applied sciences

MDPI

Article

Virtual Object Manipulation by Combining Touch and Head Interactions for Mobile Augmented Reality

Ju Young Oh [1,2], Ji Hyung Park [2] and Jung-Min Park [1,2,*]

[1] NT-IT (HCI & Robotics), Korea University of Science and Technology, Gajeong-ro, Yuseong-gu, Daejeon 34113, Korea

[2] Center for Intelligent & Interactive Robotics, Korea Institute of Science and Technology, 5, Hwarang-ro 14-gil, Seongbuk-gu, Seoul 02792, Korea

* Correspondence: pjm@kist.re.kr

Received: 21 June 2019; Accepted: 19 July 2019; Published: 22 July 2019

Abstract: This paper proposes an interaction method to conveniently manipulate a virtual object by combining touch interaction and head movements for a head-mounted display (HMD), which provides mobile augmented reality (AR). A user can conveniently manipulate a virtual object with touch interaction recognized from the inertial measurement unit (IMU) attached to the index finger's nail and head movements tracked by the IMU embedded in the HMD. We design two interactions that combine touch and head movements, to manipulate a virtual object on a mobile HMD. Each designed interaction method manipulates virtual objects by controlling ray casting and adjusting widgets. To evaluate the usability of the designed interaction methods, a user evaluation is performed in comparison with the hand interaction using Hololens. As a result, the designed interaction method receives positive feedback that virtual objects can be manipulated easily in a mobile AR environment.

Keywords: touch interaction; head movements; indirect manipulation; augmented reality; mobile HMD

1. Introduction

Manipulating a virtual object is a fundamental and essential interaction task for a user experiencing augmented reality (AR). To achieve this, selecting, moving, rotating, and changing the size of virtual objects are basic tasks [1]. At least, the selection, movement, and rotation of an (virtual) object, which are used to manipulate a real object in everyday life, should be possible in an AR environment.

Currently, owing to the development of computer hardware, several commercial head-mounted displays (HMDs) [2–5] are available, which a user can use to experience AR in a mobile environment. Providing room modeling using the simultaneous localization and mapping (SLAM) algorithm, some HMDs [3,4] allow a user to interact with virtual contents [6]. Especially, a virtual interior task [7,8], which adds virtual furniture in a real room, requires an interaction method for repeatedly and conveniently manipulating the remote virtual object in the AR environment.

In such a scenario, there is a limitation in manipulating a virtual object with an existing interface in a mobile AR environment. The touch interaction achieved using the touchpad attached to an HMD [2] provides the user with familiar touch interaction and is advantageous as the user does not need to carry the input device. However, since most of the HMD touchpads are attached to the side of the HMD, the difference between the touch input space and manipulating space makes it difficult and inconvenient for a user to manipulate a virtual object [9]. In contrast, through the hand interaction achieved with a built-in camera in an HMD, the virtual object can be manipulated naturally while the user manipulates a real object. However, because most cameras are attached in front of the HMD, the user must raise his/her hands to interact with a virtual object. When the user repeatedly conducts the task of manipulating the virtual object, this interaction increases the fatigue of the user's arm.

The interaction performed using a wand or a controller for manipulating a virtual object [5] has a wider recognition range than the hand interaction and various interactions can be performed with the different interfaces built into the wand. However, this is inconvenient to be used in a mobile environment because the user has to carry additional input devices such as wands and controllers [10]. According to a study on the manipulation of virtual objects [11], manipulating a virtual object with full degree of freedom (DOF) separation, rather than 6DOF, allows for a more precise control of a virtual object. In addition, using multiple input modalities produces less fatigue than that produced using a single input modality [12,13].

This paper proposes two interaction methods for conveniently manipulating a virtual object by combining touch interaction and head movements on a mobile HMD for experiencing AR as shown in Figure 1. As the proposed interaction methods consist of two input modalities (touch interaction and head movements), instead of a single input modality, they cause less fatigue when performing a task that iteratively manipulates a virtual object. Moreover, the proposed interaction methods can conveniently manipulate a virtual object because the proposed interaction methods consist of touch interaction, which is widely used in a smartphone or laptop in daily life, and thus, is familiar to a user, and the head movement, which is a frequently used interaction when looking around in everyday life.

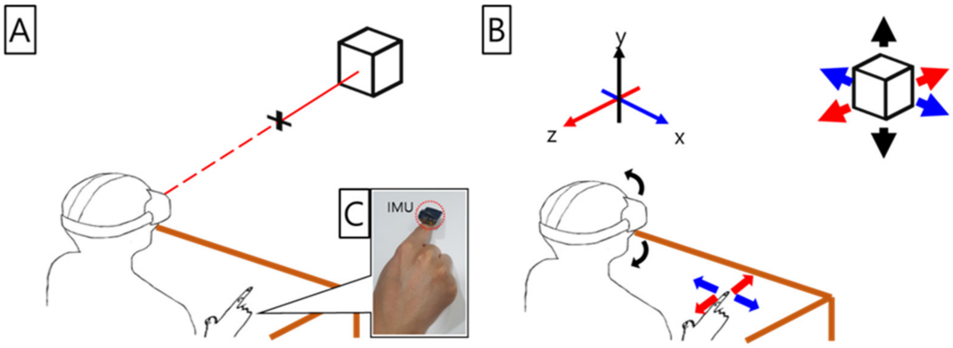


Figure 1. Conceptual diagram of manipulating a virtual object: An interaction that combines touch interaction and head movements in a mobile augmented reality (AR) environment on a head-mounted display (HMD). (**A**) The user can select the virtual object with ray casting using touch interaction and head motion. (**B**) The user can manipulate the selected virtual object using touch interaction and head motion. (**C**) A picture of the index finger on a nail-mounted internal measurement unit (IMU) sensor.

In this study, we use AnywhereTouch [14] to recognize touch interaction. AnywhereTouch is a method for recognizing the touch interaction with a nail-mounted inertial measurement unit sensor in a mobile environment regardless of the material or slope of the plane [14]. Since AnywhereTouch can recognize the touch interaction regardless of the slope of the plane, it can solve the problems when manipulating a virtual object with a touchpad attached to the HMD. Moreover, it is suitable for touch interaction in a mobile environment because it uses only a sensor, attached to the nail, as an input device.

The user's head movements (especially head rotation) can be tracked from the sensors built in most AR HMDs that can be used in a mobile environment. The head movement is a popular interaction method in commercialized HMDs [3,4] for selecting a virtual object and is faster and more efficient than the hand interaction (mid-air hand gesture) [15].

We design two interaction methods to manipulate a virtual object, which combine touch interaction and head movements with ray casting [16] and Wireframe Widget [17]. Ray casting is one of the fastest and easiest methods for selecting a virtual object [18]. Here, the user points at objects with a virtual ray emanating from a virtual hand, and then, the objects intersecting with the virtual ray can be selected and manipulated. Wireframe Widget is a widget for manipulating a 3D virtual object in Microsoft Hololens [4], which is a commercial HMD-supporting mobile AR. It can manipulate the position, rotation, and scale of a virtual object separately with widget objects. A user prefers that the position of the 3D virtual object can be manipulated with 3DOF, its angle can be rotated with 1DOF separation, and its scale can be uniformly manipulated (or adjusted) [19,20]. Wireframe Widget allows user-preferred interaction when manipulating virtual objects.

When manipulating the position of a virtual object, separation DOF can be manipulated more precisely than 3DOF [11], and when a user uses a 2DOF controller, he/she prefers to manipulate the 3D position of the virtual object by separating the 3D position into the position of a plane (2DOF) and that of an axis (1DOF). Given this background, in this study, we design two interactions to manipulate the 3D position: One input modality manipulating the plane position (2DOF) and the other manipulating the axis position (1DOF). This paper has the following contributions:

1. We design two interaction methods that combine touch interaction and head movements, which can conveniently manipulate a virtual object in a mobile AR environment.
2. We confirm that the proposed interaction methods, which are compared with mid-air hand interaction, can conveniently perform the task of repeatedly manipulating a virtual object.
3. Through user evaluation, we confirm that the interaction methods using head movements do not cause serious dizziness.

