

Received July 15, 2020, accepted August 16, 2020, date of publication August 24, 2020, date of current version September 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019012

Automatic Control of Virtual Mirrors for Precise 3D Manipulation in VR

WANSU HA¹, MYUNG GEOL CHOI², AND KANG HOON LEE¹

¹Department of Computer Science, Kwangwoon University, Seoul 01897, South Korea

²Digital Media Department, The Catholic University of Korea, Seoul 06591, South Korea

Corresponding author: Kang Hoon Lee (kang@kw.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MOE) under Grant NRF-2018R1D1A1B07043995, and in part by the Research Grant of Kwangwoon University, in 2018.

ABSTRACT Popular virtual reality systems today allow us to experience highly immersive applications in which virtual objects are realistically perceived via stereoscopic displays and can be directly manipulated based on hand-eye coordination in a very similar way as in the real world. However, the insufficiency of sensory feedback as well as the limited degrees-of-freedom of input motion still hinders precise and elaborate manipulation in virtual reality. Aiming at more precise 3D manipulation, we present a new method of extending the user's spatial perception ability with the 'virtual mirrors', which expose the hidden spatial information of given virtual scenes to the user. The movement of a virtual mirror is automatically controlled by solving an optimization problem iteratively, in which the objective function prefers the placement of the mirror that can highlight the spatial relationship between the manipulated object and the object nearest to it. The optimization process is handled efficiently for each time step based on our method for finding the closest gap between any two objects based on the OBB (oriented bounding box) trees and our sampling-based approximate approach to the optimization problem. The usefulness of our method is demonstrated by several pilot applications under various usage scenarios, such as assembling construction toys and solving 3D dissection puzzles. The quantitative results of our user study show that the virtual mirror is very helpful in increasing the precision in 3D manipulation tasks in virtual reality.

INDEX TERMS 3D manipulation, virtual reality, mirror reflection, auxiliary view, bounding volume hierarchy.

I. INTRODUCTION

We have observed the rapid development and popularization of immersive virtual reality technologies for the last decade. One of the key advantages of virtual reality over the other interaction technologies is that the user immersed in virtual reality is able to directly manipulate 3D objects by coordinating his/her hands and eyes in a very similar way as in the real world. Such a capability for intuitive interaction particularly stands out in the applications requiring intensive manipulation of virtual objects, including 3D modeling, architectural design, manufacturing, medical surgery, and military simulation. For example, 3D modeling tools for virtual reality, such as Oculus Medium and Gravity Sketch, allow the user to intuitively edit polygonal models in much the same way that sculptors interact with real-world materials.

Despite its strength in intuitive interaction, virtual reality systems often suffer from the lack of precision in

manipulating objects in virtual worlds. One of the reasons is the limited degrees of freedoms (DOFs) of input motion. Typical motion controllers nowadays can trace the rigidbody motion of 6 DOFs, but mostly fall short of the ability to track the skeletal motion of higher DOFs, such as finger movements, which would be helpful for minute control of the position and orientation of virtual objects. As another reason, the lack of the tactile feedback in virtual reality could significantly degrade the user's spatial perception ability. In the real world, if two objects were in contact with each other, the user would heavily resort on tactile feedback to finely control the degree of alignment between those objects. However, the resolution of the tactile feedback provided in the virtual reality systems today is much more limited than in the real world, and thus, is not very helpful for such detailed control.

One natural approach to precise manipulation based on these observations would be to develop advanced sensors and actuators that can reproduce the real-world experiences more faithfully. However, it is not very promising to produce such high-fidelity sensors and actuators that are cheap as

The associate editor coordinating the review of this manuscript and approving it for publication was Feng Lin.

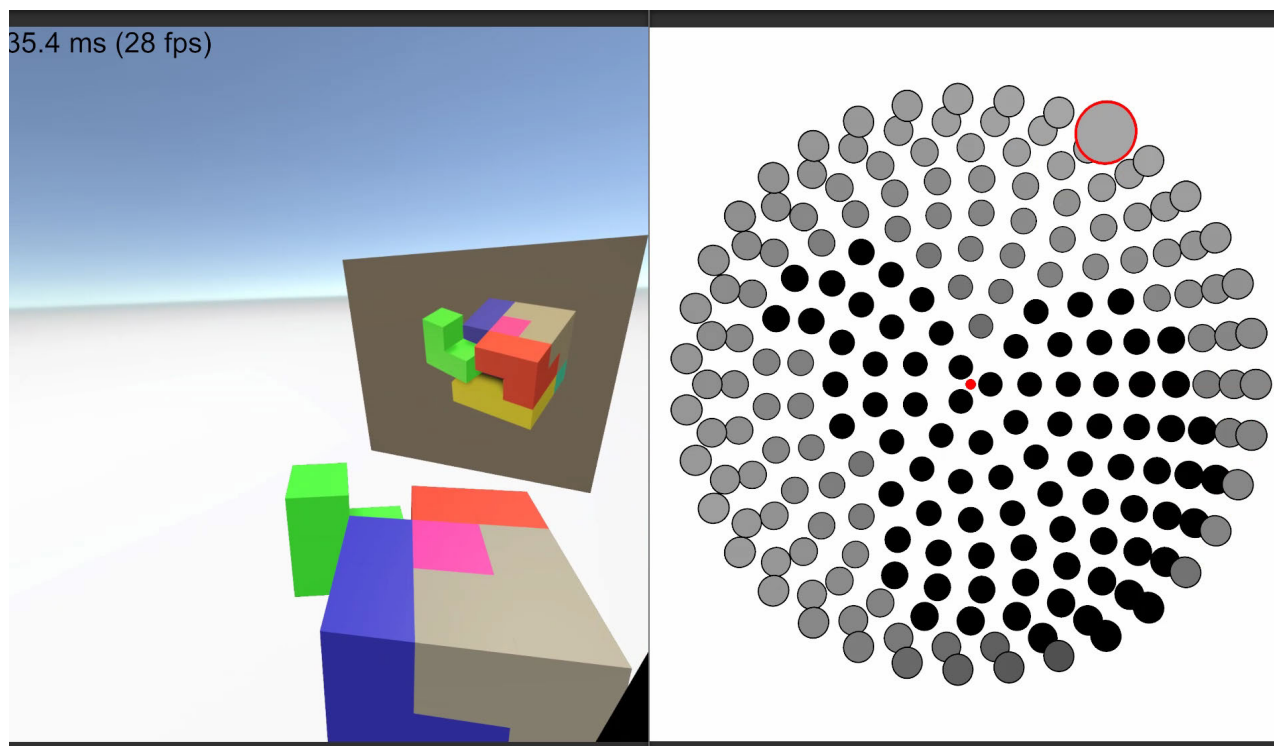


FIGURE 1. A screenshot from the Soma cube puzzle application for demonstrating the effectiveness of our approach. (Left) The user needs to find a missing region of the partially assembled cubical structure and to precisely align the given piece, colored green in the figure, with the missing region to solve the puzzle. The virtual mirror behind the cube is automatically repositioned and reoriented for each frame, in order to provide the user with additional visual information revealing the spatial relationships between the object manipulated by the user and the object nearest to it. (Right) Each new placement of the mirror is determined by scoring the candidate positions around the involved objects based on several optimization criteria and choosing the best position, as shown on the right. The brighter dots represent the higher scores at those positions, and the red circle denotes the position where the best score is acquired.

well as lightweight in the near future. Instead of pursuing the accurate imitation of the real-world experiences, we aim at extending the user's spatial perception ability in a surrealistic way that is only allowed in virtual reality, so that the user can acquire spatial information more comprehensively than in the real world and can perform spatial manipulation tasks more precisely than without such an augmented ability.

