± 1.31	56.4%	$< 0.001^*$	1.67	Very Large
	CC_3	2.92 ± 1.97	57.3%	$< 0.001^*$	1.18	Large
	CC_4	2.53 ± 1.50	50.7%	$< 0.001^*$	1.17	Large
	CC_5	1.99 ± 0.78	37.3%	$< 0.001^*$	1.19	Large

Table 3: The position error, in millimeters.

Task	Condi -tion	Avg ± std. dev.	$(CC_i\text{-}EC)$ / CC_i	p	Cohen's d	Effect size
T1	EC	11.7 ± 1.9				
	CC_1	13.1 ± 6.0	10.7%	0.335	0.32	Small
	CC_2	11.8 ± 2.4	0.9%	0.884	0.05	Very Small
	CC_3	11.4 ± 3.6	-3.2%	0.70	0.13	Very Small
	CC_4	12.1 ± 2.2	2.7%	0.631	0.16	Very Small
	CC_5	11.2 ± 2.2	-4.4%	0.449	0.27	Small
T2	EC	5.3 ± 1.5				
	CC_1	5.9 ± 2.3	18.8%	0.360	0.30	Small
	CC_2	5.7 ± 2.0	5.6%	0.581	0.18	Very Small
	CC_3	4.9 ± 3.4	-8.2%	0.638	0.15	Very Small
	CC_4	5.0 ± 1.1	-5.9%	0.492	0.23	Very Small
	CC_5	5.5 ± 1.9	18.8%	0.360	0.09	Very Small
T3	EC	2.2 ± 0.5				
	CC_1	2.0 ± 0.8	-5.4%	0.614	0.16	Very Small
	CC_2	2.4 ± 0.6	8.3%	0.294	0.34	Small
	CC_3	2.2 ± 0.6	1.7%	0.839	0.07	Very Small
	CC_4	2.2 ± 0.2	1.1%	0.856	0.06	Very Small
	CC_5	2.3 ± 0.5	6.6%	0.38	0.28	Small

times and failing several times, non-dominator will be very frustrated and unwilling to take the initiative to gain dominant position. In CC_3 and CC_4, the non-dominant user needs to manually obtain a dominance, which requires he to have great operation willingness and patience. Usually after a period of manipulation, the user's willingness and patience to operate decrease.

Compared with CC_2, the user participation of EC has improved, but not significantly improved. CC_2 is also a competitive method of collaboration, but two users have a certain probability of gaining a dominant position in each frame, so they have been actively participating in the manipulation of the object, rather than a user watching another user continue to manipulate, waiting for him to release dominance.

For with CC_5, we only computed the participation of two users who manipulated the object. Since the dominance is assigned by the director, in most cases, the director allowed both users participate in the manipulation for a similar time.

Table 4: The scale error, in times.

Task	Condi -tion	Avg ± std. dev.	$(CC_i\text{-}EC)$ / CC_i	p	Cohen's d	Effect size
T1	EC	0.024 ± 0.020				
	CC_1	0.025 ± 0.015	2.2%	0.92	0.03	Very Small
	CC_2	0.033 ± 0.018	26.2%	0.10	0.45	Small
	CC_3	0.026 ± 0.010	7.7%	0.69	0.13	Very Small
	CC_4	0.028 ± 0.021	14.3%	0.55	0.20	Very Small
	CC_5	0.029 ± 0.014	16.5%	0.39	0.28	Small
T2	EC	0.024 ± 0.020				
	CC_1	0.030 ± 0.020	20.2%	0.36	0.29	Small
	CC_2	0.032 ± 0.010	24.4%	0.17	0.45	Small
	CC_3	0.025 ± 0.020	3.9%	0.88	0.05	Very Small
	CC_4	0.025 ± 0.010	3.3%	0.88	0.05	Very Small
	CC_5	0.026 ± 0.015	4.8%	0.84	0.07	Small

4.3.2 Perception

We have also investigated task load, presence and simulator sickness using standard questionnaires.