The remainder of this paper is organized as follows. The following section briefly introduces the related work. Section 3 explains the two interactions designed by combining touch interaction and head movements and explains how to manipulate a virtual object with ray casting and the Wireframe Widget by using the designed interactions. To evaluate the proposed interaction methods, in Section 4, we describe user evaluation of manipulating a virtual object. Section 5 discusses the results of the experiment and presents the conclusions.

2. Related Work

2.1. *Virtual Object Manipulation Method Using a Mouse*

It is difficult to manipulate a 3D virtual object directly with a mouse, which is mainly used for interaction on a computer. To overcome the mapping of an input space, such as a mouse, to a manipulation space in 3D with the mouse, 3D object manipulation using widgets has been studied [21–23].

Houde proposed a widget to move and rotate a virtual object by using a bounding box that has four handles surrounding the selected virtual object [22]. Conner et al. proposed a mouse interaction method that can move, rotate, and scale a virtual object with a widget that has small spheres attached at the end of the six axes [21]. Shoemake proposed Acrball, which is a sphere widget for wrapping a virtual object, for rotation manipulation of virtual objects [23]. When a user draws an arc over the sphere widget with the mouse, Acrball calculates the rotation quaternion by determining the rotation axis and rotation angle. Three-dimensional modeling programs such as Unity3d [24], Maya [25], and Blender [26] use input techniques that can manipulate a virtual object with a mouse, based on the above technique [27].

2.2. *Virtual Object Manipulation Method with Touch Interaction*

This interaction method directly manipulates the virtual contents on the touchscreen with touch interaction. tBox can control 9DOFs of a virtual object with the Wireframe Cube Widget [28]. When a user drags the edge on the Wireframe Widget, the virtual object is moved to that axis, and when the user drags the face of the Wireframe Widget, the virtual object is rotated. LTouchIt [29] can be used for direct manipulation for translations based on the rotation of a camera, and rotate a virtual object using a widget. Au et al. proposed an input method to manipulate a virtual object by relying on the standard transformation widgets, which comprise six axes, for multi-touch gestures [30].

Manipulating a virtual object with touch interaction on the touchscreen can be handy, depending on the scenario. However, it can be inconvenient for a user wearing an HMD in a mobile environment to perform a touch interaction while holding the touchpad. Moreover, this interaction makes it difficult to freely interact with a virtual object regardless of the user's field of view (FOV), due to the limited input space for recognizing the touch interaction.

Touch interaction for manipulating a virtual object on a mobile HMD has been studied previously. FaceTouch is an input technique proposed to manipulate a virtual object with a touchpad attached to the front of a virtual-reality HMD [9]. As the touch interaction has a 2D input space, FaceTouch uses a widget to manipulate a virtual object. Moreover, since the touch input device is attached to the HMD, it has high mobility and allows a user to conveniently manipulate a virtual object in his/her FOV. Despite these advantages, it is difficult to use the touchpad as an input device in an HMD-supporting mobile AR because it is attached to the front of the HMD. To recognize a touch interaction in an HMD that can experience mobile AR, Xio et al. proposed MRTouch, which can recognize touch interaction on a plane by using a camera built in the HMD and manipulate the virtual object [31]. In addition, MRTouch uses a widget to manipulate 3D virtual objects. However, MRTouch has a limitation that the finger for recognizing the touch interaction and the virtual object to be manipulated can only be recognized on the same plane.

2.3. Virtual Object Manipulation Method with Hand Gestures

Owing to the technology of computer vision, a user can interact with a virtual object like a real object by using realistic hand postures recognized from a computer vision [32]. However, it is difficult to precisely manipulate the virtual object due to the self-occlusion of the hand or limitation of human accuracy. Moreover, because the camera for recognizing the hand gesture is attached to the front of the HMD, the user has to raise his/her hand to interact with the virtual content. Manipulating virtual objects with these interactions for a long time causes fatigue to a user.

Frees et al. proposed precise and rapid interaction through scaled manipulation (PRISM) to manipulate virtual objects accurately with hand motions [33]. PRISM is an interaction method for manipulating a virtual object by adjusting the ratio to manipulating a virtual object based on the user's hand speed. Mendes et al. proposed an interaction method to manipulate a virtual object by separating the DOF with a virtual widget [11]. Although this method can manipulate virtual objects more precisely than PRISM, the task completion time increases with the number of DOFs to be manipulated.

GoGo, which can manipulate a virtual object with a hand whose position is non-linearly adjusted, has been proposed to manipulate distant virtual objects [34,35]. According to [15], the ray casting method using head motion is more efficient than that using hand motion.

2.4. Virtual Object Manipulation Method with More Than Two Input Modalities

Nancel et al. proposed interaction methods to precisely manipulate a virtual object displayed on a wall with head motion and touch interaction using a touchpad [36]. However, this is inconvenient for a user wearing mobile HMDs in a mobile environment, because it is not preferable for a user to interact with a virtual object while holding the input device [10]. Lee et al. and Piumsomboon et al. proposed an interaction method to manipulate a virtual object in virtual reality (VR) with head motion and voice, where voice input was used to switch the modes (for example, translation or rotation) of manipulation [37,38]. Although voice input is suitable for use in a VR environment, in a mobile AR environment, the recognition rate of the speech recognition method may drastically decrease due to ambient noise. Ens et al. proposed an interaction method to manipulate the virtual content with touch interaction recognized from a ring-type touchpad and hand motion recognized from a camera built in the HMD [39]. However, because the user has to raise his hand to interact with the virtual content, the fatigue caused to the arms can be problematic when manipulating the virtual content for a long time.

3. Methods

This section describes the proposed interaction method for manipulating a virtual object. In this study, the tasks of manipulating virtual objects are selection, movement, and rotation. Translation and rotation are the most common manipulation tasks in a virtual environment and most similar to manipulating objects in the physical world [27]. Before translating or rotating a virtual object, a user must select it. We first select a virtual object using ray casting, and then, design two interaction methods; we describe these procedures in detail in the followings section.

After the virtual object is selected using ray casting, the tasks of translating the position of the 3D virtual object and rotating its angle are performed using a widget. Here, we use the Wireframe Widget [17] used in the Hologram Application in Hololens, which allows a user to translate a virtual object, rotate it around an axis, and scale it uniformly in three dimensions. Figure 2 shows a flow diagram of the overall state of the manipulation task of a virtual object. After the virtual object is selected, it is selected again and its 3D position is manipulated. When selecting a sphere in the center of the edge on the Wireframe Widget, we can rotate the virtual object around the edge's axis. When selecting a cube at the corner on the Wireframe Widget, we can uniformly scale the size of the virtual object. However, the scaling task is not conducted in the user experiments in this study. When recognizing the tap gesture in a certain manipulation, the manipulation task ends and visualizes the Wireframe Widget again.

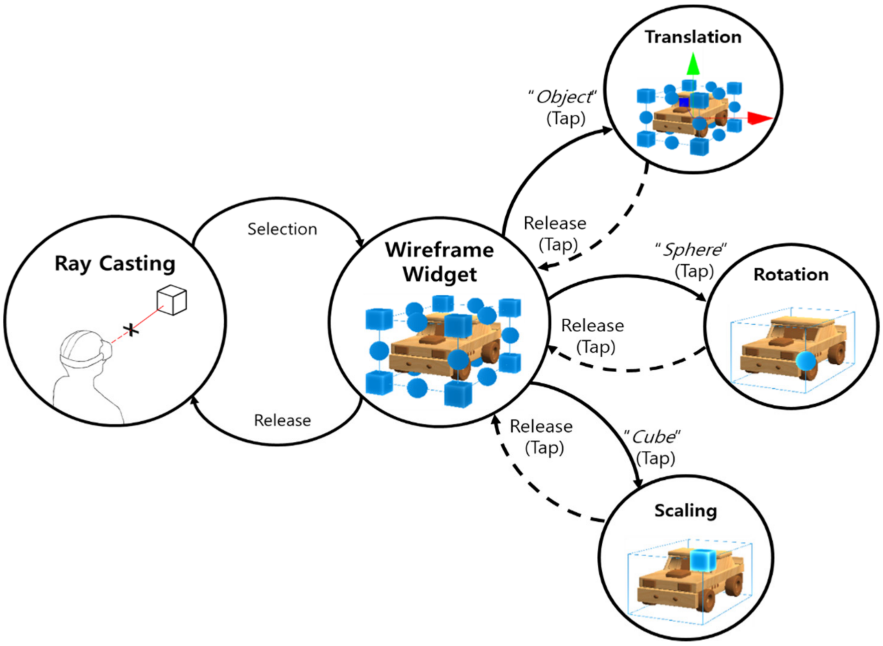


Figure 2. Flow diagram of virtual object manipulation.

