AR Pointer: Advanced Ray-Casting Interface Using Laser Pointer Metaphor for Object Manipulation in 3D Augmented Reality Environment

Hyocheol Ro [1], Jung-Hyun Byun [1], Yoon Jung Park [1], Nam Kyu Lee [2] and Tack-Don Han [1,*]

1   Media System Lab, Yonsei University, Seoul 03722, Korea
2   College of Computer Science & Software, Chung-Ang University, Seoul 06974, Korea
*   Correspondence: hantack@msl.yonsei.ac.kr

check for updates

Abstract: In this paper, we propose AR Pointer, a new augmented reality (AR) interface that allows users to manipulate three-dimensional (3D) virtual objects in AR environment. AR Pointer uses a built-in 6-degrees of freedom (DoF) inertial measurement unit (IMU) sensor in an off-the-shelf mobile device to cast a virtual ray that is used to accurately select objects. It is also implemented using simple touch gestures commonly used in smartphones for 3D object manipulation, so users can easily manipulate 3D virtual objects using the AR Pointer, without a long training period. To demonstrate the usefulness of AR Pointer, we introduce two use-cases, constructing an AR furniture layout and AR education. Then, we conducted two experiments, performance tests and usability tests, to represent the excellence of the designed interaction methods using AR Pointer. We found that AR Pointer is more efficient than other interfaces, achieving 39.4% faster task completion time in the object manipulation. In addition, the participants gave an average of 8.61 points (13.4%) on the AR Pointer in the usability test conducted through the system usability scale (SUS) questionnaires and 8.51 points (15.1%) on the AR Pointer in the fatigue test conducted through the NASA task load index (NASA-TLX) questionnaire. Previous AR applications have been implemented in a passive AR environment where users simply check and pop up the AR objects those are prepared in advance. However, if AR Pointer is used for AR object manipulation, it is possible to provide an immersive AR environment for the user who want/wish to actively interact with the AR objects.

Keywords: augmented reality; head-mounted display; intelligent user interface; ray-casting; mixed reality; augmented reality interface; head-mounted display interface; ray-casting interface

1. Introduction

Augmented reality (AR) technology, which shows virtual objects overlapping in the real world, is gaining increasing attention and various AR applications are being developed. These applications were initially developed as mobile AR based on mobile devices, such as smartphones and tablet PCs. The mobile AR systems used the camera of the smartphone to show the real world while augmenting the virtual objects in the camera screen. Since then, smartphones have become widely available, making it easy for anyone to enjoy AR applications. For example, Pokemon Go is the most successful AR mobile game based on the location-based service (LBS). In this game, creatures are registered nearby and the user hunts the creatures that can be augmented through touch gestures [1]. IKEA's catalog application is a typical example of a mobile-based AR application. Early versions of this application used a real catalog of IKEA as a kind of marker. If a user places the catalog in a location where they want to pre-position the furniture, the mobile device recognizes the marker and augments the furniture. With the development of computer vision technology, the floor and

wall can be distinguished by a mobile device with Apple ARKit or Google AR Toolkit [2] and the furniture can be placed in the desired position without markers. However, when using mobile-based AR applications, the user is required to continue to hold the device to superimpose the virtual object, which is the most significant limitation of AR based on mobile devices.

Since then, AR Google Glass, a wearable device, has been launched by Google and applications based on a head-mounted display (HMD) have been proposed in earnest. Unlike mobile AR, which needs to confirm virtual objects with a narrow display screen, HMD-based AR applications provide a convenient and immersive experience. HMD-based AR can be applied to a wide variety of fields because when the HMD is worn, the object augmentation and manipulation can be performed at the same time by the user. In particular, various AR applications using HMD are expected to be used in teleconferencing, tele-education, military, maintenance, repair operation (MRO) and exhibition [3]. As such, Google Glass has pioneered the market for wearable AR devices by implementing a simple level of augmented reality functionality. Microsoft, on the other hand, has further improved the realization of augmented reality with HoloLens, which provides immersive 3D screens with high-quality realism. Since the HoloLens is actually equipped with the simultaneous localization and mapping (SLAM) function, it can acquire the geometry information of the space in which the user is located and grasp its position. This makes it possible to place virtual objects on the desk or mid-air in 3D space. Then, the virtual object is correctly registered at a specific point and it is possible for the user to watch the object, such as a hologram, by moving around it.

Interfaces developed for 2D display [4], such as a keyboard, mouse, touch screen and a stylus pen in existing computer and mobile device environments, are difficult to use smoothly in an AR environment where objects are registered and manipulated in real 3D space, such use HMD-based AR [5]. This is because these user interfaces have a 2D input method that cannot control the z-Axis (depth) [6]. So, with the increasing number of applications and usage scenarios involving augment 3D virtual objects in the real world and interactions with them, there is a growing demand for bespoke AR interfaces that work naturally within a 3D AR environment.

Even the most advanced HMD AR device, the HoloLens, lacks an interface that allows for easy interaction. Although the precise selection and manipulation of objects are among the most important issues in the 3D AR environment [6], the current gaze-assisted selection (GaS) interface of the HoloLens is somewhat inaccurate and has poor usability [7]. This is because it generates a ray that is not visible in the user's head direction and uses the point where the ray intersects as a cursor, such as a mouse pointer. The user must continue to move their head to control the pointer, resulting in fatigue (especially of the user's neck) and the accuracy of selecting objects over long distances will be greatly reduced [5]. To use the fingertip gesture, the user's hand must be located within the field of view of the camera mounted on the HoloLens. The users must continuously raise their hand high, resulting in fatigue in shown as Figure 1. Ultimately, this fatigue limits the user's application usage time.
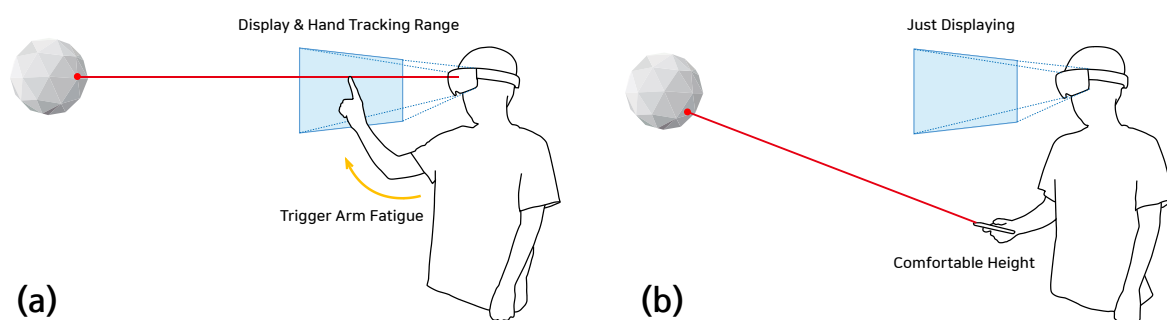
Figure 1. Hand Fatigue. (**a**) Previous HoloLens manipulation methods cause a lot of arm fatigue to the user. (**b**) The proposed interface does not need to raise the arm, so the user feels comfortable when manipulating the object.

On the other hand, a rather unintuitive way of interacting with the object is used. When you select an object, eight editing boxes in the form of a wireframe are activated and the user can adjust the size of the object by pulling these editing boxes, one by one [7]. This is borrowed from the method used to adjust the size of a 2D media file such as a photograph or graphic in office programs, such as word processing and presentation programs. Since it uses a similar method to the interface used in the existing computer environment (since it is already learned), there is no need to separately learn an interface for object manipulation. However, this interface is less accurate and usable for use in 3D environments. Because it is necessary to select a relatively small edit box compared to the object, the ray-casting based interface is generally difficult to select for small object selection [5].

There is also a limit to the object registration method. With the existing HoloLens interface, there are no significant problems when registering an object that is in contact with a plane, such as a floor, desk or wall. However, virtual reality (VR) and AR environments frequently register objects floating in mid-air differently from the real world. These include educational objects, flying objects, personal widgets and multi-purpose windows. In the HoloLens AR interface, the area in which these objects can be registered is limited to the distance the user can reach. In other words, for a user to register an object at a long distance, the user must move directly to the vicinity of the location. In an application that requires various frequent object registration and manipulation, the user must move a long distance each time to move a remote object. Therefore, if the interface is used in an AR environment, the authoring situation and task becomes somewhat inefficient.