We used Raw TLX [10,11] to measure the task load. We averaged the scores of six Raw TLX task load problems. The task load value does not follow a normal distribution, so the Wilcoxon signed-rank

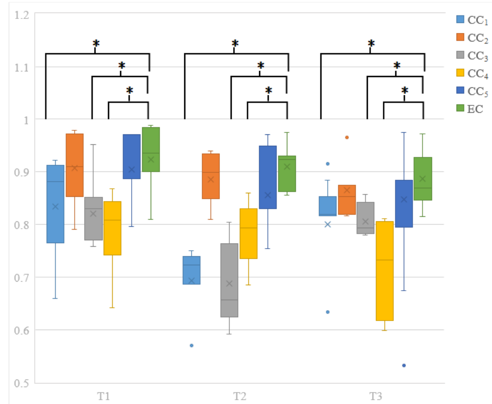

Figure 6: The participation score per task and per condition.

test was used to analyze the difference. Figure 7) shows the results of task load. Compared with all control conditions, the task load of our method is reduced significantly.

Our method is similar to CC_1 and CC_2 in that it automatically determines the dominance of each frame, but the difference is $CC1$ and $CC2$ require the active interaction of the user handle to determine the dominance, while our method directly determines the dominance based on the viewpoint, without additional handle interaction. In CC_3 and CC_4, non-dominant user need to find a suitable time to switch dominance, so the task load is higher. In CC_5, we computed the task load for all three users in each group. The task load of the director was much higher than two manipulators since he need to observe and make a decision to switch the dominance all the time.

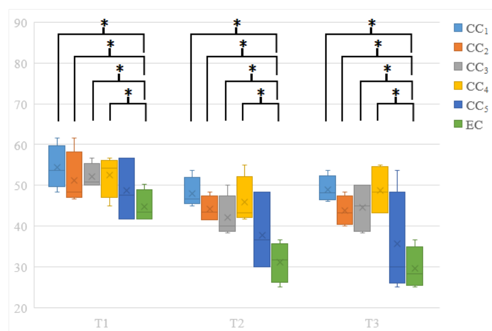

Figure 7: Participant task load per task and per condition.

We measured users' sense of presence in the VE using the standard Igroup Presence Questionnaire (IPQ) [21]. Table 5 shows our IPQ measurements broken down into the usual categories of general presence (GP), spatial presence (SP), involvement (INV), and realism (REAL). The experimental condition produces IPQ scores similar to both control conditions, and no difference is significant.

We used the standard simulator sickness questionnaire (SSQ) [13] (table6) to measure simulator sickness. The SSQ was managed before and after the experiment for each task and each condition. The SSQ scores are not normally distributed, and the differences before and after the Wilcoxon signed-rank test were used. These differences are not statistically significant. We concluded that in these experiments, the discomfort of the simulator has nothing to do with EC compared to CC_1, CC_2, CC_3, CC_4 and CC_5. No participant reported visual fatigue through related SSQ questions.

In order to investigate the influence of time decay on manipulation, we conducted a comparative experiment between our method with time decay (EC) and without time decay (EC_{nd}). Table 7 gives the results of the task completion time (CT), rotation error(RE), position

Table 5: Igroup Presence Questionnaire data.