3.1. Finger Interaction: Touch Interaction for Manipulating a Virtual Object

In Finger Interaction, the task of selecting a virtual object is performed with AnywhereTouch [14], which is an interface for touch interaction in a mobile AR environment. As AnywhereTouch takes several steps to calculate the slope of the plane on which a finger is touched, to simplify the calculation, we assume that the user performs touch interaction on a horizontal plane. As if manipulating the cursor with a typical mouse, the direction of the virtual line in ray casting is controlled by the finger position tracked from AnywhereTouch and the finger gesture (Tap) recognized from AnywhereTouch is used as a trigger to select a virtual object.

This section describes Finger Interaction, combined with head movements and touch interaction. When manipulating the position of virtual objects with Finger Interaction, the virtual object's position on the XY plane is manipulated by position of the finger on a plane and its height is manipulated by the pitch axis angle of the user's head movements. The user can manipulate the 3D position of the virtual object by using the separation DOF according to the combination of the interactions. When the head movement and touch interaction are performed simultaneously, the 3D position can also be manipulated simultaneously. When a user moves his/her finger on the plane without moving his head, he/she can manipulate only the virtual object's position on the XY plane. Conversely, when the user moves his/her head without moving his/her finger, only the height of the virtual object can be manipulated. Moreover, when manipulating the position of a virtual object, Finger Interaction causes the user to have less fatigue even when manipulating a virtual object for a long time, because it uses two input modalities.

The tasks of rotation and scaling of a virtual object using Finger Interaction can be manipulated with 1DOF by controlling objects (spheres and cubes) of the Wireframe Widget using touch interaction.

In Finger Interaction, the height of the virtual object (position in the *z*-axis) is controlled using head movement. Figure 3 illustrates how to manipulate the height of the selected virtual object using the angle of pitch axis, where Z_{init} is the height (*z*-axis value) of the selected virtual object and θ_{init} is the angle of the pitch axis when the user selects a virtual object. The height's variation (ΔZ) of the selected virtual object is calculated using the distance (XY_{pos}) between the user and the virtual object positions projected on the XY plane and the angle's variation ($\Delta\theta$) of the pitch axis. The height of the virtual object is manipulated by the sum of Z_{init} and ΔZ. The Finger Interaction column of Table 1 describes the manipulation of a virtual object using Finger Interaction.

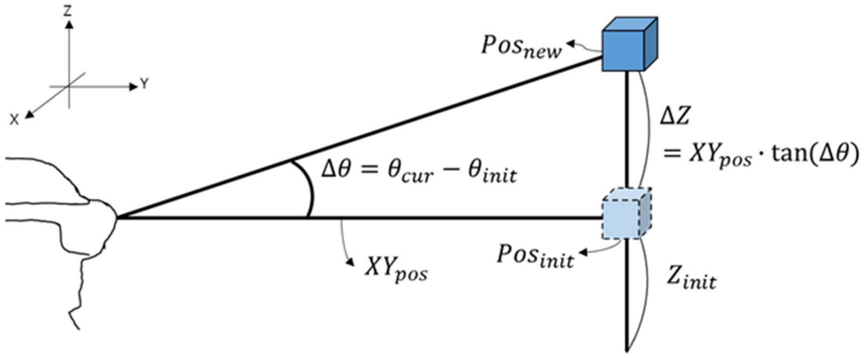
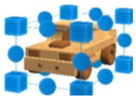
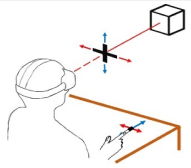
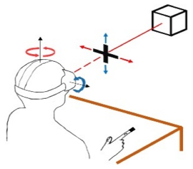
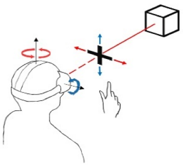
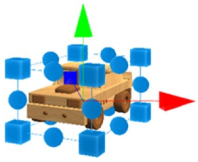
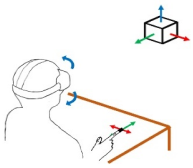
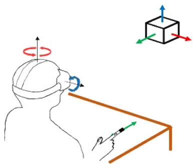
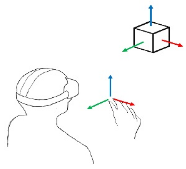
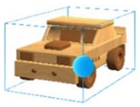
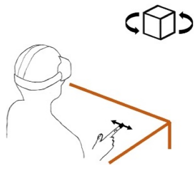
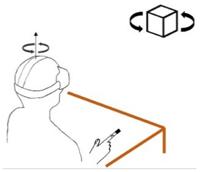
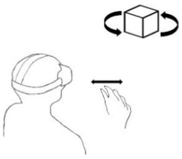

Z
Y
X
Pos_{new}
ΔZ
$= XY_{pos} \cdot \tan(\Delta\theta)$
$\Delta\theta = \theta_{cur} - \theta_{init}$
XY_{pos}
Pos_{init}
Z_{init}

Figure 3. Conceptual illustration of manipulating the height of a virtual object with head rotation.

Table 1. Each interaction method to manipulate a virtual object.

Interaction **Manipulation**	**Finger Interaction**	**Head Interaction**	**Hand Interaction**
 Selection			
 Translation			
 Rotation			

3.2. Head Interaction: Head Motion (Rotation) to Manipulate a Virtual Object

Unlike Finger Interaction, which mainly uses AnywhereTouch to manipulate a virtual object, Head Interaction mainly uses head movements. Here, the ray casting direction is manipulated using the head's pitch and yaw axes and the finger gesture (Tap) is used as a trigger, similarly to the case of

Finger Interaction. With the position of the cursor fixed in the center of the user's FOV, a virtual object in a commercialized HMD has been selected using the head rotation [2–5].

Similar to the case of Finger Interaction, in Head Interaction as well, the position of the virtual object is manipulated by separating the positions of the plane (2DOF) and axis (1DOF). However, in Head Interaction, the position of the virtual object on the YZ plane is manipulated from the head rotation (angles of pitch and yaw axes), while that on the x-axis is manipulated by using the degree to which the finger is bent from AnywhereTouch. Figure 4 shows an example image of manipulating the position of the virtual object on the YZ plane by using the rotation of the pitch and yaw axes. In Figure 4, when a virtual object is selected, it is assumed that there is a virtual YZ plane at the x-axis position of the virtual object. When rotating the head without moving the finger, the position of the virtual object is translated to that where the virtual line extending from the center of the head hits the YZ plane. The user can manipulate the rotation of a virtual object with head rotation, similarly to the position manipulation of a virtual object performed after selecting a sphere of the Wireframe Widget. The Head Interaction column of Table 1 describes the manipulation of a virtual object using Head Interaction.

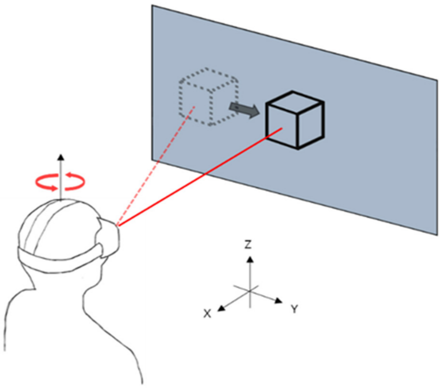


Figure 4. Example image of how to manipulate the position of a virtual object using Head Interaction.