In this paper, we propose an AR Pointer, a new user interface for convenient registration and manipulation of remote objects in the AR environment. This interface is designed to be intuitive and highly usable based on ray-casting using the Depth-variable method [5]. In particular, it will be useful to replace existing interfaces in see-through HMD-based AR applications, such as the HoloLens. Instead of creating an input device with a separate sensor, we made it possible for users to use their mobile device as an input device. This is because mobile devices are basically equipped with 6-Degree of Freedom (DoF) Inertial Measurement Unit (IMU) sensors and a touch screen. The proposed interface can use the Depth-variable method to register an object at a specific point in the distance and in the air. This AR interface also allows the user to manipulate objects using a familiar mobile and user-friendly swipe/pinch gesture. By applying this familiar interaction method, users who have no expertise or experience with the AR environment can use it easily. This design allows users to interact actively with AR contents.

In Section 2, we describe related work on 3D object manipulation. Section 3 describes how to implement AR Pointer and object manipulation interactions. Section 4 and Section 5 presents two experimental results for each two methods. In the final section, we discuss the conclusion and future work of this paper.

2. Related Work

The user interfaces for 3D object manipulation is divided into two main approaches. One is using vision technology and the other is using sensor and ray-casting technology. In the first approach, which uses vision technology, the user recognizes the user's hand with the camera and performs the interaction by grabbing the object directly. In the second approach, using sensor-based ray-casting, a sensor is used to generate a virtual ray and determine whether it intersects with the object to perform the interaction.

2.1. Vision-Based Approach

First, with a vision-based system, users can use their body parts as a 3D cursor, which is useful for intuitively selecting and manipulating 3D objects in mid-air. For example, Soh et al. [8], implemented a 3D kinematics tool that enables users to manipulate 3D objects using gestures in 3D space, using Microsoft Kinect. Furthermore, one gesture-based interface was released for 3D modeling in virtual reality (VR) with Leap Motion [9]. Vision-based interfaces are natural and intuitive,

so users do not have to learn manually because they already use their hands and body parts in this manner. Moreover, since users feel that they directly control the 3D space, this approach can provide an immersive effect [10].

However, the use of a camera to accurately detecting hands or body parts remains challenging. For example, occlusion problems might occur depending on the installation position, direction and posture of the camera and detection performance might vary significantly as a result. Occlusion problems remained even in a study that showed considerable improvement in hand tracking performance [11]. In terms of AR, 3D object manipulation based on freehand gestures is not suitable for applications that require high-level accuracy. Moreover, the method of registering a 3D object with the user's hand exhibits a significant limitation. Although it is possible to register relatively precisely within reach of the user's hands and arms [12], registering to an area further away requires additional gestures, which is associated with a longer learning curve [9].

On the other hand, HoloLens uses the bare hand as an auxiliary means with a click function but also considers the position of the hand in object registration interaction. The distance that objects can be registered in mid-air is limited to the range of the hand, thus sharing the problem of vision-based hand interfaces. The limitation of remote object manipulation when using the bare hand has been actively discussed in the early VR interaction research but even after several decades, no dominant solution (or method) has yet emerged [13]. In relatively recent studies, Whie et al. [14], attempted to solve this problem by modifying the GoGo interaction [15] and implementing a switchable interface that allows a virtual hand to reach a distance with a simple gesture change. As a result, intuitiveness, which is the greatest advantage of vision-based interfaces, is reduced and usability is also decreased.

2.2. Sensor-Based Approach

The second research approach is the use of a sensor-based controller and ray-casting technology, which has been studied for a long time in the VR field [5]. HTC Vive, one of the most popular VR devices, has already implemented ray-casting using a controller [16]. However, this requires two external trackers and requires a separate dedicated controller so that it can be used only in a limited environment. In the GyroWand [17] study, it was possible not only to use the ray in the form of a line but also to adjust it to a volumetric form if necessary. This was done to increase accuracy when selecting objects at a distance. In this study, however, there were limitations in addressing only object selection among various object manipulations. Most similar ray-casting studies [5,14,15] have focused on object selection and few studies have addressed other object manipulations. As we can see from the interfaces used for VR applications, it is often necessary to select the type of manipulation via the object selection and the pie menu or the traditional linear menu. HoloLens also shares the same issue. As mentioned earlier, this raises the problem that the object manipulations must go through several steps and this method is not intuitive. In the study of Chaconas and Hollere [7] on the HoloLens interface, the object was resized and rotated using a two-hand interface but the problem of fatigue, which was associated with the default interface of HoloLens, remained.

The computer vision-based interface is intuitive and feels natural because it uses the hand. Also, since it is a hand metaphor, little learning is required and the tasks of selecting (grabbing) and manipulating (translation, rotation, scaling) are performed continuously. However, the range that can be used is limited to the distance the user can extend their arm and in order to manipulate the remote object, the user always has to move near the object. On the other hand, controller-based interfaces mainly use ray-casting technology. Ray-casting technology allows the user to select an object through additional tasks (button selection, gesture or voice) after pointing at an object at a distance (laser pointer metaphor) and intersecting the ray. Therefore, it is less intuitive and natural than the hand manipulation method. To solve this problem, Whie et al. [13], developed an approach in which one hand is used as a pointer metaphor to point to and select an object and then, when manipulating the selected object, the object is replicated in front of the user, allowing the other hand to use the same manipulations as existing vision-based interfaces. However, since the research used only one hand for

object manipulation, they could not propose an object scaling method, which is one of the basic object manipulations. Also, their study was limited in that quantitative and qualitative analyses were not conducted to test whether this method is effective for users. In other words, traditional ray casting interactions still have used obtrusive and unintuitive methods such as pop-up menus for rotation, selection and translation. So in this paper, we proposed an object manipulation method that can be easily used by ordinary users. In other words, if the previous work [5] focuses on the selection of objects, this paper address manipulation such as rotation, selection, translation after performing selection.

3. AR Pointer

3.1. Implementation

Our proposed AR Pointer was implemented using ray-casting technology and employing a mobile device as a sensor-based controller. The Ray-casting method can reduce fatigue when using the HMD because the user does not have to raise their hand by force and it is useful for controlling the remote objects. One of the biggest problems with ray-casting technology interfaces, manipulation, was solved by using the smartphone's swipe/pinch gesture. A smartphone with an acceleration, gyro sensor and touch screen was used as the input devices. Therefore, it was possible to accurately generate the direction of the ray through the sensor data. The following is a detailed implementation of the proposed AR Pointer.

AR Pointer is an interface designed for object registration and manipulation in AR environments. Here we used the IMU sensor (gyro sensor and accelerometer sensor) of a smartphone (we used the Apple iPhone 8 but it can be implemented on all iPhone and Android phones with 6-DoF IMU sensors.) to accurately point in the desired direction and guarantee the high accuracy required for the sensor-based approach. We also implement the user interface so that it can be used similarly to a laser pointer, to provide a natural and intuitive interface. To create a method similar to a laser pointer, we constructed an interface based on ray-casting technology.

Ray-casting is a 3D user interface implementation technique often used in VR environments [14,18] and is a technology for discriminating the intersection between rays extending from the user and a virtual object. Ray-casting is mainly used to select 3D virtual objects that intersect with the ray [19] and is rarely used in manipulations such as the registration, translation, scaling and rotation of objects. In traditional ray-casting, instead of specifying a point in mid-air in a 3D space, we assume that the ray will extend indefinitely, as with the physical characteristic of a laser pointer. Therefore, it is impossible to perform interaction methods, such as registering an object in mid-air on setting its movement path. We solve this problem by including the ray's segment length information, which we define as ray-depth. Also, this ray-depth can be adjusted to specify the desired position in 3D space. In our prototype, ray-depth control and content manipulation were implemented using the smartphone touch screen.