Inspired by the rear-view and side-view mirrors in the real-world vehicles, which expose spatial information hidden in the blind spots, we provide the user in our virtual reality system with a 'virtual mirror' that reveals the hidden sides of given virtual scenes from a diversity of views. Unlike the physical mirrors requiring manual adjustment for view selection, our virtual mirror automatically relocates and reorients itself to reflect the region of interest (ROIs) encompassing the object manipulated by the user and the object nearest to it (see Section III). For computational efficiency in identifying the ROIs, we accelerate the process of finding the closest gap between any given two objects by exploiting the well-known OBB tree data structures, which are typically used for collision detection (see Section IV). Our key contribution is in the method of deciding the optimal position and orientation of the virtual mirror at each time instant (see Section V). We present a set of evaluation functions to be optimized, each of which scores the quality of the reflected image under its own unique criterion, such as the size of the

mirror relative to the size of the entire view. The optimal position and orientation that maximizes the weighted sum of these functions are efficiently computed by reducing the number of variables, restricting the domain of optimization, and sampling candidate solutions. We implemented three types of practical applications requiring precise 3D manipulation and conducted preliminary user studies with these applications, which demonstrated the effectiveness of our approach in a qualitative manner (see Figure 1). In addition, we carried out a quantitative user study with one of the applications, that is, a 3D shape reconstruction puzzle, and found that our virtual mirror was certainly effective in increasing the precision in the spatial alignment tasks involved in the puzzle.

II. RELATED WORK

The development of interactive 3D graphics technology has increased the intuitiveness and precision of manipulating 3D objects in virtual environments [1]. The manipulation of 3D objects typically includes translation and rotation in the 3D space, which requires at least 6 DOFs (degrees of freedom). The traditional 2D input devices, such as mice and joysticks, provide only 2 DOFs, so we need some techniques for mapping between the 2D movement input and the 3D translation and/or rotation of a selected object [2]. For example, many 3D modeling tools today provide special widgets for 3D transformation, which allow the user to

select the translational or rotational axis at a time and to constrain a selected object to translate along or rotate around the selected axis [3]. Touch screens, which are widely used for mobile phones or tablet PCs nowadays, also provide 2D positional inputs similarly to the traditional input devices, but have relative strengths in that they usually support simultaneous multi-touch inputs. For example, in a method called the Z-technique, the first touch directly moves the object in the plane parallel to the view, while the backward-forward motion of a second touch indirectly moves the object along the line perpendicular to the view plane [4].

Immersive virtual reality systems enable the input motion of the real world to be directly transferred to the object motion of the virtual world via motion sensing devices of high DOFs, essentially removing the necessity of dimensional remapping techniques. For example, in a typical virtual reality system today, the user can press a button on a handheld motion controller to grab a virtual object, and then move the controller to translate and rotate the object in the same way as in the real world [5]. We employed this method, called the Simple Virtual Hand, in our experimental implementation with some tweaks. Despite its intuitiveness and naturalness, the Simple Virtual Hand often suffers from its limitation in the range of translation and rotation. The one-to-one mapping between the real and the virtual space naturally leads to the out-of-reach situations in which the user cannot interact with the distant objects that are out of reach of his/her hands. One simple remedy to this problem is to extend the length of the user's arm proportionally to the distance between the hand and the body of the user [6]. This technique can be combined with ray-casting when selecting distant objects, in which the ray casted from the user's hand can be regarded as a virtual arm of an infinite length [7]. Instead of scaling up the user's reach, some researchers have approached to the out-of-reach problem in the inverse direction such that the entire virtual world, which could be of enormous size, or a specific virtual object, which could be highly distant, is copied and scaled down to a proxy object that can be easily manipulated within the user's reach [8], [9].

The lack of precision is another key challenge in direct manipulation techniques for immersive virtual reality, which is mainly due to the limitations in human motor and perceptual capabilities. First, from the view of human motor capability, it is an inherently difficult task for the user to precisely locate his/her body parts, such as arms or hands, to specific positions in the air without additional support. The PRISM (Precise and Rapid Interaction through Scaled Manipulation) is a method to alleviate this problem by dynamically adjusting the "control-display" (C/D) ratio that determines the relationship between physical hand movements and the motion of the controlled virtual object [10]. Secondly, the visual information provided to the user from a single point of view is often insufficient to recognize the relative position and orientation of an object with respect to the nearby objects surrounding it, particularly when the objects have complicated shapes and occlude each other.

Providing auxiliary views in which the same scene is rendered from multiple points of views can facilitate the recognition of spatial relationships among nearby objects [11]. However, it is not natural in virtual reality to arrange a dedicated window for such an auxiliary view in front of the user's eyes in the 3D space. Instead of overlaying separately rendered views, our method provides the additional visual information via the virtual mirrors naturally embedded in the 3D space.

III. OVERVIEW

Our method assumes that the user is immersed in a 3D virtual world via a head-mounted display and is allowed to directly manipulate each individual object arranged in the world by using one or more handheld motion controllers of 6 DOFs. We do not impose restrictions on the appearance and behavior of virtual objects except that those can be selected, translated, rotated, and released based on the Simple Virtual Hand technique [5]. However, in our experiments, we have made several specific design decisions about how the user interacts with the objects in the world, such as the capability of assembling objects together, for our purposes of demonstration and user study, which will be described in detail in Section VI.

The key component of our method is a virtual mirror, which is continuously relocated and reoriented in the world during an object is manipulated, in order to reflect the manipulated object and the object nearest to it such that the spatial relationship between those two objects can be effectively recognized by the user. The virtual mirror appears when the user grabs an object, and disappears when the user releases the object. Given a new configuration of the manipulated object at each time instant, we find the object nearest to it and quickly identify the closest gap between the two nearby objects, which will be referred to as the region of interest (ROI), based on the OBB tree encompassing each object (see Section IV). Once the ROI has been identified, we decide the new position and orientation of the virtual mirror by sampling a set of candidate configurations of the mirror around the ROI and then choosing the optimal one based on our evaluation criteria (see Section V). The virtual mirror smoothly moves toward the newly determined configuration, and we iterate this process until the user releases the manipulated object.

IV. IDENTIFYING REGIONS OF INTEREST

We assume that the user would like to carefully observe the region where the object manipulated by him/her could be collided with other objects in the near future, because a large diversity of spatial tasks in virtual reality, such as manufacturing, modeling, and surgery, require precise alignment and collision avoidance among nearby objects. Let the manipulated object and the object nearest to it be \mathcal{P} and \mathcal{Q} , respectively. When the shortest distance between \mathcal{P} and \mathcal{Q} occurs at the point \mathbf{p} on \mathcal{P} and the point \mathbf{q} on \mathcal{Q} , we define the *region of interest* (ROI) as a spherical region that is centered at the midpoint \mathbf{c} between \mathbf{p} and \mathbf{q} , that is, $\mathbf{c} = \frac{\mathbf{p} + \mathbf{q}}{2}$, and is of a radius $r = \|\mathbf{c} - \mathbf{p}\|$. We limit the maximum radius of the ROI to a predefined value r_{max} to avoid trying to reflect

too distant objects. If the shortest distance between \mathcal{P} and \mathcal{Q} exceeds $2 \cdot r_{max}$, the ROI and the configuration of the virtual mirror is not updated at the current frame.