3.3. Hand Interaction: Hand Gesture for Manipulating a Virtual Object

The interaction described in this section is a mid-air hand interaction used in Hololens. In this paper, we refer to this interaction as Hand Interaction, where the Tap gesture (called the air-tap in Hololens) is performed by the index finger and thumb with the other finger bent [4]. This air-tap is used as a trigger in ray casting. The method of manipulating a virtual line of ray casting is the same as the case of Head Interaction.

After selecting a virtual object in Hand Interaction, the user can manipulate its 3D position by selecting the virtual object again. The 3D position of the virtual object can be manipulated by moving the hand freely in three-dimensional space with an air-tap motion. The user can rotate a virtual object with the air-tap motion after selecting the sphere of the Wireframe Widget around the axis he/she wants to rotate. The Hand Interaction column in Table 1 describes the manipulation of a virtual object using Hand Interaction.

4. Experiments

In this study, we conducted user experiments to compare our designed interactions, i.e., Finger Interaction and Head Interaction, with the Hand Interaction used in Hololens. Each interaction was conducted with the user wearing Hololens. When completing the manipulation of the target object, the complete time, position error, and angular error were measured. In addition, to evaluate the usability of each interaction method, the participants were addressed the computer system usability questionnaire (CSUQ) [40], simulator sickness questionnaire (SSQ) [41], and the questionnaire in Table 2, and then, were interviewed individually. While answering the questions in Table 2, the participants were asked to provide their opinions on a five-grade scale ranging from strongly disagree (1) to strongly agree (5). The experiments were approved by the Institutional Review Board in Korea Institute of Science and Technology with ID: 2018-010, and informed consent has been obtained for each subject.

Table 2. Questionnaire.

No.	Question
1	This interaction is convenient to select the target object.
2	This interaction is convenient to translate the target object.
3	This interaction is convenient to rotate the target object.
4	This interaction is convenient for manipulating a virtual object.
5	This interaction is useful for manipulating virtual objects repeatedly.
6	This interaction is convenient for use in public places.

4.1. Task

Before conducting the experiments, the participants were made to practice the use of AnywhereTouch for about 10 min, during which they practiced moving a cursor and tapping to select the menu with AnywhereTouch, like controlling the cursor with a mouse. To prevent drift of the IMU sensor, calibration was performed for each task. The participants performed the tasks of translating the virtual object on the y-axis (Manipulation 1), translating the virtual object on the XY plane (Manipulation 2), rotating the virtual object (Manipulation 3), and manipulating the 3D position and rotation of the virtual object (Manipulation 4) with three interactions. They repeated each task three times. In these experiments, the order of interaction was random, while that of the tasks was sequential (Manipulation 1, Manipulation 2, Manipulation 3, and Manipulation 4). Figure 5 illustrates each task of manipulating a virtual object. After selecting a target object, in each task, the participants were requested to translate or rotate the target object with each interaction to match the objective object. After the experiment for manipulating the virtual object, the participants were requested to answer the questionnaires and were interviewed individually. We evaluated the usability of each interaction from the results of the CSUQ, questionnaire in Table 2, and individual interviews. In addition, we evaluated the degree of dizziness of each interaction from the result of the SSQ.

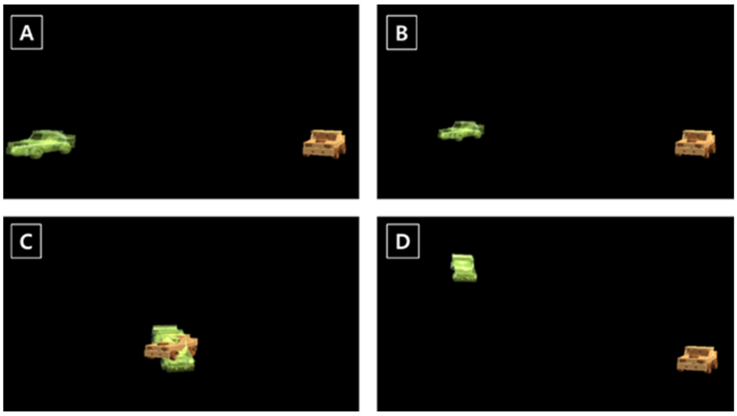


Figure 5. Pictures of each task about the target object (brown Car) and objective object (green Car). (**A**) The task of translating the target object (brown Car) to the objective object (green Car) along the *y*-axis (Manipulation 1). (**B**) The task of translating the target object (brown Car) to objective object (green Car) on the XY plane (Manipulation 2). (**C**) The task of rotating the target object (brown Car) to match the objective object (green Car) around the y- and -axes (Manipulation 3). (**D**) The task of translating and rotating the target object (brown Car) to match the objective object (green Car; Manipulation 4).

4.2. Apparatus

We used a computer with an Intel i5 4690 CPU, 8GB of RAM, and an Nvidia GTX 1060 graphics card. For the recognition of AnywhereTouch, a wireless IMU sensor (EBIMU24GV2, E2Box, Gyeonggi-do, Korea) was used. The HMD used in the experiment was Microsoft's Hololens. The virtual environment is created by Unity3d [24].

4.3. Participants

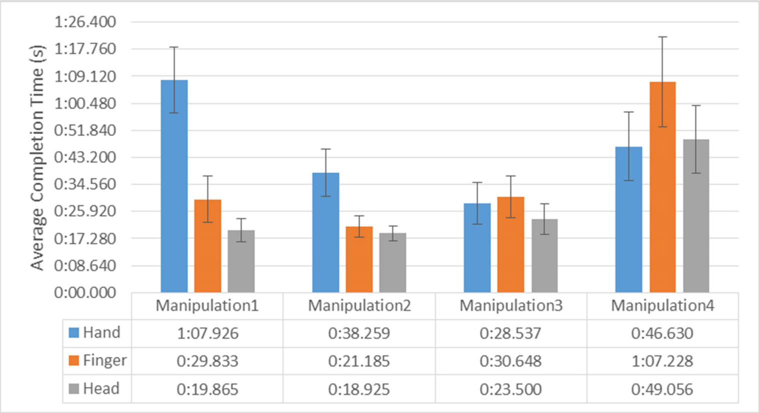
Eighteen subjects (average age = 26.72, STD: 2.49, male = 11, female = 7) participated in this experiment, all of who were right-handed and 10 subjects had AR experience.

5. Results

During the experiments, we collected the data from logging mechanisms and ran the Shapiro–Wilk test to assess the normality of the collected data. The normally distributed data were analyzed using the repeated measures analysis of variance (RMANOVA) test and the other data were analyzed using the Friedman test.

5.1. Completion Time

The completion time was measured from the time when the target object was selected to that when it was released. For Manipulation 1 and Manipulation 2, RMANOVA (Greenhouse–Geisser correction for the violation of sphericity) showed statistically significant differences in completion time (Manipulation 1: $F(2, 34) = 45.902, p < 0.001, \eta^2 = 0.73$; Manipulation 2: $F(2, 34) = 22.186, p < 0.001, \eta^2 = 0.566$). However, for Manipulation 3 and Manipulation 4, there was no statistically significant difference in completion time. Figure 6 shows the graph of completion time for each task.



	Manipulation1	Manipulation2	Manipulation3	Manipulation4
■ Hand	1:07.926	0:38.259	0:28.537	0:46.630
■ Finger	0:29.833	0:21.185	0:30.648	1:07.228
■ Head	0:19.865	0:18.925	0:23.500	0:49.056

Figure 6. Mean completion time for each interaction (+/- standard deviation of the average).

As a result of the post-hoc test conducted using Bonferroni correction, in Manipulation 1, the completion time of Hand Interaction (M ≅ 1 min 08 s) was longer than those of Finger Interaction (M ≅ 30 s, $p < 0.001$) and Head Interaction (M ≅ 20 s, $p < 0.001$). In addition, the post-hoc test revealed that the completion time of Finger Interaction (M ≅ 30 s) was more than that of Head Interaction (M ≅ 20 s, $p < 0.017$).