The information needed to specify the location of the AR Pointer (i.e., the starting point of the ray) is based on user location. Because of the high accuracy of the SLAM, the HoloLens can determine the position of the user in the 3D space created. Next, we need to determine the location of the mobile device the user is holding. As described above, the HoloLens can detect only the hand located in a narrow area in the viewing direction. In other words, it cannot keep track of the exact location of the user's mobile device. Therefore, in order to perform a second experiment 2 comparing the basic interface of the HoloLens with the proposed AR Pointer, we measured the relative position of the mobile device and the HMD when using the AR Pointer. An external camera was needed to determine the precise location of the AR Pointer. Because our prototype was able to use predefined and reconstructed space, we installed a Microsoft Kinect v2 device to detect the user's hand position. The Kinect device measures and stores the user's 3D skeleton joints information, determines position using pre-calibrated values and then converts the camera coordinate system to the world coordinate system. This information can be applied when using the AR Pointer in HMD-based AR used in predefined environments or other AR environments without SLAM capabilities (such as

spatial AR like a projection-based AR). We used this 3D skeleton information throughout experiment 1. Experiment 1 compares the manipulation method of the Unity Engine [20] with the AR Pointer method in the monitor screen environment instead of the HMD.

To implement the AR Pointer, we estimated its horizontal and vertical angles through the IMU sensor data of the smartphone. We used the gyro sensor because it provides accurate values within a short time while the user is moving. The accelerometer was also used because it provides an accurate slope value for a long period when the user stops. However, the mobile device's naive rotational data presents certain problems. As the gyro sensor uses the integral value of angular velocity, the error increases with time. In the case of the acceleration sensor, the user's acceleration is added to the sensor value and an error occurs. Therefore, a complementary filter is used for the sensor value [1].

In the AR Pointer, ray-depth information is included in addition to sensor information, because ray-depth must be adjusted for pointing in mid-air in a 3D space. Touch gesture methods are used for depth control: swiping the finger up (pointing direction) on the touch screen allows the user to point farther, while swiping the finger down (towards the user) allows them to point closer in shown as Figure 2. It is possible to point to a particular point 3D space by combining the position of the user and AR Pointer, IMU sensor value and ray-depth information.

In Figure 2a D means the end point of the AR Pointer, θ_v and θ_h denote the vertical and horizontal angles, respectively and R represents the AR Pointer's ray-depth information. By using these three pieces of information, the end point can be obtained in the orthogonal coordinate system when the user's position S is fixed. The height of the pointing device held by the user is set to y_p and included in the end point formula. In addition, when the AR Pointer is used while the person is moving, x_p and z_p are added to the X and Z coordinates, respectively. As shown in Figure 2b–g, we were finally able to visualize the user's location in 3D space, as well as the direction and ray-depth of AR Pointer.
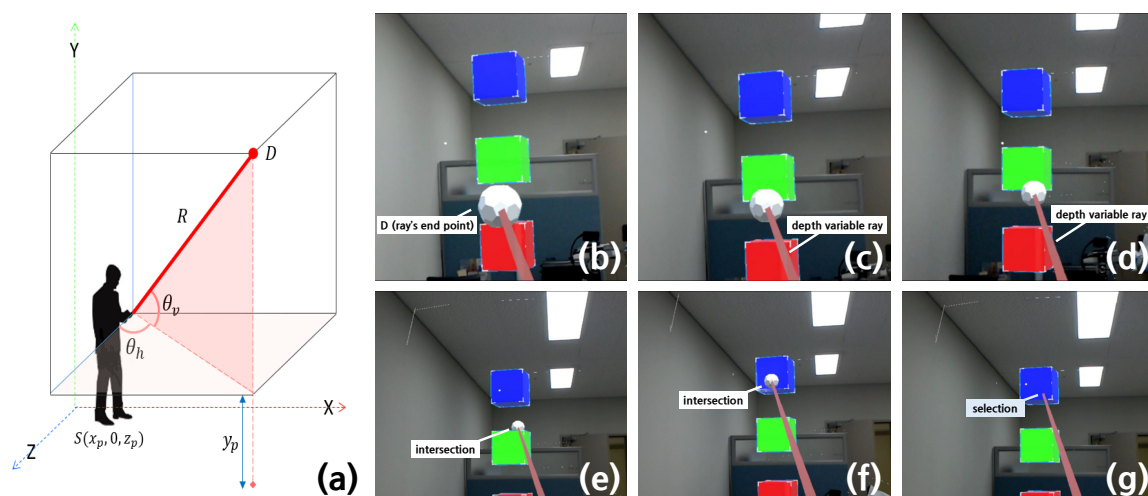

Figure 2. AR Pointer Implementation. (**a**) AR Pointer's ray-casting using 6-Degrees of Freedom (DoF) Inertial Measurement Unit (IMU) sensor. (**b–g**) AR Pointer is capable of adjusting the depth of the ray and when the end point of the ray intersects the object, the user can select the object for manipulation.

3.2. User Interaction for Object Manipulations

AR Pointer can specify ray direction and the direction's ray-depth can be adjusted to calculate the ray's endpoint. Because this method is based on ray-casting, when a touch event is an input, it checks whether any virtual object intersects with the ray. When the ray does not intersect with any object (Figure 3c), object registration mode is entered, in which virtual objects can be registered in space. To register an object at a specific location (Figure 3d), a double-tap gesture is used on the smartphone. The virtual object to be registered in the real world can be confirmed in advance through a single tap. When a ray intersects with an object (Figure 3f), the object is selected using a single tap (Figure 3g)

and the object manipulation mode is entered, in which various manipulations can be performed. In this mode, core gestures (Figure 3h,i), mainly those used in smartphones, are used similarly to manipulate the 3D object [2,21]. The object control method takes place as follows: a pinch gesture, which places two fingers together, enables the object shrinking operation. Conversely, a magnification manipulation can be performed by a spread gesture. Also, two-finger drag gestures can be used to rotate in either direction of the 3-DoF. To move the object, a long tap gesture is used to de-register the object and cause it to be ready to re-register. Finally, a single tap releases the selected object and exits the manipulation mode.
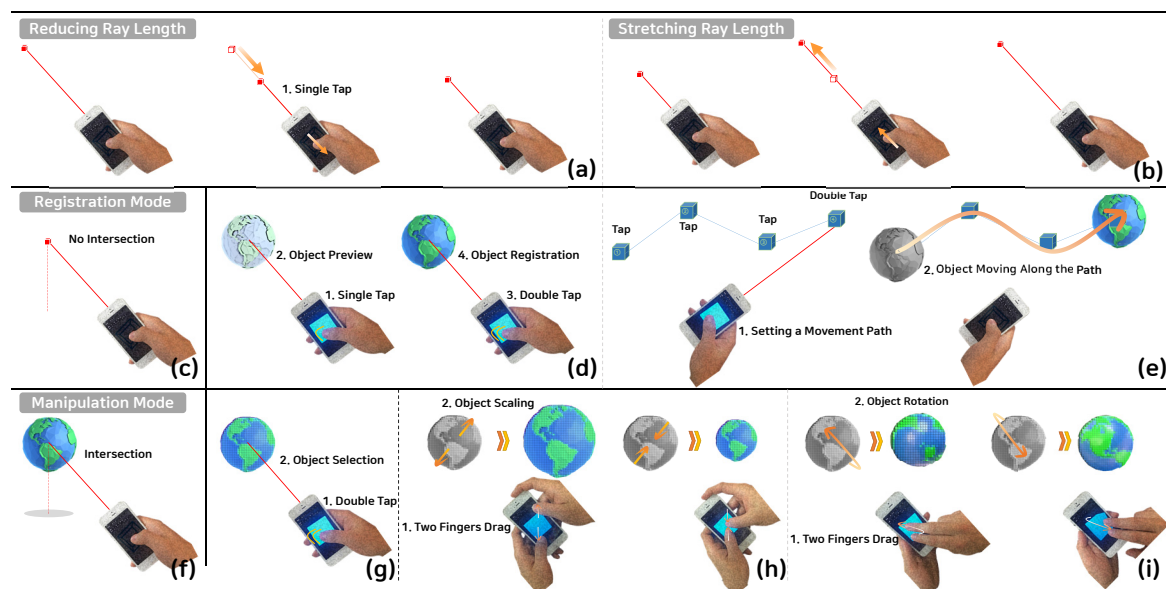
Figure 3. Object Manipulation. (**a**) Reducing ray length (depth). (**b**) Stretching ray length. (**c**) Registration mode (when nothing intersected). (**d**) Object registration. (**e**) Setting a movement path. (**f**) Manipulation mode (when object and ray's end point intersected. (**g**) Object selection. (**h**) Object Scaling. (**i**) Object Rotation.