Task	Condition	GP	SP	INV	REAL
T1	EC	4.2 ± 0.6	3.9 ± 0.4	3.2 ± 0.5	3.1 ± 0.7
	CC_1	4.3 ± 0.3	4.1 ± 0.3	3.1 ± 0.2	2.4 ± 0.5
	CC_2	4.0 ± 0.2	4.4 ± 0.3	3.5 ± 0.5	2.9 ± 0.3
	CC_3	4.2 ± 0.4	4.2 ± 0.5	3.6 ± 0.6	2.9 ± 0.8
	CC_4	4.1 ± 0.4	4.6 ± 0.1	3.1 ± 0.7	2.7 ± 1.0
	CC_5	4.3 ± 0.3	3.6 ± 0.2	3.5 ± 0.4	3.1 ± 1.1
T2	EC	4.9 ± 0.3	4.5 ± 0.6	3.7 ± 0.3	3.0 ± 0.6
	CC_1	4.1 ± 0.5	4.4 ± 0.7	4.0 ± 0.6	3.1 ± 0.9
	CC_2	4.6 ± 0.6	4.3 ± 0.5	3.0 ± 0.7	2.8 ± 0.6
	CC_3	4.1 ± 0.7	4.9 ± 0.9	3.4 ± 0.9	2.9 ± 0.5
	CC_4	4.2 ± 0.9	3.9 ± 0.5	3.6 ± 0.6	2.6 ± 1.2
	CC_5	4.3 ± 0.6	4.7 ± 0.3	3.5 ± 0.5	2.7 ± 0.8
T3	EC	4.2 ± 0.8	5.1 ± 0.5	3.3 ± 0.6	3.2 ± 0.8
	CC_1	4.3 ± 0.7	4.3 ± 0.6	3.2 ± 0.4	3.3 ± 1.1
	CC_2	4.9 ± 0.8	4.2 ± 0.4	3.3 ± 0.6	3.1 ± 1.2
	CC_3	4.5 ± 0.2	4.5 ± 0.2	3.4 ± 0.8	3.4 ± 0.9
	CC_4	4.8 ± 0.6	4.2 ± 0.5	3.5 ± 0.7	3.5 ± 0.8
	CC_5	4.5 ± 0.8	4.1 ± 0.6	3.8 ± 0.4	3.5 ± 1.1

Table 6: Simulator Sickness Questionnaire data.

Task	Condition	preAvg $\pm$ std. dev.	postAvg $\pm$ std. dev.	p
T1	EC	1.74 ± 0.77	2.08 ± 0.62	0.41
	CC_1	1.84 ± 0.70	2.42 ± 0.74	0.19
	CC_2	1.77 ± 0.49	2.09 ± 0.81	0.43
	CC_3	1.73 ± 0.46	2.02 ± 0.47	0.30
	CC_4	1.83 ± 0.50	2.47 ± 0.83	0.13
	CC_5	1.87 ± 0.55	2.74 ± 0.93	0.07
T2	EC	1.44 ± 0.45	1.75 ± 0.68	0.38
	CC_1	1.55 ± 0.67	2.12 ± 1.31	0.36
	CC_2	1.48 ± 0.71	1.76 ± 1.00	0.59
	CC_3	1.57 ± 0.74	2.11 ± 1.32	0.40
	CC_4	1.44 ± 0.45	1.75 ± 0.69	0.38
	CC_5	1.86 ± 0.35	2.44 ± 0.74	0.11
T3	EC	1.80 ± 0.78	2.07 ± 0.36	0.17
	CC_1	1.88 ± 0.54	2.24 ± 0.88	0.17
	CC_2	1.77 ± 0.48	2.12 ± 0.56	0.26
	CC_3	1.72 ± 1.04	2.11 ± 1.20	0.16
	CC_4	1.89 ± 0.40	2.26 ± 0.60	0.21
	CC_5	1.87 ± 0.58	2.54 ± 0.78	0.12

error(PE), scale error(SE) and participation (PS) of our method without time decay for Task 1. Statistical significance is indicated by an asterisk. Compared with $ECnd$, EC has made significant improvements in the reduction of rotation error and position error, as well as the increase in participation, and the effect size ranges from ”Large” to ”Very Large”.

5 Conclusions, Limitations, and Future Work

We have proposed a new VR collaborative manipulation method based on viewpoint quality. Our method analyzes different views rendered from different viewpoints of multiple users, uses a new function containing three factors to evaluate the viewpoint quality, and automatically determines the dominance of object manipulation. Compared with the previous method, our method is proved to be more efficient and accurate, the user's task load is smaller, the multi-user participation is more balanced, and it does not reduce the user's presence and increase the simulator sickness.

One limitation of our method is that the proposed viewpoint quality function does not consider the appearance details of the surface of

Table 7: The task performance of EC_{nd} in Task 1.