A naïve approach to identify the region of interest is to compute the distance between every pair of vertices (\mathbf{p} , \mathbf{q}) belonging to every pair of objects (\mathcal{P} , \mathcal{Q}) and choose the shortest one, which would require too much computation time to be handled at real-time frame rates especially with a large number of objects and complicated shapes of objects. In order to accelerate this process, we search for the approximate solutions for the ROI instead of the exact ones based on the well-known OBB trees, which have been popularly used for collision detection [12]. At a preprocessing phase, we build an OBB tree for every object by recursively subdividing the polygonal mesh of the object while fitting an oriented bounding box to each subdivided part. Figure 2 shows that tighter bounding boxes are obtained as the depth of a tree increases. In our experiments, we set the tree depth variably between 1 and 3 according to the shape complexity of each object.

For each time instant at runtime, given the manipulated object \mathcal{P} and the other objects $\{\mathcal{Q}_1, \dots, \mathcal{Q}_N\}$, we first compute the distance between the root node of \mathcal{P} 's OBB tree and the root node of \mathcal{Q}_i 's OBB tree for every $1 \leq i \leq N$, and then choose the closest object \mathcal{Q} that yields the shortest distance. In our experiments, we computed the distance between any two nodes of OBB trees approximately as the minimum distance among the 1-D distances projected to 15 separating axes obtained from their associated bounding boxes for efficiency. Once the nearest pair (\mathcal{P} , \mathcal{Q}) has been determined, we find the closest pair of the leaf nodes belonging to the OBB trees of \mathcal{P} and \mathcal{Q} by using the recursive procedure described in Algorithm 1.

V. DECIDING PLACEMENT OF MIRRORS

The image reflected by the virtual mirror is not only determined by the position and orientation of the mirror, but also by the position and orientation of the viewer. However, the latter two parameters, that is, the position and orientation of the viewer, are manually controlled by the user via the head-mounted display he/she wears. Therefore, we can affect the quality of the reflected image only through adjusting the former two parameters, that is, the position and orientation of the mirror. We determine the desired position $\hat{\mathbf{p}} \in \mathbb{R}^3$ and orientation $\hat{\mathbf{q}} \in \mathbb{S}^3$ of the virtual mirror at regular intervals such that the mirror can effectively reflect the ROI to the user.

If the mirror has been created at the current frame, its position and orientation are immediately set to $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$. Otherwise, the mirror changes its position and orientation gradually toward $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$ based on linear interpolation. Specifically, the position and orientation of the mirror at the next frame \mathbf{p}_{t+1} and \mathbf{q}_{t+1} are computed through the following equations based on its position and orientation at the current frame \mathbf{p}_t and \mathbf{q}_t :

$$\begin{aligned} \mathbf{p}_{t+1} &= \text{lerp}(\mathbf{p}_t, \hat{\mathbf{p}}, \alpha) = (1 - \alpha)\mathbf{p}_t + \alpha\hat{\mathbf{p}} \\ \mathbf{q}_{t+1} &= \text{slerp}(\mathbf{q}_t, \hat{\mathbf{q}}, \alpha) = (\hat{\mathbf{q}}\mathbf{q}_t^{-1})^\alpha \mathbf{q}_t \end{aligned} \quad (1)$$

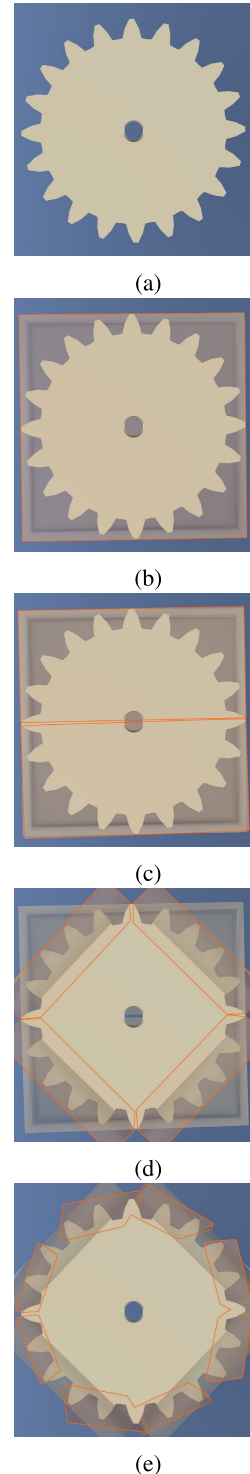


FIGURE 2. OBB tree. (a) The original object. (b) The bounding box enclosing the entire object, which corresponds to the root node of the tree. (c) Two bounding boxes (upper and lower) at the depth 1. (d) Four bounding boxes (upper left, upper right, lower left, and lower right) at the depth 2. (e) Eight bounding boxes at the depth 3, which correspond to the leaf nodes of the tree.

Given the ROI, (\mathbf{c}, r) , and the position and orientation of the viewer, \mathbf{p}_v and \mathbf{q}_v , we formulate the problem of determining the desired position and orientation of the mirror, $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$, as an optimization problem based on our

Algorithm 1 Find the Closest Pairs of Leaf Nodes

```

1: procedure FindClosest( $p, q$ )
2:    $d \leftarrow \text{DistBoxBox}(p, q)$ 
3:   if  $d \geq 2 \cdot r_{max}$  then
4:     return  $[\infty, \text{nil}, \text{nil}]$ 
5:   end if
6:   if  $p.child = \emptyset$  and  $q.child = \emptyset$  then
7:     return  $[d, p, q]$ 
8:   end if
9:   if  $p.child = \emptyset$  then
10:     $P_c \leftarrow \{p\}$ 
11:  else
12:     $P_c \leftarrow p.child$ 
13:  end if
14:  if  $q.child = \emptyset$  then
15:     $Q_c \leftarrow \{q\}$ 
16:  else
17:     $Q_c \leftarrow q.child$ 
18:  end if
19:   $d_{min} \leftarrow \infty$ 
20:   $p_{min} \leftarrow \text{nil}$ 
21:   $q_{min} \leftarrow \text{nil}$ 
22:  for all  $(p_c, q_c) \in P_c \times Q_c$  do
23:     $[d_l, p_l, q_l] \leftarrow \text{FindClosest}(p_c, q_c)$ 
24:    if  $d_l < d_{min}$  then
25:       $d_{min} \leftarrow d_l$ 
26:       $p_{min} \leftarrow p_l$ 
27:       $q_{min} \leftarrow q_l$ 
28:    end if
29:  end for
30:  return  $[d_{min}, p_{min}, q_{min}]$ 
31: end procedure

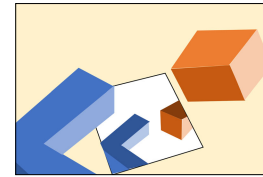
```

objective function that quantitatively measures the quality of the reflected image. For computational efficiency, we regard only the desired position $\hat{\mathbf{p}}$ as the independent variable for the optimization and determine the desired orientation $\hat{\mathbf{q}}$ subordinately to $\hat{\mathbf{p}}$. To do so, we first compute the desired normal vector $\hat{\mathbf{n}}$ of the mirror and then find the rotation between the original normal vector \mathbf{n} and the desired normal vector $\hat{\mathbf{n}}$, which can be compactly represented as a unit quaternion $(\cos \frac{\theta}{2}, \mathbf{v} \sin \frac{\theta}{2})$, where the axis \mathbf{v} is $\frac{\mathbf{n} \times \hat{\mathbf{n}}}{\|\mathbf{n} \times \hat{\mathbf{n}}\|}$ and the angle θ is $\tan^{-1}(\frac{\|\mathbf{n} \times \hat{\mathbf{n}}\|}{\mathbf{n} \cdot \hat{\mathbf{n}}})$. In order to make the virtual mirror reflect the ROI at the center of it, the desired normal vector is obtained by the following equation:

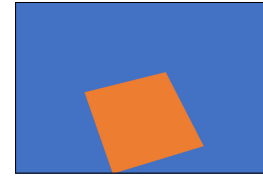
$$\hat{\mathbf{n}} = \frac{\mathbf{d}_v + \mathbf{d}_c}{\|\mathbf{d}_v + \mathbf{d}_c\|} \quad (2)$$

where \mathbf{d}_v and \mathbf{d}_c are the direction vectors from the mirror's position to the viewer's position $\frac{\mathbf{p}_v - \hat{\mathbf{p}}}{\|\mathbf{p}_v - \hat{\mathbf{p}}\|}$ and the direction vector from the mirror's position to the ROI position $\frac{\mathbf{c} - \hat{\mathbf{p}}}{\|\mathbf{c} - \hat{\mathbf{p}}\|}$, respectively.