3.3. Use-Cases

For existing AR contents and applications, a professional authoring tool must be used to register 3D virtual objects in the real world. Therefore, only advanced application programmers capable of handling an authoring tool (3D graphics software) can register contents, such as 3D virtual objects, in a specific location and users can just check the registered objects passively. As such, the interaction that can be experienced by ordinary users in AR is greatly limited. However, with the proposed AR Pointer, users can directly participate in registering and manipulating virtual objects, making it possible to create a wider variety of applications. We expect our proposed interface to be useful in 3D AR environments, including the registration and manipulation of 3D virtual objects. We also introduce some use-case examples where the proposed interface can be useful.

First, there are deployment scenarios for furniture and appliances. Because these products occupy a large area in the home, there are limited products that can be placed according to the free space. Also, the atmosphere of the room and the harmony of the furniture are important matters in the interior of the room. So if users can render these products and place them in the room in advance, they will save the effort of actually moving and rotating large and heavy objects. In addition, when you decorate your home, you will be able to expand they choices and improve their satisfaction by deployment virtual furniture. There are already a few applications that have implemented this scenario. For example, there is an IKEA furniture layout application [20] that includes 3D virtual object registration and manipulation scenarios. In this application, the catalog is used as a marker for mobile marker recognition and furniture is placed at the corresponding position by recognizing the marker. Therefore, the user must move the catalog to the desired position each time and it is difficult to place

several furniture items at the same time. In such scenarios, using the AR Pointer can overcome the inconvenience of existing applications. It was possible to compute and generate a ray from the AR Pointer according to the actual user position. In this application, users can place virtual furniture in the desired location, even where there are no markers. Also, the object manipulation methods introduced in Section 3.2 can be applied to enable interaction with virtual furniture.

The next scenario is the smart AR classroom. In this smart class, 3D virtual objects can be used to provide students with an immersive learning environment. Because the virtual objects are the multimedia materials that complement the contents of the class, the teacher should be able to freely and effectively register and manipulate each object for lecture and explanation. Among the STEM domain, AR education in spatial orientation skill [22,23] or chemistry can be a good example. A variety of studies have been conducted on various AR technologies to support the understanding of atomic and molecular structures [1,24]. However, these applications almost always use markers or computer vision-based feature point matching to augment atoms at the location. As a result, the space available to place the objects was limited to the desk and the additional interactions that the teacher or the students could perform were merely moving the markers closer to each other to make the atoms into a molecule. In this scenario, if the AR Pointer is used, the teacher can register and augment the atom to be described at the student's desk or the middle of the classroom. Then, for example, hydrogen and oxygen atoms can be moved closer to each other to illustrate the form of the H_2O molecular bond. They also can rotate the molecule object slowly to identify the three-dimensional structure of the molecule bond.

Another example is a scenario in astronomy class that describes and observes the movements and physical laws of the celestial bodies. To explain the orbits of the solar system and the orbital movement, the teacher can use the AR Pointer to register solar system objects in the middle of the classroom. Then, for example, to explain the rings of Saturn, it would be possible to select and scale up the object. In the situation where the teacher and students exchange questions and answers, this time, students can interact with these objects in the same way. At this time, instead of passing the interface used by the teacher to the students, the students' own mobile devices can be used as AR Pointers to participate in the object manipulation tasks.

4. Experiments 1

We conducted two experiments to evaluate the performance and usability of the proposed AR interface, AR Pointer. The first experiment is a comparison with the existing authoring tools. In this experiment, object manipulation using the authoring tools, such as OpenGL [25] and Unity Engine [20], which are used by programmers, was compared with object manipulation using our proposed AR Pointer. Then, we also tested whether there is any meaningful difference between the experimental results of users who are familiar with 3D AR environments (Users with 3D game/rendering/graphics programming experience) and those who are not. In the second experiment, we performed three interactions (Rotation, Scaling and Translation) using the GaS-assisted wireframe interface and our proposed AR Pointer in the HoloLens environment and compared the results in terms of task completion time, fatigue and usability.

4.1. Design

The evaluations of the 3D user interface focus on three main aspects [26]: object manipulation [21,27], viewpoint manipulation [28] and application control [29]. Since the AR Pointer is proposed for the efficient registration and manipulations of 3D virtual objects, our experiment was conducted with a focus on the first aspect only. Also, interviews were carried out about three categories: speed, intuition and ease of learning. A total of 26 participants participated in this experiment 1. One group was an experienced group of 3D graphics tools, aged between 25 and 33 years old (mean [M] + standard deviation [SD] = 28.2 + 2.2, male = 10, female = 3, one left-handed). The other group was an inexperienced group of 3D graphics tool, aged between 25 and 31 years old (mean [M] + standard

deviation [SD] = 27.8 + 1.9, male = 9, female = 4, all right-handed). Each group performed four tasks: registration, translation, scaling and rotation of 3D virtual objects. Also, each task was performed four times with proposed AR Pointer interactions and keyboard and mouse combinations, respectively. In this experiment, we implemented the furniture layout application prototype shown in Section 3.3 and used it for the experiment. Completion time was calculated according to how quickly the furniture was moved to the suggested location in the room and transformed into the correct scale and pose.

The subjects manipulated the objects using the interfaces of the Unity Engine [20]. They could register and manipulate the objects while continuing to view the actual surrounding environment through the PC screen. The objects could be manipulated using the mouse in the scene view or by entering the number through the keyboard in the inspector view. The basic usage of the authoring tool (Unity Engine) was briefly explained about 5 min before the experiment started. The experiment proceeded in two stages. First, a ground truth object is randomly generated in the Unity test scene. The subjects had to register the object at the corresponding location (registration experiment). In the next step, a guide object was created at a new location and the re-registration (translation) task is performed by selecting the object registered in the previous step and then manipulate it to orientation (rotation) and size (scaling). The first experiment was performed in the Unity Engine, which is the authoring tool, without wearing the HMD. When manipulating objects using the AR Pointer, subjects still had to check their work on a PC monitor. The first experiment was usually completed within 5 min.

4.2. Results

We found that, when using the proposed AR Pointer, the users achieved a faster completion time than the keyboard and mouse combinations represented by the existing 2D interface for all interaction methods. The participants familiar with the 3D AR environment did not notice a significant difference in completion time between the keyboard and mouse combinations and using AR Pointer. Certain manipulation methods even showed similar or superior completion times using the existing 3D graphics tool method. However, participants who were not familiar with 3D AR environments achieved significantly superior results when using the AR Pointer in shown as Figure 4.
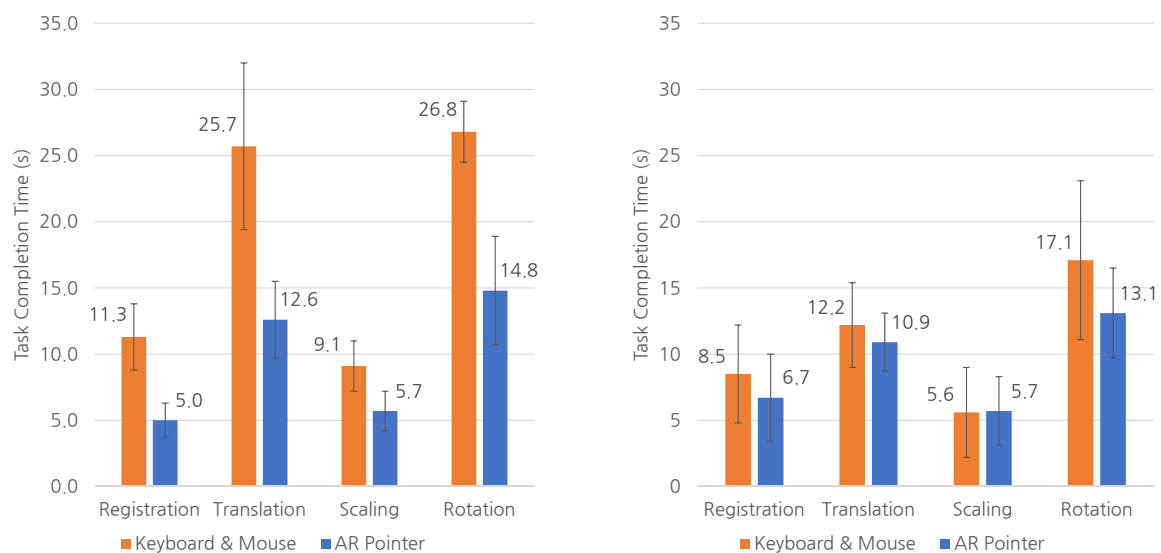

Figure 4. Result of experiment 1 about task completion time

4.3. Data Analysis and Discussion

We documented and analyzed the reactions and responses of the participants throughout the experiments and during the interviews conducted afterward. Participants who were not familiar with the 3D AR environment found it very difficult to understand the 3D space. This confusion continued

throughout the four experimental attempts and we noticed that the participants continued trying to identify where each axis was but with little success. Therefore, we conclude that the existing 3D graphics tool method is difficult to learn, as well as having a low level of intuitiveness for general users. Conversely, users familiar with the 3D AR environment are also familiar with the 3D graphics tool, so completion time does not differ significantly. Since only professionals capable of using the 3D graphics tool could register and manipulate the 3D AR application contents as desired, ordinary users were able to experience only limited, passive interaction with object manipulation. This results in limited contents and interaction with objects with various AR applications.