Metric	Avg $\pm$ std. dev.	(EC_{nd}-EC) / EC_{nd}	p	Cohen's d	Effect size
CT	170.4±35.01	-0.90%	0.890	0.05	Very Small
RE	9.02±1.82	19.20%	0.002*	1.07	Large
PE	16.2±5.5	27.60%	0.001*	1.09	Large
SE	0.032±0.028	25.00%	0.312	0.33	Small
PS	0.797±0.070	15.80%	$< 0.001^*$	1.91	Very Large

the manipulated object. For example, if the surface of the object has a detailed texture, it is very helpful for the user to match the object and the target. A future work is to integrate appearance features into the viewpoint quality evaluation function. Another limitation is that the value calculated by our viewpoint quality function may be affected by the geometric details on the target surface. When the viewing direction is perpendicular to the target or parallel to the target, it is difficult to manipulate the object and match the target. We think oblique observation is more useful for manipulation and matching. Therefore, we use the weighted sum of the target's depth and distance factors and the target's visible area factor to ensure that the point of view obliquely viewing the target has a higher score. For a target with planar surface, the higher the score of the weighted sum, the more oblique the viewing direction and the target surface. If there are undulating geometric details on the target surface, the value of the depth and distance factor may increase, which may result in ”false high” viewpoint quality. A future direction is to make the viewpoint quality function insensitive to geometric details. Another limitation is that although our method is suitable for 2 or more collaborators, we only tested each group of 2 collaborators in the user study, and did not complete the test for more collaborators because in the situation of COVID-19, multiple people are not allowed to gather. In the future, multi-person collaborative experiments can be carried out to test whether our method is practical.

In each frame, our method determines the manipulation dominance and only allows one user to manipulate the object. This may not be efficient when the object is far away from the target. Future work can divide the manipulation process into stages adaptively. When the object is far away from the target, the manipulation of multiple users can be combined in each frame, so that the object can approach the target as soon as possible, and then the users can start refined manipulation and matching. Our current method is to determine the dominance based on the quality of the viewpoint, and the user with the dominance can use all types of manipulations to manipulate the object: translation, rotation, and scaling. One future work is a more refined method of dominance determination, which can further improve the efficiency. For example, at a certain moment, according to the viewpoint quality, it is determined that a certain user is only suitable for translating the object, while another user is suitable for rotating manipulations. Another future work is to use viewpoint quality to switch dominance and guide the users' navigation path in virtual environment, which may improve the accuracy and efficiency of manipulation. Since sharing gaze information between collaborators in some applications can improve collaboration efficiency, integrating gaze awareness into our method is also a future work.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China through Projects 61932003 and 61772051, by National Key R&D plan 2019YFC1521102, by the Beijing Natural Science Foundation L182016, by the Beijing Program for International S&T Cooperation Project Z191100001619003, by the funding of Shenzhen Research Institute of Big Data (Shenzhen 518000).

References

[1] N. Baron. Collaborativeconstraint: Ui for collaborative 3d manipulation operations. In 2016 IEEE Symposium on 3D User Interfaces (3DUI), pp. 273–274, 2016. doi: 10.1109/3DUI.2016.7460076

[2] M. L. Chenechal, J. Lacoche, J. Royan, T. Duval, V. Gouranton, and B. Arnaldi. When the giant meets the ant an asymmetric approach for collaborative and concurrent object manipulation in a multi-scale environment. In 2016 IEEE Third VR International Workshop on Collaborative Virtual Environments (3DCVE), pp. 18–22, 2016. doi: 10.1109/3DCVE.2016.7563562

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

[4] T. Duval, A. Lecuyer, and S. Thomas. Skewer: a 3d interaction technique for 2-user collaborative manipulation of objects in virtual environments. In 3D User Interfaces (3DUI'06), pp. 69–72, 2006. doi: 10.1109/VR.2006.119

[5] S. Freitag, B. Weyers, and T. W. Kuhlen. Automatic speed adjustment for travel through immersive virtual environments based on viewpoint quality. In 3d User Interfaces, 2016.