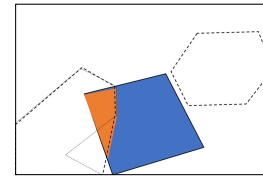
The objective function for our optimization problem is defined by a weighted sum of several evaluation functions



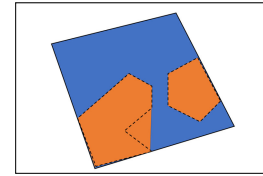
(a) Example scene



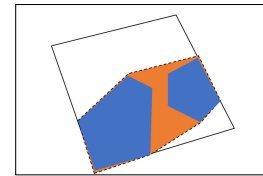
(b) Size (E_s)



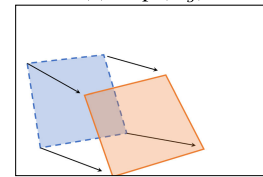
(c) Visibility (E_v)



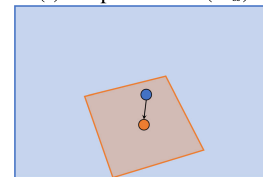
(d) Reflection (E_r)



(e) Gap (E_g)



(f) Displacement (E_d)



(g) Fovea (E_f)

FIGURE 3. Visual description of objective functions.

as follows:

$$E = \omega_s E_s + \omega_v E_v + \omega_r E_r + \omega_g E_g + \omega_d E_d + \omega_f E_f \quad (3)$$

Each evaluation function E_x scores the quality of the reflected image based on its own unique criterion and the weight ω_x balances its contribution to the total score against other functions, as visually depicted in Figure 3.

(In our experiments, we set $\omega_s, \omega_v, \omega_r, \omega_g, \omega_d$, and ω_f to 0.1, 0.4, 0.3, 0.3, 0.2, and 0.1, respectively.) We explain how each evaluation function is defined in detail as follows.

- **Size (E_s):** This function measures how large area the mirror occupies in the user's field of view (see Figure 3 (b)). Given the size of the entire view S_{view} and the size of the mirror S_m in the image space, this function is defined as follows.

$$E_s = \frac{A(\mathcal{P}_m)}{S_{view}} \quad (4)$$

The size of the mirror S_m in the image space can be calculated by projecting the four corners of the mirror onto the view plane, clipping the projected quadrilateral against the view frustum, and measuring the size of the clipped polygon $\mathcal{P}_m = [(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})]$, which can be obtained by the following equation:

$$A(\mathcal{P}) = \frac{1}{2} \left| \sum_{i=0}^{n-1} x_i y_{(i+1)\%n} - \sum_{i=0}^{n-1} x_{(i+1)\%n} y_i \right| \quad (5)$$

where the $\%$ symbol represents the modulo operator in computing.

- **Visibility (E_v):** This function measures how large area in the mirror is visible from the user (see Figure 3 (c)). In order to calculate the area occluded by the objects associated with ROI (the manipulated object and the object nearest to it), which we will call *ROI objects* from now on, we first project every vertex of each object onto the view plane. For computational efficiency, we obtain the convex hull of the projected vertices of each object instead of extracting the precise silhouette. We find the intersection \mathcal{I}_i between each convex hull \mathcal{O}_i and the polygon of the mirror \mathcal{P} to take only the area inside the mirror into account. Because the intersections \mathcal{I}_1 and \mathcal{I}_2 can be overlapped with each other, we compute the entire size of the obstructed area by adding the size of each intersection and then subtracting the size of the overlapped area from it. Finally, the visibility of the mirror is computed as follows:

$$E_v = 1 - \frac{A(\mathcal{I}_1) + A(\mathcal{I}_2) - A(\mathcal{I}_1 \cap \mathcal{I}_2)}{A(\mathcal{P}_m)} \quad (6)$$

- **Reflection (E_r):** For the main purpose of the virtual mirror, this function evaluates how largely the ROI objects are reflected by the mirror (see Figure 3 (d)). In order to obtain the reflected image of each object in the mirror approximately, we first reflect the object about the mirror in the 3D space, and find the intersection between the ray from the viewer toward each vertex of the reflected object and the plane including the mirror, and create the convex hull of the intersected points, and finally clip the convex hull against the mirror (see Figure 4). Given the clipped polygon of each object in the mirror as \mathcal{P}'_i , we evaluate the ratio of the reflected image to the size of the mirror in a similar

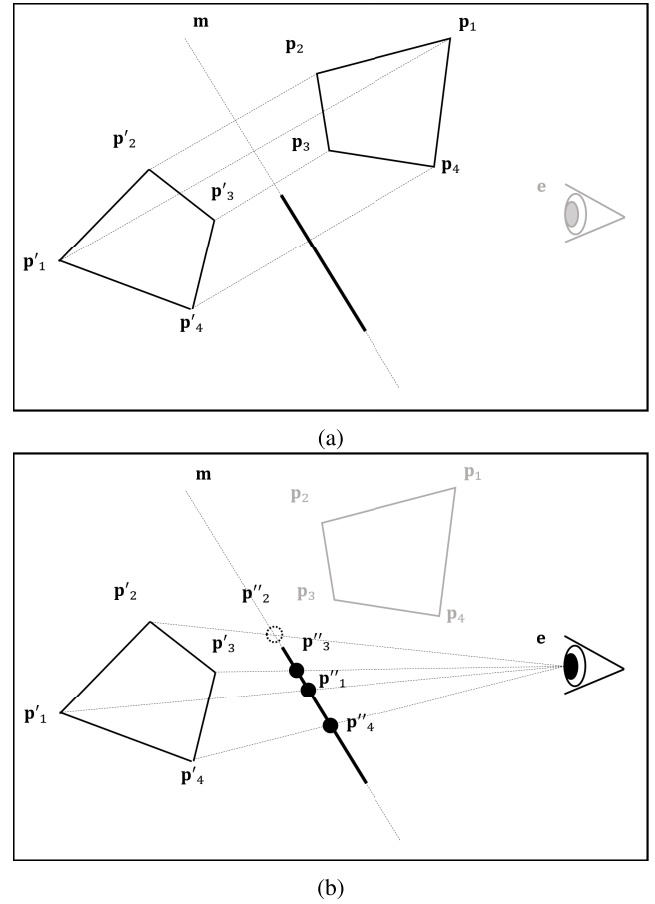


FIGURE 4. Calculating the mirror reflection of the vertices belonging to the given object $\{p_1, \dots, p_n\}$. (a) Reflect the given object across the plane including the mirror to generate the vertices $\{p'_1, \dots, p'_n\}$. (b) Find the intersections between the rays from the viewer toward the reflected vertices and the plane including the mirror to generate the vertices $\{p''_1, \dots, p''_n\}$.

way to E_v as follows:

$$E_r = \frac{A(\mathcal{P}'_1) + A(\mathcal{P}'_2) - A(\mathcal{P}'_1 \cap \mathcal{P}'_2)}{A(\mathcal{P}_m)} \quad (7)$$