However, with the AR Pointer, a user could intuitively register the object in the pointing direction, without considering the x-, y- and z-axes, thereby reducing completion time. Also, the AR Pointer continued to show a shorter completion time than the keyboard and mouse combination, in which completion time did not change significantly even when attempts are repeated. Furthermore, while the keyboard and mouse combination did not significantly change the completion time, even with repeated attempts, the AR Pointer showed usability in ease of learning, as the completion time was continuously reduced.
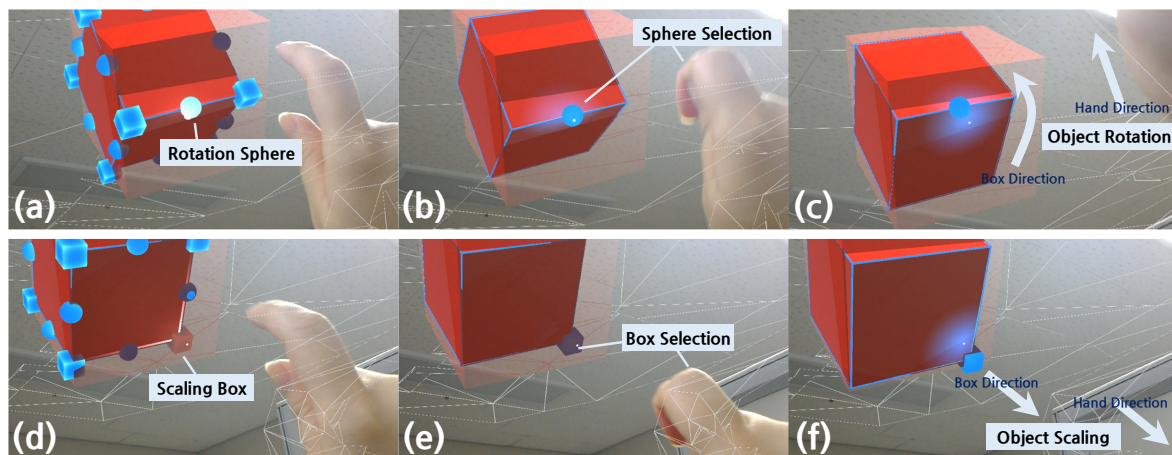
5. Experiment 2

In the second experiment, we experimented with docking tasks of 3D objects [18] in shown as Figure 5b,c. We also conducted a direct comparison with the actually used HoloLens interface and proposed AR Pointer, both a quantitative experiment (that compares the task completion time) and a subjective assessment (through a questionnaire completed after the task's sequence).
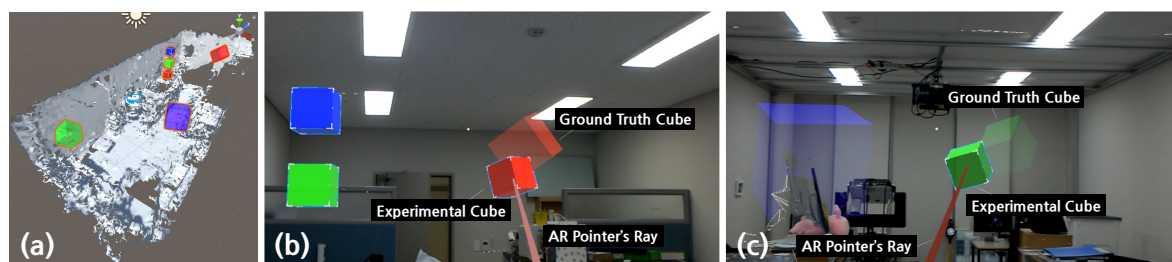
Figure 5. Experiment 2 test environment. (**a**) Demonstration room overview. (**b**,**c**) Docking tasks of 3D objects using AR Pointer.

5.1. Design

The experiment was conducted in a demonstration room, which was designed as a living room with a size of about $4 \times 3 \times 8$ m (width, height, depth) in shown as Figure 5a. Since we assume that HMD and AR Pointer are mainly used indoors, we cut off outdoor lighting with blinds and experimented in the lighting environment where only indoor fluorescent lamps are affected. The subjects consisted of 21 people ranging in age from 24 to 31 (mean [M] + standard deviation [SD] = 27.4 + 2.5, male = 15, female = 6) and one subject was left-handed. Eight of them had experience with head-mounted display HMD-based AR. Participants were briefed about the HoloLens and two methods and given time to practice these techniques. The duration of the exercise session was finished when the subjects were well-informed and those without HMD-based AR experience took somewhat longer (up to 10 min, about 5 min on average). In the case of the HoloLens interface, the default method in which the GaS and wireframe methods are used, was chosen for the second experiment in shown as Figure 6. The AR Pointer is used to perform the task through object manipulation method as proposed in Section 3.2.

Figure 6. Gaze-assisted-selection (GaS) + Wireframe Manipulation Method. (**a**–**c**) Rotation sphere can be selected to rotate objects. (**d**–**f**) Scaling Box can be selected to scale objects.

When the experiment was started, three dummy (ground truth) objects in the form of a cube was created in which the ground truth (GT) positions in specified positions in the room were placed. After a few seconds, experimental objects were created at another location and the user performed four tasks (selection, translation, rotation, scaling) so that the experimental objects overlaps with the GT object. The distances between the three sets of experimental objects and GT objects were set to 2.5, 4.5 and 6.5 m, respectively. These operations can be performed in the order chosen by the user. This task was performed six times by each user (2 interfaces × 3 target positions). Every time a task is completed (designated as the time when the users feel finished and the volume of unions > 90%), it measures the task completion time and the error rate between the two objects. The experiment took an average of 19.2 min per person.

After the task completion time experiment, we conducted two surveys to obtain a subjective assessment of the subjects. One is the system usability scale (SUS) [30] for comparing the usability of the two interfaces and the other is the NASA task load index (NASA-TLX) [31,32] for comparing the fatigue when using both interfaces. Both questionnaires are standard questionnaires frequently used in human-computer interaction (HCI) and are listed in Tables 1 and 2. The cronbach's alpha [33] value for questionnaires reliability is $\alpha > 0.9$ [34] for SUS and $\alpha > 0.7$ [35–37] for NASA-TLX. Cronbach's alpha $\alpha > 0.7$, it was accepted as a questionnaire [38]. In the survey using SUS qestionnaire, the users answered questions in ten categories about each AR interface. A 7-point Likert scale was used in this survey. In the case of this questionnaire, it is designed to alternate positive and negative questions in order to prevent accidental bias. Each question was answered with a score between 1 and 7 (indicating strongly disagree and strongly agree, respectively). We also conducted a short interview concerning subjective comments about these interfaces. The NASA-TLX survey consists of two parts. First of all, the subjects fill in the score of 5 point steps within the range of 100 points for 6 questions. Next, the subjects will group all six items together and conduct a 15-pair comparison. The subjects conducted a 15-pair comparison of the question "When designing the interface to be used in HoloLens, which of the two items should be considered more?". We also accumulated the selected responses to get the weighted scores. The weighted scores thus obtained was used as the workload weight when obtaining the total fatigue score from each NASA-TLX result.

Table 1. The System Usability Scale (SUS) Questionnaire.