As a result of the post-hoc test conducted using Bonferroni correction, in Manipulation 2, the completion time of Hand Interaction (M ≅ 38 s) was more than those of Finger Interaction (M ≅ 21 s, $p < 0.001$) and Head Interaction (M ≅ 19 s, $p < 0.001$). However, there was no statistically significant difference between the completion times of Finger Interaction and Head Interaction in Manipulation 2.

5.2. Position Accuracy

The position error in millimeters was calculated using the L2 norm between the positions of the target and objective objects. Figure 7 illustrates the graph of position error for each task, except Manipulation 3. Manipulation 3 did not calculate the position error because it was the task to only rotate the virtual object. The position error for each task did not find a statistically significant difference.

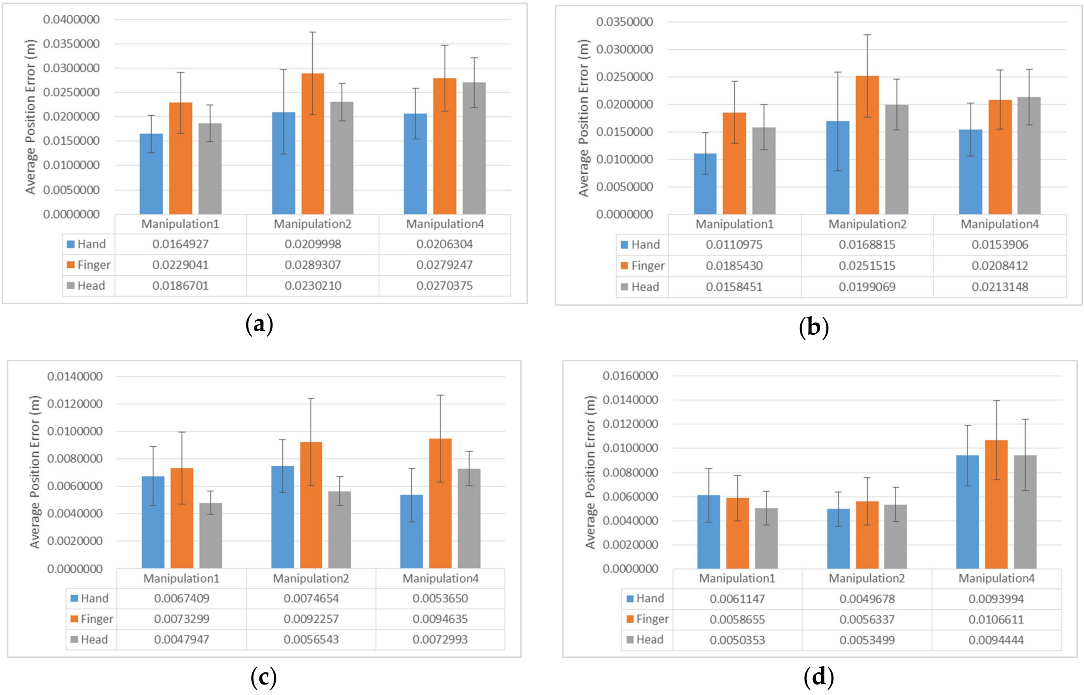


Figure 7. Position error graph (+/− standard deviation of the average). The mean of the (**a**) position error for each interaction, (**b**) x-axis position error for each interaction, (**c**) y-axis position error for each interaction, and (**d**) z-axis position error for each interaction.

Furthermore, we analyzed the position error of each axis. The Friedman test showed no statistically significant difference in the x- and z-axis positional errors among the interaction methods, but in the Manipulation 4 task, the y-axis position error showed a significant difference ($\chi^2(2) = 6.778$, $p = 0.034$). In the Manipulation 4 task, the average of y-axis position error for Hand Interaction (M = 5.36 mm) was significantly less than those for Finger Interaction (M = 9.46 mm, Z = −2.243, $p = 0.025$) and Head Interaction (M = 7.29 mm, Z = −2.417, $p = 0.016$). However, there was no statistically significant difference between Head Interaction and Finger Interaction.

5.3. Angle Accuracy

The angle error in degrees was calculated using the L2 norm between the angles of the target object and objective object. Figure 8 illustrates the graph of the angle error for each task. Manipulation 1 and Manipulation 2 did not calculate the angle error because they were the tasks for only translating the virtual object. As a result of the RMANOVA test (Greenhouse–Geisser correction for the violation of sphericity), there was no statistically significant difference for the three interaction methods $(F(1.680, 28.554) = 3.022,\ p = \ 0.072)$.

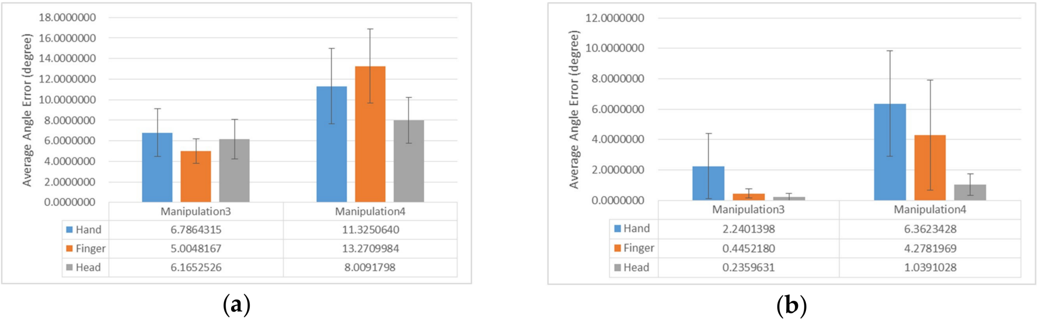
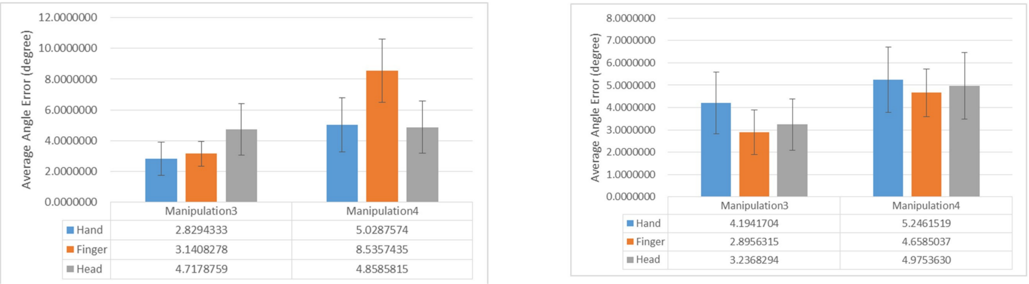


Figure 8. *Cont.*



(**c**) (**d**)

Figure 8. Angle error graph (+/- standard deviation of the average). The mean of the (**a**) angle error for each interaction, (**b**) *x*-axis angle error for each interaction, (**c**) *y*-axis angle error for each interaction, and (**d**) *z*-axis angle error for each interaction.

In addition, we analyzed the angle error of each axis. In Manipulation 3, the average angle error for each interaction was not statistically significant. However, in Manipulation 4, the average of angle error about the *x*-axis ($\chi^2(2) = 13.059$, $p = 0.001$) and *y*-axis ($\chi^2(2) = 6.778$, $p = 0.034$) for each interaction was statistically significant.

The angle error of the *y*-axis for Finger Interaction (M = 8.585°) was statistically significantly greater than those for Hand Interaction (M = 5.028°, Z = −2.156, $p = 0.031$) and Head Interaction (M = 4.858°, Z = −2.548, $p = 0.011$). However, there was no statistically significant difference between Head Interaction and Finger Interaction. On the other hand, the angle error of the *x*-axis for Head Interaction (M = 1.039°) was statistically significantly less than those for Finger Interaction (M = 4.278°, Z = –1.977, $p = 0.048$) and Hand Interaction (M = 6.362°, Z = −3.593, $p < 0.001$).