- **Gap (E_g):** Not only the reflected images of the ROI objects but also the empty space between those contribute significantly to the visual clarity of the spatial relationship between the objects. If there is no empty space between the reflected images in the mirror, it would be hard for the user to judge whether the ROI objects are distant from each other or not in the 3D space. Given the convex hull \mathcal{P}' that encloses the reflected images of both objects \mathcal{P}'_1 and \mathcal{P}'_2 , we regard the gap between the reflected images as the area corresponding to $(\mathcal{P}' - \mathcal{P}'_1 \cup \mathcal{P}'_2)$. The function E_g evaluates the ratio of the size of this gap to the size of the surrounding convex hull as follows (see Figure 3 (e)):

$$E_g = 1 - \frac{A(\mathcal{P}'_1) + A(\mathcal{P}'_2) - A(\mathcal{P}'_1 \cap \mathcal{P}'_2)}{A(\mathcal{P}')} \quad (8)$$

- **Displacement (E_d):** In order to avoid rapid movement of the virtual mirror along successive frames, this function

measures the ratio of the average displacement of the four corners of the mirror in the image space to the maximum length of the viewport (L_{view}), and then subtracts the ratio from one as follows (see Figure 3 (f)):

$$E_d = 1 - \frac{1}{4} \sum_{i=1}^4 \frac{\|\mathbf{p}_i^t - \mathbf{p}_i^{t-1}\|}{L_{view}} \quad (9)$$

where \mathbf{p}_i^t and \mathbf{p}_i^{t-1} correspond to the coordinates of the i -th corner of the mirror at the current and the previous frame, respectively.

- **Fovea (E_f):** The researchers in cognitive science have found that the capability of foveal vision is generally superior to the capability of peripheral vision in the system of human visual perception, particularly in terms of the visual acuity as well as the vulnerability to clutter [13]. We also observed that the user in our experiments paid more attention to the virtual mirror when it was close to the center of the view than when it was far from the center. This function measures how close the mirror is from the center of the view \mathbf{c}_{view} as follows (see Figure 3 (g)):

$$E_f = 1 - \frac{1}{L_{view}} \left\| \mathbf{c}_{view} - \frac{1}{4} \sum_{i=1}^4 \mathbf{p}_i \right\| \quad (10)$$

Note that the above functions are defined with respect to a monoscopic image from a single viewpoint. For stereoscopic displays found in common virtual reality systems today, these functions can either be evaluated for both stereoscopic pairs of images from the left and right viewpoints and then averaged, or be evaluated for a single monoscopic image from a virtual, intermediate viewpoint. In our experiments, we chose the latter option for simplicity.

Given the objective function E , we search for the optimal position of the mirror $\hat{\mathbf{p}}$ such that the image rendered with the mirror at that position gives us the maximum score from E . We constrain the distance between the mirror and the center of the ROI to a real multiple of the ROI radius, αr , which consequentially limit the candidate positions of the mirror to the positions over the sphere around the center of the ROI. In addition, we prune the half of the sphere that is closer to the viewing position, because the mirrors located at the positions over the near hemisphere would not face toward the viewer. For practical optimization, we regularly sample a set of positions over the far hemisphere, as shown in the Figure 5, evaluate E for every position in the set, and finally choose the optimal position that maximizes E .

$$\hat{\mathbf{p}} = \operatorname{argmax}_{\mathbf{p} \in \mathcal{HS}} E \quad (11)$$

For a large set \mathcal{HS} , it might take more than a few tens of milliseconds to evaluate E for all of the sampled positions, which would prohibit the optimization process to be performed at real-time frame rates. In our experiments, we evaluated E for the entire set over an interval of 7 consecutive frames and updated the desired position and orientation of the

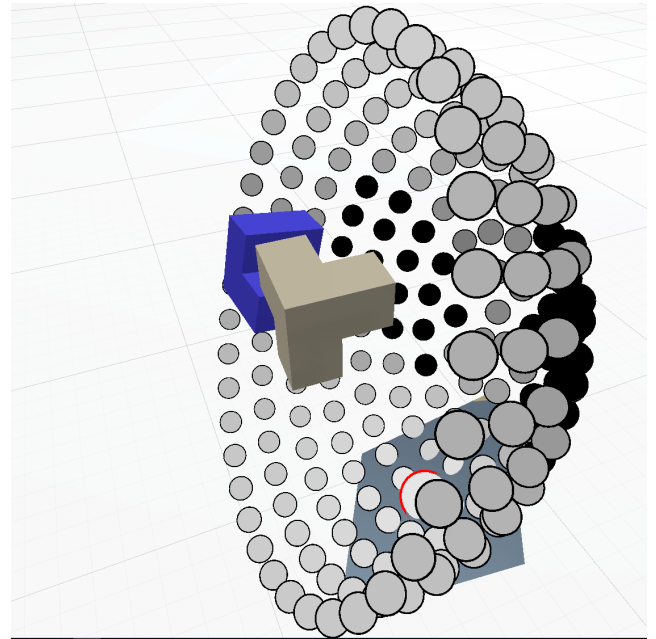


FIGURE 5. A set of candidate positions of the virtual mirror regularly sampled on the hemisphere around the ROI objects. Each position is drawn as a sphere, which is colored according to the score evaluated at its position. The next position of the mirror is determined as the position where the best score is obtained, depicted by the red boundary of a sphere.

virtual mirror at that interval to keep our applications running at real-time frame rates.

VI. EXPERIMENTAL RESULTS

We implemented an interactive 3D manipulation system in virtual reality using the Unity game engine. Unity VR SDKs allowed us to easily integrate any of the currently popular VR platforms into our system, and we mainly used the HTC Vive headset for our experiments. For implementing mirror effects in virtual reality, we used the Vive Stereo Rendering Toolkit obtained from the Unity Asset Store. The user in our system was immersed in a virtual world consisting of a large number of objects, which were basically stationary at the initially arranged locations and were kinematically controllable via hand-held motion controllers. Any of the objects could be grabbed and released by pressing and releasing a trigger button on one of the controllers. The object grabbed by the user could be directly relocated and reoriented by translating and rotating its associated controller. The object released by the user kept stationary at the final location and orientation that had been lastly updated before being released.

On top of this basic framework for interactive manipulation in virtual reality, we implemented three distinctive applications requiring precise 3D manipulation to demonstrate the effectiveness of our virtual mirrors under various usage scenarios as follows.

- **Assembling construction toys:** This first application provided the user with a collection of building blocks which were designed based on Lego technic construction kit. The user could assemble those blocks together

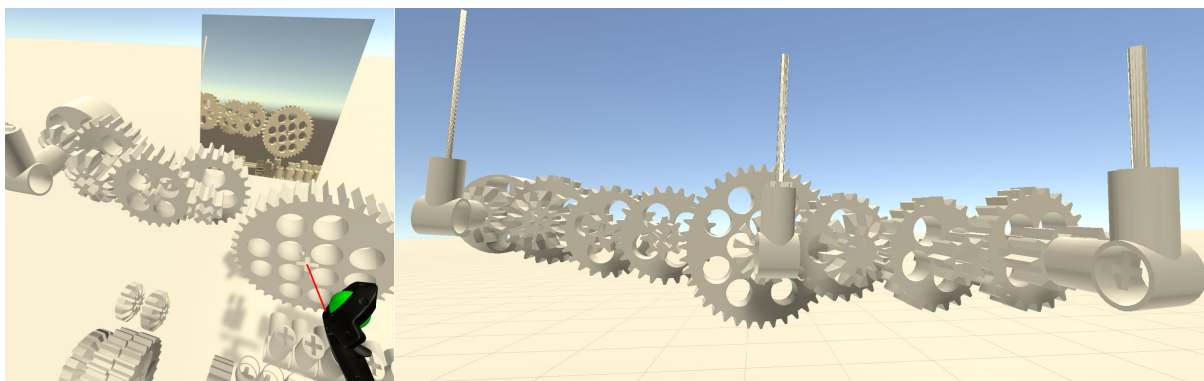


FIGURE 6. Assembling construction toys. (Left) A screenshot capturing the process of assembling a mechanical structure in virtual reality with the support of the virtual mirror. (Right) The completely assembled structure, which behaves like a mechanical clock.