	Question
1	I think that I would like to use this system frequently.
2	I found the system unnecessarily complex.
3	I thought the system was easy to use.
4	I think that I would need the support of a technical person to be able to use this system.
5	I found the various functions in this system were well integrated.
6	I thought there was too much inconsistency in this system.
7	I would imagine that most people would learn to use this system very quickly.
8	I found the system very cumbersome to use.
9	I felt very confident using the system.
10	I needed to learn a lot of things before I could get going with this system.

Table 2. The NASA Task Load Index (NASA-TLX) Questionnaire.

Title	Question
Mental Demand	How much mental and perceptual activity was required?
Physical Demand	How much physical activity was required?
Temporal Demand	How much time pressure did you feel due to the pace at which the tasks or task elements occurred?
Overall Performance	How successful were you in performing the task?
Effort	How hard did you have to work (mentally and physically) to accomplish your level of performance?
Frustration Level	How irritated, stressed, and annoyed did you feel during the task?

5.2. Results

Since our experiments were designed for within-subjects, the same group used both AR interfaces. A repeated measures two-way analysis of variance (ANOVA) was used for analysis. The *p*-value for determining statistical significance was set at 0.05. First, Figure 7 shows the experimental results of how quickly the user completed a given task. We examined Mauchly's test of sphericity as a premise of repeated measures ANOVA. The Mauchly's W value and its alternatively available Greenhouse-Geisser and Huynh-Feldt values were all set at an ideal value of 1 and consequently the sphericity assumption was satisfied. Based on these results, we confirmed that F (1, 20) = 17.581 and $p < 0.001$. In other words, when they performed three tasks according to the distance, we found that all completion times showed a significant difference. Compared to the conventional HoloLens interface GaS and wireframe methods, the AR Pointer showed a 39.4% faster performance on average compared to the average required time. On the other hand, we found that the within-subjects effect for the distance and distance * methods yielded F (2, 40) = 0.352, $p = 0.705$ and F (2, 40) = 0.682 and $p = 0.512$, respectively, with no statistical significance between the two measurements. In other words, although the distances to be moved by the set of cubes were different, the two interfaces did not show a significant distance-based difference. This is because the time it takes to move the object is less than the time it takes to manipulate the rotation and scale. The times spent on each manipulation were not measured separately but were clearly confirmed as the experiment progressed. (The reason why we could not measure the time spent for each operation was that the manipulation sequence for completing the task was different for each subject. For example, some subjects work in the order of translation, rotation and scaling, while others work in the order of translation, scaling, rotation, translation and scaling.)

The results of the SUS usability experiment are shown in Figure 8. For SUS surveys, both positive and negative questions are mixed. Therefore, when calculating the total score, the low score of the negative question was unified so that it was calculated as a high score in the total score. The final score was 100 points. The final score obtained by combining all the items was 55.56 points using the GaS + Wireframe method and 64.17 points using the AR Pointer method. The items that showed a big difference were Q3, Q5 and Q9. In the three questions, the AR Pointer scores were higher,

1.83, 1.50 and 1.50, respectively. The results of the NASA-TLX fatigue experiment are shown in Figure 9. NASA-TLX questions are designed to give higher scores for all negative, except for the fifth, performance. Therefore, for the consistency of the graph, the score obtained from the performance is inverted and displayed on the graph. The weighted score for each item was obtained through a pairwise comparison response corresponding to the second step ($weight_{metal}$ = 0.178, $weight_{physical}$ = 0.156, $weight_{temporal}$ = 0.089, $weight_{effort}$ = 0.156, $weight_{performance}$ = 0.256, $weight_{frustration}$ = 0.167). The mean score of NASA-TLX obtained from the first step survey results was M = 59.31 (SD = 20.48) for the GaS + Wireframe method and M = 50.69 (SD = 19.62) for the AR Pointer method. The final total average score of the weighted score was M = 56.53 (SD = 15.72) in the GaS + Wireframe method and M = 48.02 (SD = 11.94) in the AR Pointer.

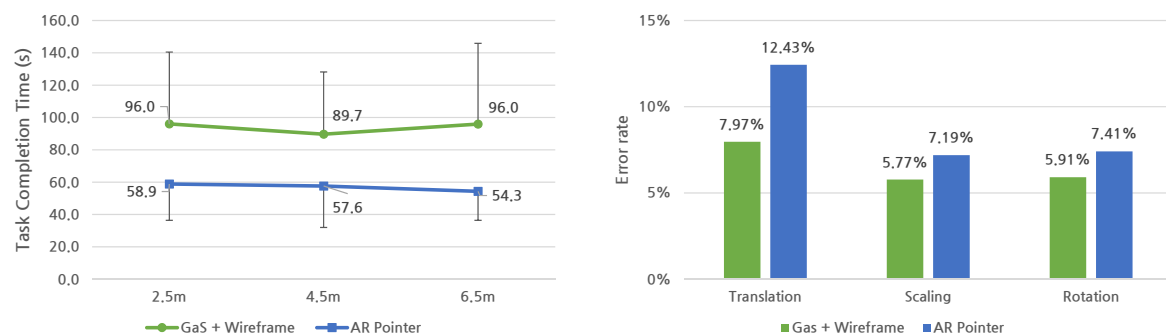**Figure 7.** Result of experiment 2. (**left**) task completion time. (**right**) error rate.
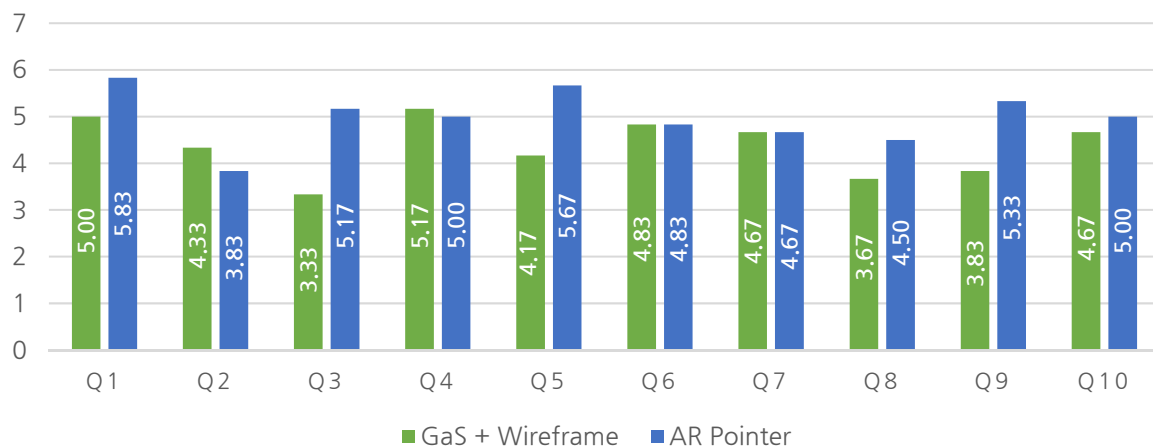

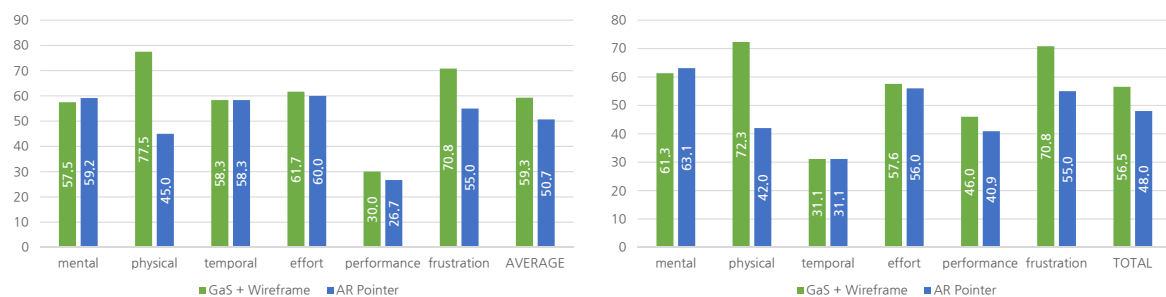**Figure 8.** system usability scale (SUS) questionnaire survey result.

Figure 9. NASA task load index (NASA-TLX) questionnaire survey result.