[6] S. Freitag, B. Weyers, and T. W. Kuhlen. Interactive exploration assistance for immersive virtual environments based on object visibility and viewpoint quality. In IEEE Virtual Reality 2018, 2018.

[7] A. Gelman. Analysis of variance . Quality Control Applied Statistics, 20(1):págs. 295–300, 2005.

[8] P. C. Gloumeau, W. Stuerzlinger, and J. H. Han. Pinnpivot: Object manipulation using pins in immersive virtual environments. IEEE Transactions on Visualization and Computer Graphics, PP(99):1–1, 2020.

[9] J. G. Grandi, H. G. Debarba, and A. Maciel. Characterizing asymmetric collaborative interactions in virtual and augmented realities. In IEEE Conference on Virtual Reality and 3D User Interfaces, 2019.

[10] S. Hart. Nasa-task load index (nasa-tlx); 20 years later. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 50:904 – 908, 2006.

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

[12] T. Kamada and S. Kawai. A simple method for computing general position in displaying three-dimensional objects. Comput. Vis. Graph. Image Process., 41:43–56, 1988.

[13] R. Kennedy, N. E. Lane, K. Berbaum, and L. Mg. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. The International Journal of Aviation Psychology, 3:203–220, 1993.

[14] W. Lages. Ray, camera, action! a technique for collaborative 3d manipulation. In 3d User Interfaces, 2016.

[15] D. Mendes, M. Sousa, R. Lorena, A. Ferreira, and J. Jorge. Using custom transformation axes for mid-air manipulation of 3d virtual objects. In Acm Symposium, pp. 1–8, 2017.

[16] M. S. Pinho, D. Bowman, and C. Freitas. Cooperative object manipulation in collaborative virtual environments. Journal of the Brazilian Computer Society, 14(2):53–67, 2008.

[17] D. Plemenos and M. Benayada. Intelligent display in scene modelling. new techniques to automatically compute good views. In GraphiCon'96. Saint Petersburg (Russia), July 1996.

[18] D. Rey and M. Neuhäuser. Wilcoxon-signed-rank test. In International Encyclopedia of Statistical Science, 2011.

[19] K. Riege, T. Holtkamper, G. Wesche, and B. Frohlich. The bent pick ray: An extended pointing technique for multi-user interaction. In 3D User Interfaces (3DUI'06), pp. 62–65, 2006. doi: 10.1109/VR.2006.127

[20] J. D. Ruddle R., Savage J. Symmetric and asymetric action integration during cooperative object manipulation in virtual environments. ACM Transactions on Computer-Human Interaction, 9:285–308, 2002.

[21] T. W. Schubert. The sense of presence in virtual environments: A three-component scale measuring spatial presence, involvement, and realness. Z. für Medienpsychologie, 15(2):69–71, 2003.

[22] S. S. Shaphiro and M. B. Wilk. An analysis of variance test for normality (complete samples). Biometrika, 52:591–611, 1965.

[23] L. P. Soares, R. Kopper, and M. S. Pinho. Ego-exo: A cooperative manipulation technique with automatic viewpoint control. In 2018 20th Symposium on Virtual and Augmented Reality (SVR), 2018.

[24] D. Sokolov and D. Plemenos. Viewpoint quality and scene understanding. In M. Mudge, N. Ryan, and R. Scopigno, eds., The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST. The Eurographics Association, 2005. doi: 10.2312/VAST/VAST05/067-073

[25] D. Sokolov, D. Plemenos, and K. Tamine. Viewpoint quality and global scene exploration strategies. In GRAPP 2006: Proceedings of the First International Conference on Computer Graphics Theory and Applications, Setúbal, Portugal, February 25-28, 2006, 2006.

[26] P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint selection using viewpoint entropy. In Vision Modeling Visualization Conference, 2001.

[27] H. Wang and B. E. Shi. Gaze awareness improves collaboration efficiency in a collaborative assembly task. In Proceedings of the 11th ACM Symposium on Eye Tracking Research amp; Applications, ETRA '19. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3317959.3321492