to create a diversity of mechanical constructions such as mechanical doors and clocks (see Figure 6). Our system supported the so-called ‘snap-alignment’ interface by which the block manipulated by the user was automatically translated and rotated to fit exactly into a nearby block if those two blocks were aligned with each other approximately within a predefined threshold. Even with such an assistive interface, the task of putting blocks together in virtual reality was often challenging, particularly when the user had to satisfy two or more spatial constraints at the same time. In our preliminary user study with the virtual mirror, we observed that the participants could assemble complicated mechanical constructions which had a lot of interlinked spatial constraints in relatively easy ways, because the virtual mirror allowed the participants to quickly recognize the spatial relationships among blocks from various perspectives simultaneously.

- Solving 3D dissection puzzles:** In this second application, the user was challenged with the Some cube puzzle, which was a kind of solid dissection puzzle requiring the user to assemble seven unique pieces into a $3 \times 3 \times 3$ cube [14]. Each piece was made by connecting three or four unit cubes such that each adjacent pair of cubes joined at their faces. One typical strategy for solving this kind of puzzle is to incrementally extend and reduce a partially assembled structure by putting a piece at a time and occasionally backtracking to the previous structure (see Figures 1 and 7). Putting each new piece requires the user be able to enumerate every possible arrangement of the piece with which the piece can be tightly fitted to the existing structure without intersection, which is usually accompanied by rotating either the structure itself or the viewing direction instead to reveal the hidden parts. In our preliminary user study, we observed that the virtual mirror disclosed the hidden parts effectively, so that the participants could solve the given puzzles with less efforts than without the mirror.
- Reconstructing 3D shapes:** This last application was similar to the second application in that the user needed to fit pieces together into a target shape. However, the

pieces were not manually designed, but rather were automatically generated by simulating the shattering process of target shapes. We used the Stanford bunny model and the Utah teapot model for target shapes, and ran the solid shatter effect of Autodesk Maya to both models to generate 10 pieces for each model (see Figure 8). Interlocking any two pieces precisely was highly challenging because of their irregular and uneven surface features, and usually required iterative adjustment of their positions and orientations from various perspectives. Our preliminary user study showed that the virtual mirror could clearly elevate the precision of fitting pieces due to its provision of additional visual information from a diversity of viewing positions and directions.

In order to demonstrate the usefulness of our approach for precise 3D manipulation in virtual reality in a quantitative manner, we conducted a user study based on the last application above in which the user reconstructed a target shape by fitting a collection of shattered parts together within a time limit. A total of 10 undergraduate students (9 males and 1 female) participated in this user study. All of the participants took part in the test based on the Utah teapot model, and only seven of them participated in the test based on the Stanford bunny model additionally. Each test consisted of two sessions; one session without the virtual mirror, and the other session with the virtual mirror. The two sessions were presented in a counterbalanced order to avoid sequence biases, because we were worried that the participants could learn from their earlier sessions and exhibit better performances in their later sessions.

We provided every participant with the same sequence of 10 consecutive solid dissection puzzles for both sessions in a test. For each puzzle, a participant was asked to fit just one of the randomly selected piece into the target model (either the Stanford bunny or the Utah teapot) within a maximum duration of one minute. To solve the puzzle, the participant first needed to find a hole in the model, where the given piece had been taken out of the model, and then to precisely align the piece with the shape of the hole such that the gap between the piece and the hole could be minimized. When the participant reported the completion of the given task by pressing the

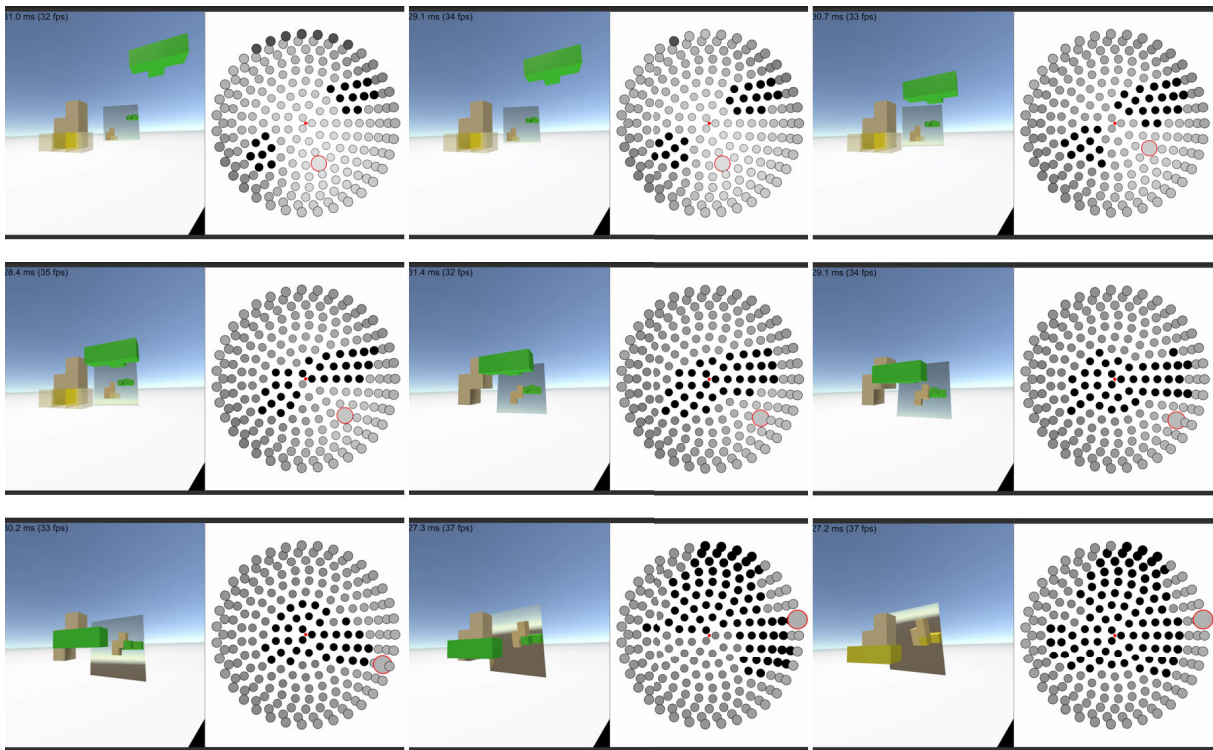


FIGURE 7. A sequence of fitting the green piece to the stationary yellow piece, from the upper left to the lower right.