5.4. Dizziness

As a result of the Friedman test for the SSQ, there was no statistically significant difference in the degree of dizziness for each interaction (nausea: $\chi^2(2) = 1.217$, $p = 0.544$, oculomotor: $\chi^2(2) = 0.133$, $p = 0.936$, disorientation: $\chi^2(2) = 0.133$, $p = 0.945$, total: $\chi^2(2) = 0.034$, $p = 0.983$). The maximum difference in the average SSQ score was about 6. Figure 9 shows the graph of the SSQ score for each interaction.

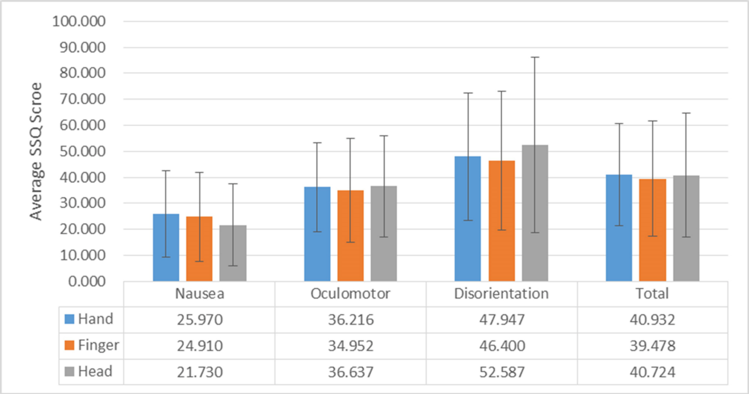


Figure 9. Average of the total simulator sickness questionnaire (SSQ) score and three scores for nausea, oculomotor symptoms, and disorientation for each interaction (+/- standard deviation of the average).

5.5. Usability

The average of the CSUQ score for Head Interaction was slightly higher than those for the other interaction methods. However, we found no significant difference in the CSUQ score using RMANOVA (Huynh–Feldt correction for the violation of sphericity; overall: $\mathrm{F}(2, 34) = 1.693, p = 0.199$, system usability: $F(2,34) = 1.669, p = 0.203$, information quality: $F(2,34) = 1.415, p = 0.257$, interface quality: $F(2,34) = 1.084, p = 0.350$). Figure 10 illustrates a graph of the CSUQ score for each interaction.

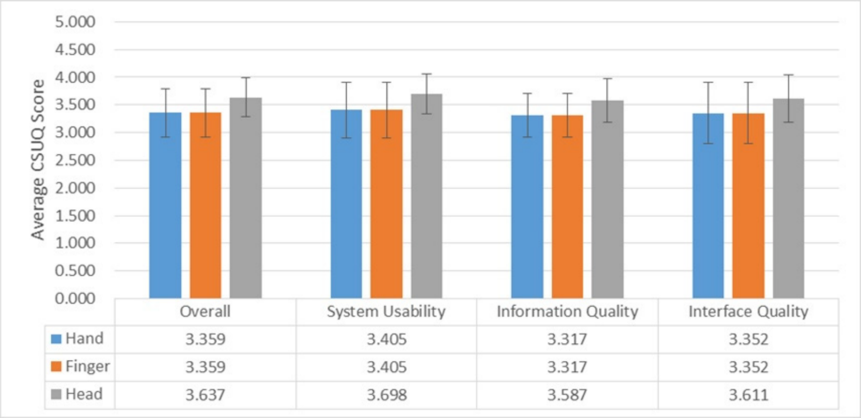


Figure 10. Average of the computer system usability questionnaire (CSUQ) scores for each interaction (+/- standard deviation of the average).

As a result of the questionnaire in Table 2 shown as Figure 11, there were statistically significant differences between Q5 ($\chi^2(2) = 16.259$, $p < 0.0005$) and Q6 ($\chi^2(2) = 7.283$, $p = 0.026$). When repeatedly manipulating a virtual object, the participants did not prefer Hand Interaction (M = 2.056) over Head Interaction (M 3.778, Z = –3.247, $p = 0.001$) and Finger Interaction (M = 3.111, Z = −2.799, $p = 0.005$). Moreover, in this case, the average preference of Finger Interaction was higher than that of Hand Interaction, but there was no statistically significant difference between the two methods. We found that the participants did not prefer Hand Interaction (M = 2.833) over Finger Interaction (M = 3.778, Z = −2.555, $p = 0.011$) and Head Interaction (M = 3.556, Z = −1.960, $p = 0.05$) in public places. Moreover, in this case, there was no statistically significant difference in users' preference between Finger Interaction and Head Interaction.

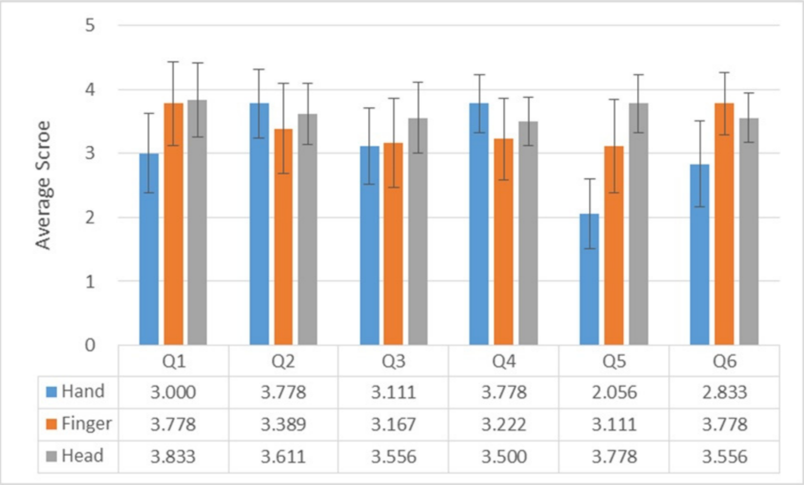


Figure 11. Average score of questionnaires for each interaction (+/- standard deviation of the average).

6. Discussion

6.1. Completion Time

When a participant fulfilled Manipulation 1 and Manipulation 2, which were the tasks of translating a virtual object, Hand Interaction took longer than Head Interaction and Finger Interaction. While translating a virtual object with Hand Interaction, the virtual object was also rotated according to the head rotation. As the participant rotated the virtual object after translating a virtual object with Hand Interaction in Manipulations 1 and 2, the completion time of Hand Interaction was longer than those of the other two interaction methods.

For Manipulation 1 with Hand Interaction, it might have taken a long time to manipulate the virtual objects because of the learning effect. Before manipulating a virtual object for the four tasks, the participants had time to practice AnywhereTouch but did not have time to practice Hand Interaction. However, the participants practiced only to control the cursor with AnywhereTouch before the experiments were initiated. Although the reason why the completion time of Hand Interaction in Manipulation 1 took longer than the other interactions might be that the participants did not have time to practice Hand Interaction, in Manipulation 2, which was performed after Manipulation 1, the completion time of Hand Interaction was longer than those of other interactions because both the position and angle of the virtual object were changed when translating it with Hand Interaction.

In Manipulation 4, the completion time for each interaction was not statistically significant different, but that of Finger Interaction was the longest. Since the control display (CD) ratio of Finger Interaction was higher than those of the other interaction methods, the completion time of Finger Interaction might have been longer than those of the other interaction methods. We set the CD ratio of the finger position in Finger Interaction to above 1 (in this paper, we set CD ratio to 2) because the space whose finger position can be tracked by AnywhereTouch was narrower than that whose virtual object was manipulated. As the CD ratio of Finger Interaction was higher than those of the other interactions when the virtual object was translated after being selected, the virtual object sometimes disappeared in the user's FOV. As the CD ratio of Finger Interaction was high, its position and angle errors tended to be higher than those of the other interaction methods.

6.2. Position Accuracy

Although there was no statistically significant difference in the position error among the three interaction methods, the mean of the position error for Hand Interaction was the smallest, while that for Finger Interaction was the largest. Although the position error difference between Hand Interaction and Head Interaction was about 3 mm in the task of only translating a virtual object, it was increased to about 7 mm in Manipulation 4, which was a task of manipulating the angle and position of a virtual object.