5.3. Data Analysis and Discussion

There are several reasons why the AR Pointer showed better results with respect to the time of task completion. First, when using the AR Pointer, it is possible to move the object remotely through the gesture of the mobile device without the need for the subject to directly move the object. Subjects move the objects around to the GT cube every time they used the GaS interface but they moved very little when using the AR Pointer. Next, selecting the wireframe editing box for rotation and scaling is a time-consuming task. Each editing box is very small compared to the object and it is more difficult to select such a small object in the GaS method in which the user rotates the face to position the cursor. There is also a problem with the clicking method using the tip of the finger. The subjects were confused about how much they had to bend their fingers to get a click action and conversely, how much they had to stretch their fingers to produce a release action. As a result, frequent click mistakes can occur. On the other hand, the AR Pointer is free from this problem because it uses a touch screen.

Figure 7 shows the error rate between the GT and experimental cubes with respect to how well the subjects manipulated and finally registered the object. In this experiment, position, rotation and scale differences were calculated for each cube set. The position difference is expressed with a Euclidean distance. Experimental results show that a distance error of 7.97 cm occurs on average when using the GaS + Wireframe method and a distance error of 12.43 cm on average when using the AR Pointer method. The proposed method has higher error rate than the wireframe method of the HoloLens. Considering that the size of the cube used as the ground truth was $53.4 \times 53.4 \times 53.4$ cm, each has an error of 8.61% (about 4.49 cm on one side) and 13.44% (about 7.95 cm on the other side) compared to one side of the cube. There are two reasons why the accuracy of the AR Pointer is lower than the existing interface. First, it is the problem of precision of mobile gestures for depth control. We set the modulation sensitivity of the ray to the pixel of the mobile touch screen and set the ray to change 1 mm when moving one pixel through the gesture. The Apple iPhone 8 used in the experiment had 326 pixels per inch (PPI), so it had about 128.34 pixels per cm (PPCM) in terms of metric units. A 1 cm displacement swipe gesture would detect approximately 128 pixels on the touch screen, resulting in a 12.8 cm ray change in a holographic 3D environment. Therefore, it was difficult to control the precise ray in the current structure. In order to solve this problem, it was possible to lower the ray control coefficient for the pixel but it would have lead to a trade-off relationship with the completion of the fast task. A follow-up study would be needed to determine the ray control factor in order to ensure accuracy and speed. Second, as remote manipulation became easier, subjects were no longer near the GT cube. When moving an object away from the individual using the GaS interface, there are two ways to move: (1) the object is moved to the remote position while maintaining the object selection or (2) the object is released after the object is selected by extending the maximum arm, moving the object a certain distance and repeating the operation. Since most of the subjects chose the former method, they could see the area of the GT cube nearby and could manipulate the object more precisely when operated at a distance. On the other hand, in the case of using the AR Pointer, there was no case in which a user moved closer to an object and manipulated the object. Rather, the subjects tried to keep the AR Pointer's distance within a certain length so that the ray of the AR Pointer could be seen clearly in the HoloLens field of view (FoV). For the same reason, the AR Pointer had a relatively high error rate in the scaling interaction. The AR Pointer had a scaling error rate of 7.19% and an error rate of 5.77% in the conventional interface. The AR Pointer showed a higher error rate for object rotation angle error. For this reason, the same reason that caused the error of the position Euclidean distance above occurs and additionally, there is a problem in the rotation interaction method of the AR Pointer. First, subjects were unfamiliar with the arcball rotation method. Although it is a rotation method that is commonly used in mobile devices, it was felt that it was an unfamiliar interaction when manipulating an object identified through HMD while giving a feeling of direct manipulation when rotating an object in a mobile screen. Another problem is the limitation of the arcball rotation, indicating that it is not possible to directly control rotation about the z-axis in the 3D environment. For rotation in the z-axis, it is necessary to rotate the combination of the x- and y-axes or the user must

move in the other direction. In the case of using the AR Pointer, the users naturally minimize their movement and eventually the object's rotation interaction becomes more difficult.

The SUS usability test for subjective evaluation of each interface is shown in Figure 8. The AR Pointer proposed in this experiment was similar or better than the existing GaS + Wireframe interface method with respect to nine test items, especially the higher pointer scores for Q1 (M = 5.83), Q5 (M = 5.67) and Q9 (M = 5.33). Another notable point is that the Q3, Q5 and Q9 questions showed a big difference of 1.83, 1.50 and 1.50, respectively. Each of the questions was about easy to use, functions integration and confidence of the interface and the results showed that the proposed method of manipulating the AR Pointer was intuitive and easy to use. On the other hand, post-interviews conducted after the survey showed that the rotation of the AR Pointer was somewhat difficult. Since the current rotation interaction method has been a problem even with respect to the error rate, it is necessary to improve the interaction method so that it can be used more easily by users rather than restricting them to the existing mobile gesture interaction method. The NASA-TLX questionnaire conducted to determine the fatigue level of each interface is shown in Figure 9. Also, looking at the weighted score obtained through relative comparison in Table 3, performance of interface was the most important design consideration in the HMD AR environment when the participants considered it. The question in the Performance item was how well the given object was moved, rotated and scaled to the specified position. Although the difference from the GaS + Wireframe was not large, it proved that the highest score among the evaluation items was achieved and that the AR Pointer can achieve the best performance. The overall questionnaire results showed a difference of 8.61 before weighted score correction and 8.51 after correction, so we could see that the fatigue problem can be overcome by using AR Pointer in AR HMD environment. It is not intuitive, which is the biggest problem of the existing HoloLens interface and it is a very good result because we proposed our interface to solve the problems that cause user fatigue. The two major problems with the existing HoloLens interface are not being intuitive and causing user fatigue. Since we proposed the AR Pointer to solve these problems, the obtained usability test results have been encouraging. In particular, the physical demand item showed an average difference of 32.50, which indicates that the interface using the existing hand gesture can be a cause of great fatigue in the HMD environment. In an actual experiment, when using the GaS + wireframe interface, the subjects complained of frequent pain in the arm and this was the reason for the increase in completion time.

Table 3. The weighted scores from NASA-TLX survey result.

$weight_{metal} = 0.178$	$weight_{physical} = 0.156$	$weight_{temporal} = 0.089$
$weight_{effort} = 0.156$	$weight_{performance} = 0.256$	$weight_{frustration} = 0.167$

6. Conclusions

In this paper, we proposed a new AR interface, AR Pointer, which can conveniently register 3D virtual objects and manipulate registered objects in various places in 3D AR environment. The proposed AR Pointer is implemented using a 6-DoF IMU sensor and Ray-casting technology and is designed to manipulate virtual objects through touch gestures. Conventional AR interfaces are limited in remote object registration. On the contrary, we can control the length of the ray through the swipe gesture to specify a remote point and conveniently register objects in the point. In addition, since the AR Pointer adapts the same/similar gestures commonly used in smartphones for object manipulation, general users who are not familiar with the AR environment can easily use the gestures without prior learning. To demonstrate the high performance and usability of the AR Pointer, we performed two experiments on object manipulation. First, in comparison with the existing authoring tool interface, it has been proven that even a general user who has no programming experience can also easily author the AR environment by manipulating the object using the AR Pointer. Second, we measured time to complete tasks for object manipulation with AR Pointer and other AR interfaces. The results show that AR Pointer is able to complete the given tasks on average 39.4% faster than other AR interfaces

do. In addition, the participants gave an average of 8.61 points (13.4%) on the AR Pointer in the usability test conducted through the SUS questionnaires and 8.51 points (15.1%) on the AR Pointer in the fatigue test conducted through the NASA-TLX questionnaire. In summary, AR Pointer shows good results both in object manipulation performance and usability. It also has the advantage of registering an object at a specific mid-air point in three dimensions. Therefore, it will be useful for manipulating objects in various 3D AR applications to be released later.

Author Contributions: Conceptualization, H.R.; Methodology, H.R. and J.-H.B.; Software, H.R.; Validation, H.R., J.-H.B. and Y.J.P.; Formal Analysis, H.R. and J.-H.B.; Investigation, H.R. and Y.J.P.; Data Curation, H.R.; Writing—Original Draft Preparation, H.R. and N.K.L.; Writing—Review & Editing, H.R., J.-H.B., Y.J.P., and N.K.L.; Visualization, H.R. and Y.J.P.; Supervision, N.K.L. and T.-D.H.; Project Administration, T.-D.H.; Funding Acquisition, T.-D.H.