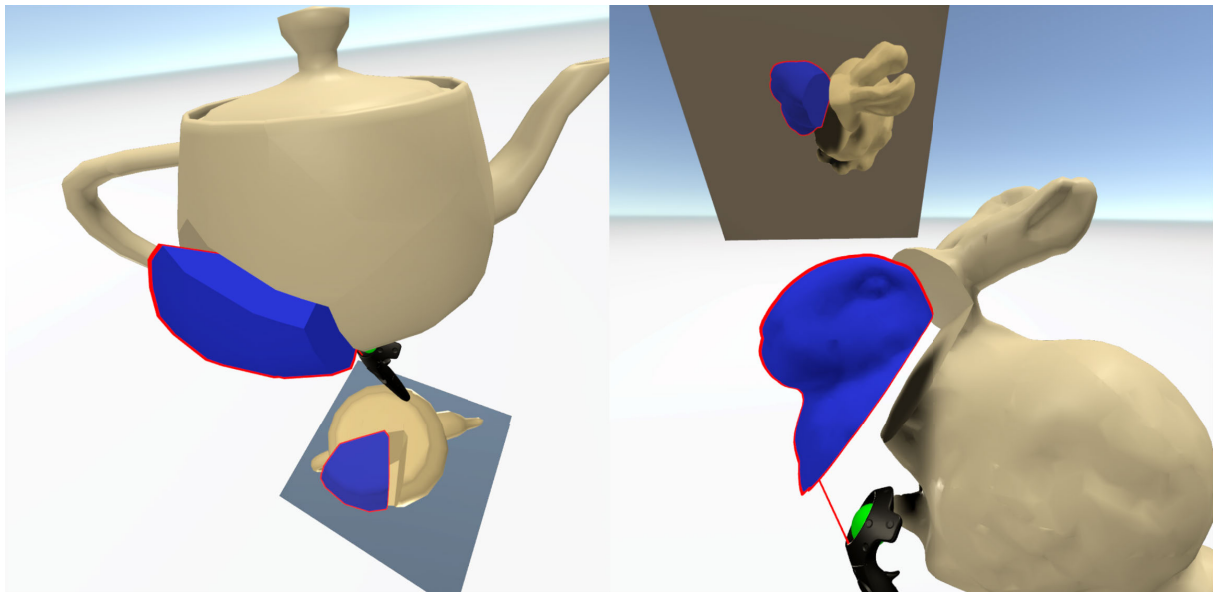


FIGURE 8. Reconstructing 3D shapes. (Left) Utah teapot model. (Right) Stanford bunny model.

grip button of a motion controller, the system measured the error by averaging over the distance between each pair of the corresponding vertices of the piece aligned by the participant and the piece originally fitted in the model. When the maximum allowed duration passed before the participant’s reporting of the completion, the system measured the error with respect to the lastly released position and orientation of the puzzle piece instead of just the last position and orientation. In addition to the spatial error, we also measured the

completion time and the movement distances of the head-mounted display.

Figure 9 summarizes the results from the quantitative test. For both models, the errors of the spatial alignment decreased when using our virtual mirrors, as shown in the Figure 9 (a). On the other hand, it took a bit longer time on average to complete the given task with our virtual mirrors, as shown in the Figure 9 (b). Such an increase of the time to completion looked somewhat puzzling at first, but soon became clarified

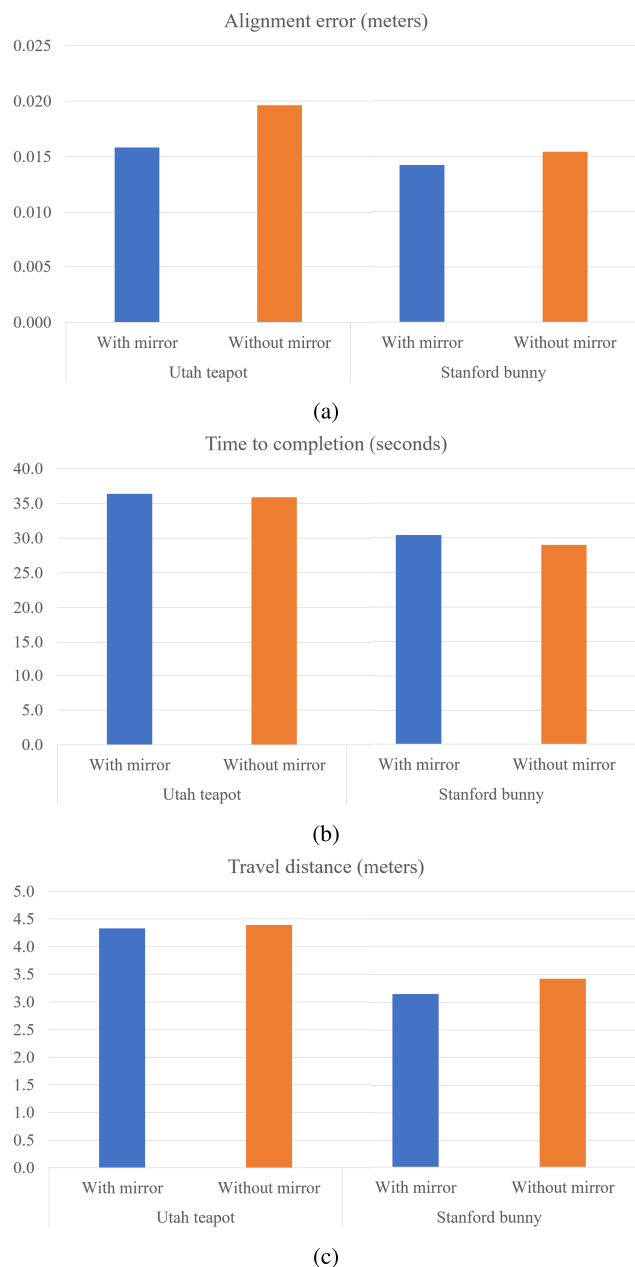


FIGURE 9. The results from our quantitative user study.

as a natural result from the increased cognitive load for interpreting the additional visual information given by our virtual mirrors. In spite of such longer time intervals for task completion, the total travel distances, which were measured by summing over the displacements of the head-mounted display, were not lengthened but rather shortened on average, particularly in the experiment with the Stanford bunny model. Combining these results led to a conclusion that our virtual mirrors facilitated precise 3D manipulation in virtual reality by providing additional visual information, which usually increased the time to interpret the given scene but decreased the body movement for observing the scene from a diversity of viewpoints.

After each test, we asked the participant to fill in a questionnaire consisting of the following 6 questions:

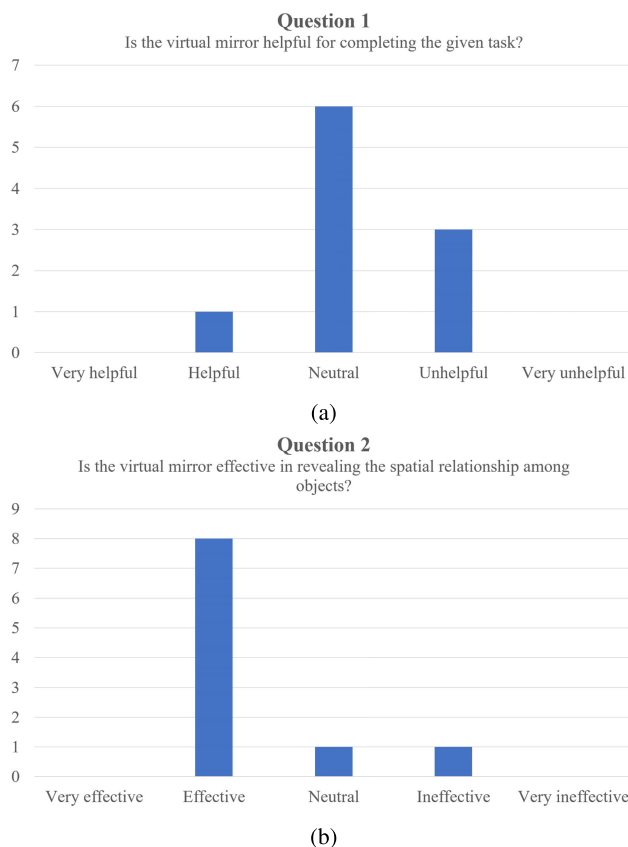


FIGURE 10. The responses to the survey questions 1 and 2.