Through analysis of the position error of each axis, we found the cause of the position error of Head Interaction to be greater than that of Hand Interaction. As shown in Figure 7, the average position error between the x- and y-axes in Head Interaction was less than that in Hand Interaction. However, the average position error of the z-axis in Head Interaction was greater than that in Hand Interaction. As the z-axis position of the virtual object was undesirably translated when the participants performed the Tap operation slowly using Head Interaction, the position error of Manipulation 4 using Head Interaction was greater than that of Manipulation 1 or 2.

The position error of the Finger Interaction was higher than those of the other interaction methods because the CD ratio was too high to precisely manipulate a virtual object [42] and the interaction to translate a virtual object along the -axis was the same as the Head Interaction.

6.3. Angle Accuracy

The angle error for the three interaction methods was not statistically significant, but the average angle error for Head Interaction was less than that for Hand Interaction. For the angle error around the x- and z-axes, Head Interaction and Finger Interaction were less than Hand Interaction. However, in Manipulation 3, the angle error around the y-axis for Hand Interaction was larger than that for Hand Interaction and the angle error around the y-axis for Finger Interaction was larger than the y-axis rotation error of Hand Interaction in Manipulation 4. The reason the angle error around the y-axis for Head Interaction and Finger Interaction was larger than that for Hand Interaction is similar to the reason the position error along the x-axis for Head Interaction and Finger Interaction was larger than that for Hand Interaction. As the rotation of the virtual object was performed with the head movement and release was performed with a Tap gesture recognized by AnywhereTouch, there was a little gap in the angle error between Head Interaction and Hand Interaction. However, as both translation of the virtual object and release were performed with finger movement recognized by AnywhereTouch, a gap between the position errors of Head Interaction and Hand Interaction occurred.

6.4. Dizziness

In this paper, we proposed an interaction combining head movement and touch interaction. Since head movements can cause dizziness to the user, we compared the degrees of dizziness of Finger Interaction and Head Interaction using head movement with Hand Interaction through SSQ.

SSQ showed no significant difference in the total SSQ score and three scores for nausea, oculomotor symptoms, and disorientation. Rebenitsch et al. found that the simultaneous pitch- and roll-axis rotations caused higher dizziness to the user than one-axis rotation in the virtual environment [43]. Although we used only the head rotation of one axis to rotate the virtual object in Head Interaction, we used the head rotation of the yaw and pitch axes simultaneously to translate the virtual object in Head Interaction. AR is a technology that visualizes a virtual object in a real physical environment, so the background does not rotate dramatically. Therefore, even if the user's head is rotated (slowly) around the yaw and pitch axes in AR, it does not cause serious dizziness to the user because the surrounding background does not change dramatically.

6.5. Usability

Although the results of the CSUQ score showed that there is no significant difference among the three interaction methods, the results of CSUQ for the three interaction methods were positive for all four aspects: System Usefulness, Information Quality, Interface Quality, and Overall. In addition, Head Interaction got the highest average of the CSUQ score among the three interaction methods.

For Hand Interaction, some participants gave positive feedback that they could manipulate the virtual object by moving their hands. On the other hand, some participants gave negative feedback that during the manipulation, the virtual object was stopped in the middle. This is because the participants performed hand gestures outside the camera's FOV. This feedback was because the user did not know the recognition range of the camera that could recognize the hand gesture when manipulating the virtual object using Hand Interaction.

We found that the participants preferred Head Interaction over Hand Interaction or Finger Interaction when repeatedly manipulating the virtual object. Since Hand Interaction requires the hand to be raised to manipulate the virtual object, this interaction increases the fatigue of the arm when manipulating the virtual object for a long time. In contrast, Finger Interaction does not have to interact with the virtual object with the hand raised, but has a problem that the virtual object is occasionally out of sight due to the problem of high CD ratio. In Head Interaction, on the other hand, the user can manipulate the virtual object without raising his/her hand at the center of his/her FOV.

Moreover, the participants did not prefer to use Hand Interaction in public because the manipulation of the virtual object requires the users to raise their hands. These results can also be found in other studies [10]. On the other hand, Finger Interaction and Head Interaction, which can manipulate the virtual objects without the user needing to raise his/her hands, received high scores from participants.

7. Conclusion

This paper proposed interaction methods to conveniently manipulate a virtual object by combining touch interaction and head movements on a mobile HMD for experiencing AR. The proposed interaction methods (Head Interaction and Finger Interaction) were evaluated through user experiments through a comparison with Hand Interaction on Hololens. There was no statistically significant difference in the objective data, which comprised of the completion time, position error, and angle error obtained from the user experiments. The usability assessment obtained from the questionnaire and individual interviews showed that the proposed interaction methods (Head Interaction and Finger Interaction) are better than Hand Interaction. In particular, the results of the questionnaires addressed to the participants indicated that Head Interaction was preferred over other interaction methods for selecting the virtual object, repeatedly manipulating it, and manipulating it in public places. Therefore, Head Interaction makes it possible to conveniently manipulate a virtual object for a user wearing a mobile HMD to provide AR. In the future, we plan to research interfaces and interactions that can precisely manipulate the virtual object in a mobile AR.

Author Contributions: Conceptualization, J.Y.O. and J.H.P.; methodology, J.Y.O.; software, J.Y.O.; validation, J.Y.O.; writing—original draft preparation, J.Y.O.; writing—review and editing, J.-M.P.; supervision, J.-M.P.; project administration, J.H.P.; funding acquisition, J.H.P. and J.M.P.

Funding: This research was funded by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government (MSIP) (2011-0031380, 2011-0031425)

Conflicts of Interest: The authors declare no conflict of interest.

References

1. LaViola, J.J., Jr.; Kruijff, E.; McMahan, R.P.; Bowman, D.; Poupyrev, I.P. *3D User Interfaces: Theory and Practice*; Addison-Wesley Professional: Boston, MA, USA, 2017.
2. Google Glass. Available online: https://www.google.com/glass/start/ (accessed on 21 July 2019).
3. Leap, M. Magic Leap. Available online: https://www.magicleap.com/ (accessed on 21 July 2019).
4. Microsoft Hololens. Available online: https://www.microsoft.com/en-us/p/microsoft-hololens-development-edition/8xf18pqz17ts?activetab=pivot:overviewtab (accessed on 21 July 2019).
5. Sony Smarteyeglass SED E1. Available online: https://developer.sony.com/develop/smarteyeglass-sed-e1/ (accessed on 21 July 2019).
6. Joolee, J.; Uddin, M.; Khan, J.; Kim, T.; Lee, Y.K. A Novel Lightweight Approach for Video Retrieval on Mobile Augmented Reality Environment. *Appl. Sci.* **2018**, *8*, 1860. [CrossRef]
7. Phan, V.T.; Choo, S.Y. Interior design in augmented reality environment. *Int. J. Comput. Appl.* **2010**, *5*, 16–21. [CrossRef]
8. Nóbrega, R.; Correia, N. CHI'11 Extended Abstracts on Human Factors in Computing Systems. In *Design Your Room: Adding Virtual Objects to a Real Indoor Scenario*; ACM: New York, NY, USA, 2011; pp. 2143–2148.
9. Gugenheimer, J.; Dobbelstein, D.; Winkler, C.; Haas, G.; Rukzio, E. Face Touch: Enabling Touch Interaction in Display fixed Uis for Mobile Virtual Reality. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology, Tokyo, Japan, 16–19 October 2016; ACM: New York, NY, USA, 2016; pp. 49–60.
10. Tung, Y.C.; Hsu, C.Y.; Wang, H.Y.; Chyou, S.; Lin, J.W.; Wu, P.J.; Valstar, A.; Chen, M.Y. User-Defined Game Input for Smart Glasses in Public Space. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, Seoul, Korea, 18–23 April 2015; ACM: New York, NY, USA, 2015; pp. 3327–3336.
11. Mendes, D.; Relvas, F.; Ferreira, A.; Jorge, J. The Benefits of DOF Separation in Mid-Air 3d Object Manipulation. In Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology, Munich, Germany, 2–4 November 2016; ACM: New York, NY, USA, 2016; pp. 261–268.
12. Bader, T.; Vogelgesang, M.; Klaus, E. Multimodal Integration of Natural Gaze Behavior for Intention Recognition During Object Manipulation. In Proceedings of the 2009 International Conference on Multimodal Interfaces, Cambridge, UK, 2–4 November 2009; ACM: New York, NY, USA, 2009; pp. 199–206.
13. Fukazawa, R.; Takashima, K.; Shoemaker, G.; Kitamura, Y.; Itoh, Y.; Kishino, F. Comparison of multimodal interactions in perspective-corrected multi-display environment. In Proceedings of the 2010 IEEE Symposium on 3D User Interfaces (3DUI), Waltham, MA, USA, 20–21 March 2010; pp. 103–110.
14. Oh, J.Y.; Lee, J.; Lee, J.H.; Park, J.H. Anywheretouch: Finger tracking method on arbitrary surface using nailed-mounted imu for mobile hmd. In Proceedings of the International Conference on Human-Computer Interaction, Vancouver, BC, Canada, 9–14 July 2017; pp. 185–191.