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

References

1. Dáskalos Chemistry. 2019. Available online: http://prefrontalcortex.de/ (accessed on 30 July 2019).
2. Ro, H.; Park, Y.J.; Byun, J.H.; Han, T.D. Mobile device interaction using projector metaphor. In Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion, Marina del Ray, CA, USA, 17–20 March 2019; ACM: New York, NY, USA, 2019; pp. 47–48.
3. Yu, J.; Jeon, J.; Park, J.; Park, G.; Kim, H.I.; Woo, W. Geometry-aware Interactive AR Authoring using a Smartphone in a wearable AR Environment. In Proceedings of the International Conference on Distributed, Ambient, and Pervasive Interactions, Vancouver, BC, Canada, 9–14 July 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 416–424.
4. MacKenzie, I.S. Input devices and interaction techniques for advanced computing. In *Virtual Environments and Advanced Interface Design*; Oxford University Press: Oxford, UK, 1995; pp. 437–470.
5. Ro, H.; Chae, S.; Kim, I.; Byun, J.; Yang, Y.; Park, Y.; Han, T. A dynamic depth-variable ray-casting interface for object manipulation in ar environments. In Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 5–8 October 2017; pp. 2873–2878.
6. Bellarbi, A.; Zenati, N.; Otmane, S.; Belghit, H.; Benbelkacem, S.; Messaci, A.; Hamidia, M. A 3D interaction technique for selection and manipulation distant objects in augmented reality. In Proceedings of the 2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B), Boumerdes, Algeria, 29–31 October 2017; pp. 1–5.
7. Chaconas, N.; Höllerer, T. An Evaluation of Bimanual Gestures on the Microsoft HoloLens. In Proceedings of the 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Reutlingen, Germany, 18–22 March 2018; pp. 1–8.
8. Soh, J.; Choi, Y.; Park, Y.; Yang, H.S. User-friendly 3D object manipulation gesture using Kinect. In Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry, Hong Kong, China, 17–19 November 2013; pp. 231–234.
9. Coelho, J.C.; Verbeek, F.J. Pointing task evaluation of leap motion controller in 3D virtual environment. *Creat. Diff.* **2014**, *78*, 78–85.
10. Periverzov, F.; Ilieş, H. IDS: The intent driven selection method for natural user interfaces. In Proceedings of the 2015 IEEE symposium on 3D user interfaces (3DUI), Arles, France, 23–24 March 2015; pp. 121–128.
11. Sharp, T.; Keskin, C.; Robertson, D.; Taylor, J.; Shotton, J.; Kim, D.; Rhemann, C.; Leichter, I.; Vinnikov, A.; Wei, Y.; et al. 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; pp. 3633–3642.
12. Kim, H.I.; Woo, W. Smartwatch-assisted robust 6-DOF hand tracker for object manipulation in HMD-based augmented reality. In Proceedings of the 2016 IEEE Symposium on 3D User Interfaces (3DUI), Greenville, SC, USA, 19–20 March 2016; pp. 251–252.

13. Jung, W.; Woo, W.T. Duplication Based Distance-Free Freehand Virtual Object Manipulation. In Proceedings of the 2017 International Symposium on Ubiquitous Virtual Reality (ISUVR), Nara, Japan, 27–29 June 2017; pp. 10–13.

14. Jung, W.; Cho, W.; Kim, H.; Woo, W. BoostHand: Distance-free Object Manipulation System with Switchable Non-linear Mapping for Augmented Reality Classrooms. In Proceedings of the 2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), Nantes, France, 9–13 October 2017; pp. 321–325.

15. 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.

16. Baloup, M.; Pietrzak, T.; Casiez, G. RayCursor: A 3D Pointing Facilitation Technique based on Raycasting. In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2019), Scotland, UK, 4–9 May 2019.

17. Hincapié-Ramos, J.D.; Ozacar, K.; Irani, P.P.; Kitamura, Y. GyroWand: IMU-based raycasting for augmented reality head-mounted displays. In Proceedings of the 3rd ACM Symposium on Spatial User Interaction, Los Angeles, CA, USA, 8–9 August 2015; pp. 89–98.

18. Ha, T.; Feiner, S.; Woo, W. WeARHand: Head-worn, RGB-D camera-based, bare-hand user interface with visually enhanced depth perception. In Proceedings of the 2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Munich, Germany, 10–12 September 2014; pp. 219–228.

19. Fender, A.; Lindlbauer, D.; Herholz, P.; Alexa, M.; Müller, J. HeatSpace: Automatic Placement of Displays by Empirical Analysis of User Behavior. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, Québec City, QC, Canada, 22–25 October 2017; pp. 611–621.

20. Unity Engine. 2019. Available online: https://unity.com/ (accessed on 30 July 2019).

21. Grandi, J.G.; Berndt, I.; Debarba, H.G.; Nedel, L.; Maciel, A. Collaborative 3D manipulation using mobile phones. In Proceedings of the 2016 IEEE Symposium on 3D User Interfaces (3DUI), Greenville, SC, USA, 19–20 March 2016; pp. 279–280.

22. Carbonell Carrera, C.; Bermejo Asensio, L.A. Augmented reality as a digital teaching environment to develop spatial thinking. *Cartogr. Geogr. Inf. Sci.* **2017**, *44*, 259–270.

23. Carbonell Carrera, C.; Bermejo Asensio, L.A. Landscape interpretation with augmented reality and maps to improve spatial orientation skill. *J. Geogr. High. Educ.* **2017**, *41*, 119–133.

24. Nechypurenko, P.P.; Starova, T.V.; Selivanova, T.V.; Tomilina, A.O.; Uchitel, A. Use of Augmented Reality in Chemistry Education. In Proceedings of the 1st International Workshop on Augmented Reality in Education, Kryvyi Rih, Ukraine, 2 October 2018; pp. 15–23.

25. OpenGL. 2019. Available online: https://www.opengl.org/ (accessed on 30 July 2019).

26. Hand, C. A survey of 3D interaction techniques. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 1997; Volume 16, pp. 269–281.

27. Liu, J.; Au, O.K.C.; Fu, H.; Tai, C.L. Two-finger gestures for 6DOF manipulation of 3D objects. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2012; Volume 31, pp. 2047–2055.

28. Rakita, D.; Mutlu, B.; Gleicher, M. Remote Telemanipulation with Adapting Viewpoints in Visually Complex Environments. In Proceedings of the Robotics: Science and Systems 2019, Freiburg im Breisgau, Germany, 22–26 June 2019.

29. Chittaro, L.; Sioni, R. Selecting Menu Items in Mobile Head-Mounted Displays: Effects of Selection Technique and Active Area. *Int. J. Hum.. Comput. Interact.* **2018**, 1–16, doi:10.1080/10447318.2018.1541546.

30. Brooke, J. SUS-A quick and dirty usability scale. *Usability Eval. Ind.* **1996**, *189*, 4–7.

31. Hart, S.G.; Staveland, L.E. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in Psychology*; Elsevier: Amsterdam, The Netherlands, 1988; Volume 52, pp. 139–183.

32. Hart, S.G. NASA-task load index (NASA-TLX); 20 years later. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, San Francisco, CA, USA, 16–20 October 2006; Sage Publications: Los Angeles, CA, USA, 2006; Volume 50, pp. 904–908.

33. Cronbach, L.J. Coefficient alpha and the internal structure of tests. *psychometrika* **1951**, *16*, 297–334.

34. Sauro, J. *Measuring the Quality of the Website User Experience*; University of Denver: Denver, CO, USA, 2016.

35. Longo, L. *On the Reliability, Validity and Sensitivity of Three Mental Workload Assessment Techniques for the Evaluation of Instructional Designs: A Case Study in A Third-Level Course*; Technological University Dublin: Dublin, Ireland, 2018.

36. Ahram, T.; Jang, R. Advances in Physical Ergonomics and Human Factors: Part II. In Proceedings of the 5th International Conference on Applied Human Factors and Ergonomics 20 Volume Set, Kraków, Poland, 19–23 July 2014.

37. Adnan, W.A.W.; Noor, N.L.M.; Daud, N.G.N. Examining Individual Differences Effects: An Experimental Approach. In Proceedings of the International Conference on Human Centered Design, San Diego, CA, USA, 19–24 July 2009; pp. 570–575.

38. Knapp, T.R. Focus on psychometrics. Coefficient alpha: Conceptualizations and anomalies. *Res. Nurs. Health* **1991**, *14*, 457–460.