- 1) Is the virtual mirror helpful for completing the given task? (1: Very helpful, 2: Helpful, 3: Neutral, 4: Unhelpful, 5: Very unhelpful)
- 2) Is the virtual mirror effective in revealing the spatial relationship among objects? (1: Very effective, 2: Effective, 3: Neutral, 4: Ineffective, 5: Very ineffective)
- 3) Which task is the most appropriate for the virtual mirror? (1: Utah teapot, 2: Stanford bunny)
- 4) Which interface is more convenient to use? (1: With the virtual mirror, 2: Without the virtual mirror, 3: Same)
- 5) Describe the advantages of the virtual mirror freely.
- 6) Describe the weaknesses of the virtual mirror freely.

Figure 10 shows the responses to the questions 1) and 2). Whereas the responses to the question 1) were somewhat negative, the responses to the question 2) were clearly in favor of the virtual mirrors. We guess that the negative responses to the question 1) are due to the increased time to task completion, and that the positive responses to the question 2) are due to the decreased error, or increased precision, in aligning pieces. For the question 3), Utah teapot and Stanford bunny were evenly selected by the participants. Four of the participants selected ‘With the virtual mirror’, and the remaining six participants chose ‘Same’ for the question 4). Most of the participants described the advantage of the virtual mirror as ‘being able to observe the given object from behind without moving or rotating one’s own body’. The disadvantages

described by the participants ranged from ‘no disadvantage’ to ‘obstructing the main object’, ‘not controllable at will’, and so on. We will briefly discuss about these limitations in the Section VII.

VII. CONCLUSION

We presented a new method for precise 3D manipulation in virtual reality by embedding and controlling a virtual mirror so that the visual information about the spatial relationships among multiple objects could be augmented in a natural way. An algorithm based on the OBB trees, precomputed for each object, efficiently identifies the region of interest (ROI) where the gap between the object manipulated by the user and the object closest to it is minimized. Once the ROI has been identified, the position and orientation of the virtual mirror are determined based on an optimization process in which the objective function measures how the spatial relationship between the ROI objects can be effectively captured by the mirror and reflected to the user. Experimenting with three different practical applications demonstrated the usefulness of our method under various usage scenarios requiring high precision of 3D manipulation in virtual reality. The quantitative results from our user study confirmed that the user could complete the spatial alignment tasks in virtual reality very precisely with the virtual mirror.

The key limitation of our approach is that the automatic behavior of the virtual mirror sometimes can be regarded as unintelligible and even annoying because of its insufficient understanding of the entire scene’s organization and the user’s intention, particularly when the object manipulated by the user moves in a crowded region where a number of objects are densely clustered. Because our objective function takes only the two objects involved in the ROI into account, the optimal position and orientation of the virtual mirror can result in somewhat unnatural situations including the obstruction of and/or the collision with the other objects in the scene. Also, the ROI determined by our algorithm can be significantly different from the real ROI in which the user is actually interested, because the algorithm does not consider any domain-specific features besides the distances among objects. One possible approach to partially address this limitation is to make the virtual mirror a physically simulated object based on the rigid-body dynamics and to apply repulsive forces to the mirror so that it can smoothly avoid collision with any objects in the scene. Taking even the user’s intention into account would be a more challenging research problem, which might be tackled by using the recent machine learning techniques, including the deep reinforcement learning method.

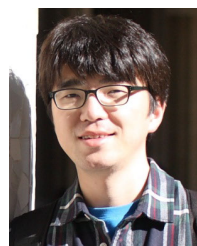
REFERENCES

- [1] D. Mendes, F. M. Caputo, A. Giachetti, A. Ferreira, and J. Jorge, “A survey on 3D virtual object manipulation: From the desktop to immersive virtual environments,” *Comput. Graph. Forum*, vol. 38, no. 1, pp. 21–45, Feb. 2019.
- [2] G. M. Nielson and D. R. Olsen, “Direct manipulation techniques for 3D objects using 2D locator devices,” in *Proc. Workshop Interact. 3D Graph. SI3D*, 1987, pp. 175–182.

- [3] B. D. Conner, S. S. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik, and A. van Dam, “Three-dimensional widgets,” in *Proc. Symp. Interact. 3D Graph. SI3D*, 1992, pp. 183–188.
- [4] A. Martinet, G. Casiez, and L. Grisoni, “The effect of DOF separation in 3D manipulation tasks with multi-touch displays,” in *Proc. 17th ACM Symp. Virtual Reality Softw. Technol. VRST*, 2010, pp. 111–118.
- [5] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev, *3D User Interfaces: Theory and Practice*. Reading, MA, USA: Addison-Wesley, 2004.
- [6] I. Poupyrev, M. Billinghamurst, S. Weghorst, and T. Ichikawa, “The go-go interaction technique: Non-linear mapping for direct manipulation in VR,” in *Proc. 9th Annu. ACM Symp. User Interface Softw. Technol. UIST*, 1996, pp. 79–80.
- [7] D. A. Bowman and L. F. Hodges, “An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments,” in *Proc. Symp. Interact. 3D Graph. SI3D*, 1997, p. 35.
- [8] R. Stoakley, M. J. Conway, and R. Pausch, “Virtual reality on a WIM: Interactive worlds in miniature,” in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. CHI*, 1995, pp. 265–272.
- [9] J. S. Pierce, B. C. Stearns, and R. Pausch, “Voodoo dolls: Seamless interaction at multiple scales in virtual environments,” in *Proc. Symp. Interact. 3D Graph. SI3D*, 1999, pp. 141–145.
- [10] S. Frees and G. D. Kessler, “Precise and rapid interaction through scaled 3D manipulation,” in *Proc. IEEE VR*, Mar. 2005, pp. 99–106.
- [11] M. G. Choi, J. H. Lee, W. Ha, and K. H. Lee, “Optimal close-up views for precise 3D manipulation,” *Comput. Animation Virtual Worlds*, vol. 30, nos. 3–4, p. 1884, May 2019.
- [12] S. Gottschalk, M. C. Lin, and D. Manocha, “OBBTree: A hierarchical structure for rapid interference detection,” in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn. SIGGRAPH*, 1996, pp. 171–180.
- [13] R. Rosenholtz, “Capabilities and limitations of peripheral vision,” *Annu. Rev. Vis. Sci.*, vol. 2, no. 1, pp. 437–457, Oct. 2016.
- [14] M. Gardner, *The 2nd Scientific American Book of Mathematical Puzzles and Diversions*. New York, NY, USA: Simon & Schuster, 1961.



WANSU HA was born in Seoul, South Korea, in 1991. He received the B.S. and M.S. degrees in computer science from Kwangwoon University, in 2017 and 2020, respectively. His research interests include computer animation, human–computer interaction, and virtual reality.



MYUNG GEOL CHOI was born in Daegu, South Korea, in 1980. He received the Ph.D. degree in computer science from Seoul National University, in 2010. He is currently an Associate Professor with the Department of Media Technology and Contents Technology, The Catholic University of Korea. He is a member of board of directors of the Korea Computer Graphics Society. His research interests include character animation, interactive animation authoring, and virtual reality.



KANG HOON LEE was born in Seoul, South Korea, in 1976. He received the B.S., M.S., and Ph.D. degrees in computer engineering from Seoul National University, in 2000, 2002, and 2007, respectively. Since 2008, he has been a Professor with the Department of Computer Science, Kwangwoon University. From 2014 to 2015, he has been a Visiting Researcher with the University of California at Los Angeles, Los Angeles. He is a member of board of directors of

Korea Computer Graphics Society. His research interests include computer animation, human–computer interaction, and virtual reality.

...