15. Özacar, K.; Hincapié-Ramos, J.D.; Takashima, K.; Kitamura, Y. 3D Selection Techniques for Mobile Augmented Reality Head-Mounted Displays. *Interact. Comput.* **2016**, *29*, 579–591. [CrossRef]
16. Mine, M.R. *Virtual Environment Interaction Techniques*; UNC Chapel Hill CS Dept: Columbia, SC, USA, 1995.
17. Microsoft Bounding Box Default Handle Sytle. Available online: https://github.com/microsoft/MixedRealityToolkit-Unity/blob/mrtk_release/Documentation/README_HandInteractionExamples.md (accessed on 21 July 2019).
18. Bowman, D.A.; Johnson, D.B.; Hodges, L.F. Testbed evaluation of virtual environment interaction techniques. *Presence Teleoperators Virtual Environ.* **2001**, *10*, 75–95. [CrossRef]
19. Hoppe, A.H.; van de Camp, F.; Stiefelhagen, R. Interaction with three dimensional objects on diverse input and output devices: A survey. In Proceedings of the International Conference on Human-Computer Interaction, Vancouver, BC, Canada, 9–14 July 2017; pp. 130–139.
20. Chen, M.; Mountford, S.J.; Sellen, A. *A Study in Interactive 3-D Rotation Using 2-D Control Devices*; ACM: New York, NY, USA, 1988; pp. 121–129.
21. Conner, B.D.; Snibbe, S.S.; Herndon, K.P.; Robbins, D.C.; Zeleznik, R.C.; Van Dam, A. Three-dimensional widgets. In Proceedings of the 1992 Symposium on Interactive 3D graphics, San Diego, CA, USA, 10–11 December 1990; ACM: New York, NY, USA, 1992; pp. 183–188.
22. Houde, S. Iterative design of an interface for easy 3-D direct manipulation. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Monterey, CA, USA 3–7 May 1992; ACM: New York, NY, USA, 1992; pp. 135–142.
23. Shoemake, K. ARCBALL: A User interface for specifying three-dimensional orientation using a mouse. In *Graphics Interface*; Canadian Information Processing Society: Mississauga, ON, Canada, 1992; pp. 151–156.
24. Techologies, U. Unity3d. Available online: https://unity.com/ (accessed on 21 July 2019).
25. Autodesk Maya. Available online: https://www.autodesk.com/products/maya/overview (accessed on 21 July 2019).
26. Blender Blender. Available online: https://www.blender.org/ (accessed on 21 July 2019).
27. Mendes, D.; Caputo, F.M.; Giachetti, A.; Ferreira, A.; Jorge, J. Computer Graphics Forum. In *A Survey on 3d Virtual Object Manipulation: From the Desktop to Immersive Virtual Environments*; Wiley Online Library: Hoboken, NJ, USA, 2019; pp. 21–45.
28. Cohé, A.; Dècle, F.; Hachet, M. tBox: A 3d Transformation Widget Designed for Touch-Screens. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, 7–12 May 2011; ACM: New York, NY, USA, 2011; pp. 3005–3008.
29. Mendes, D.; Lopes, P.; Ferreira, A. Hands-on Interactive Tabletop Lego Application. In Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology, Lisbon, Portugal, 8–11 November 2011; ACM: New York, NY, USA, 2011; pp. 19–26.
30. Au, O.K.C.; Tai, C.L.; Fu, H. Computer Graphics Forum. In *Multitouch Gestures for Constrained Transformation of 3d Objects*; Wiley Online Library: Hoboken, NJ, USA, 2012; pp. 651–660.
31. Xiao, R.; Schwarz, J.; Throm, N.; Wilson, A.D.; Benko, H. MRTouch: Adding Touch Input to Head-Mounted Mixed Reality. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 1653–1660. [CrossRef] [PubMed]
32. Sharp, T.; Keskin, C.; Robertson, D.; Taylor, J.; Shotton, J.; Kim, D.; Rhemann, C.; Leichter, I.; Vinnikov, A.; Wei, Y. Accurate, Robust, and Flexible Real-Time Hand Tracking. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, Seoul, Korea, 18–23 April 2015; ACM: New York, NY, USA, 2015; pp. 3633–3642.
33. Frees, S.; Kessler, G.D. Precise and rapid interaction through scaled manipulation in immersive virtual environments. In Proceedings of the IEEE Proceedings Virtual Reality (VR 2005), Bonn, Germany, 12–16 March 2005; pp. 99–106.
34. Poupyrev, I.; Billinghurst, M.; Weghorst, S.; Ichikawa, T. The go-go interaction technique: Non-linear mapping for direct manipulation in VR. In Proceedings of the ACM Symposium on User Interface Software and Technology, Seattle, WA, USA, 6–8 November 1996; pp. 79–80.
35. Bowman, D.A.; Hodges, L.F. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. *SI3D* **1997**, *97*, 35–38.
36. Nancel, M.; Chapuis, O.; Pietriga, E.; Yang, X.D.; Irani, P.P.; Beaudouin-Lafon, M. High-precision pointing on large wall displays using small handheld devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Paris, France, 27 April–2 May 2013; ACM: New York, NY, USA, 2013; pp. 831–840.

37. Lee, M.; Billinghurst, M.; Baek, W.; Green, R.; Woo, W. A usability study of multimodal input in an augmented reality environment. *Virtual Real.* **2013**, *17*, 293–305. [CrossRef]
38. Piumsomboon, T.; Altimira, D.; Kim, H.; Clark, A.; Lee, G.; Billinghurst, M. Grasp-Shell vs gesture-speech: A comparison of direct and indirect natural interaction techniques in augmented reality. In Proceedings of the 2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Munich, Germany, 10–12 September 2014; pp. 73–82.
39. Ens, B.; Byagowi, A.; Han, T.; Hincapié-Ramos, J.D.; Irani, P. Combining Ring Input with Hand Tracking for Precise, Natural Interaction with Spatial Analytic Interfaces. In Proceedings of the 2016 Symposium on Spatial User Interaction, Tokyo, Japan, 15–16 October 2016; ACM: New York, NY, USA, 2016; pp. 99–102.
40. Lewis, J.R. IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *Int. J. Hum. Comput. Interact.* **1995**, *7*, 57–78. [CrossRef]
41. Kennedy, R.S.; Lane, N.E.; Berbaum, K.S.; Lilienthal, M.G. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *Int. J. Aviat. Psychol.* **1993**, *3*, 203–220. [CrossRef]
42. Argelaguet, F.; Andujar, C. A survey of 3D object selection techniques for virtual environments. *Comput. Graph.* **2013**, *37*, 121–136. [CrossRef]
43. Rebenitsch, L.; Owen, C. Review on cybersickness in applications and visual displays. *Virtual Real.* **2016**, *20*, 101–125. [CrossRef]