BoostHand : Distance-free Object Manipulation System with Switchable Non-linear Mapping for Augmented Reality Classrooms

Whie Jung* Woojin Cho† Hayun Kim‡ Woontack Woo§

KAIST UVR Lab.

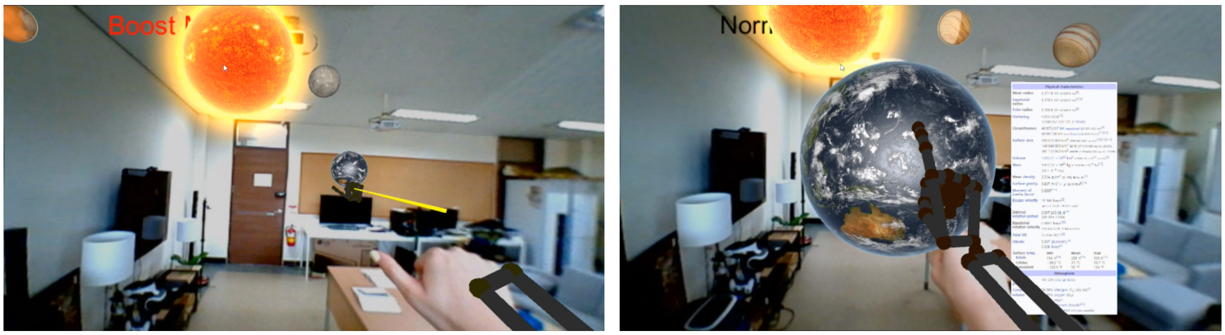


Figure 1: AR class using BoostHand: A lecturer plans to teach the solar system and planets to students, and can select and render virtual teaching materials in a shared, augmented space.

Abstract

In this paper, we propose BoostHand, a freehand, distance-free object-manipulation system that supports simple trigger gestures using Leap Motion. In AR classrooms, it is necessary to allow both lecturers and students to utilize virtual teaching materials without any spatial restrictions, while handling virtual objects easily, regardless of distance. To provide efficient and accurate methods of handling AR classroom objects, our system requires only simple intuitive freehand gestures to control the users virtual hands in an enlarged, shared control space of users. We modified the GoGo interaction technique [5] by adding simple trigger gestures, and we evaluated performance against gaze-assisted selection (GaS) capabilities. Our proposed system enables both lecturers and students to utilize virtual teaching materials easily from their remote positions.

Index Terms: H.5 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; H.5 [Information Interfaces and Presentation]: User Interfaces—Evaluation/methodology

1 Introduction

In contrast to real-world classrooms, it is strongly advantageous that AR classroom users work from their own spaces without any spatial restrictions. To fully implement this advantage, it is essential to provide users the ability to reach anywhere into their AR space. In the case of an AR classroom, the lecturer should be able to freely control the teaching materials, even those at otherwise unreachable distances, to maximize lecture efficiency. If both lecturers and students can access all materials floating around the AR classroom, many more virtual reality (VR) and AR capabilities can be produced. Therefore, a remote object-manipulation system is a prerequisite for an AR classroom.

*e-mail : whieya@kaist.ac.kr
†e-mail : dnwls2416@kaist.ac.kr
‡e-mail : hayunkim@kaist.ac.kr
§e-mail : wwoo@kaist.ac.kr

So far, most remote object-manipulation systems have required additional controllers and secondary devices. However, in an AR classroom, freehand manipulation systems are essential to support the lecturer's robust manipulation of AR and VR content, which can lead to higher lecture quality. Therefore, we propose our intuitive freehand, distance-free object-manipulation system.

Our proposed system modifies the GoGo interaction system into one that is switchable using simple trigger gestures. The GoGo interaction technique provides an intuitive method of handling virtual objects using a virtual hand avatar; but it uses a fixed mapping function. We instead employ simple trigger gestures that enable users to easily switch between normal and boost modes. Boost mode enables amplified movement of the virtual hand avatar. Target users of the system are people who need to handle virtual objects in a limited space, for instance: an AR classroom.

The biggest advantage of our system is that the user's control space, which is usually bounded to physical arm length, can be largely expanded to cover additional space by simply leveraging widely available technologies, such as Leap Motion. The detection range of Leap Motion is around 600 mm, but that is enough to cover the entire AR space in our system. Moreover, our system emphasizes simple and intuitive manipulation, where the virtual hand avatar functions just like a real hand, except that the virtual hand avatar can be sent far into the distance. Anyone can utilize this system without inconvenient or complex instructions. Users just move their hands, and the motions are directly applied to the virtual hand avatar, wherever it is. The system does not require any additional devices or controllers, just simple trigger gestures for handling the virtual objects.

We evaluated our system against the Gaze-assisted Selection (GaS) technology, which is commonly used for AR/VR systems. We conducted quantitative and qualitative analyses using task-based evaluations and post-experiment interviews. We show that our proposed system provides better performance at long distances, even while avoiding virtual classroom obstacles, which typically increase completion time and distance errors. The post-experiment interviews show that our proposed system is overall more convenient for controlling virtual objects.

Section 2 of this paper summarizes related work and Section 3 describes our methods in detail. Section 4 then presents quantitative and qualitative evaluations of our proposed system. In Section 5, we discuss the results of the evaluations and a possible user experience (UX) scenario for an AR classroom using our system. Finally, Section 6 presents our conclusions and recommended future work.

2 Related Work

Designing an effective 3D object selection technique for virtual environments is essential for distance-free virtual object-manipulation. Various systems have been suggested to solve this challenging object-selection challenge [1]. Selection techniques can be classified per their selection tools. Examples include virtual ray or 3D cursors (e.g., virtual hand avatars), and primitives (e.g., box and sphere).

Mark R. Mine [4] proposed an at-a-distance system using rays projected from a user's hand, providing a simple way of selecting remote virtual objects. However, this method cannot guarantee precision, since the ray tip is greatly displaced at long distances, even with a miniscule hand motion.

Volumetric selection is an inspiring virtual object-manipulation capability. For example, TunnelSlice [3] allows user to naturally acquire a subspace in an augmented scene from a first-person perspective, and it shows stability compared to existing volumetric selection systems in wearable AR. It also performs well in dense and occluded environments. It undeniably enables accurate 3D selection and possesses strengths for multiple selections, but it requires more time and complexity for single object-manipulation. It also presents difficulties as a general methodology for general cases, like an AR classroom.

Systems such as WeARHand [2] focuses on object-manipulation and interaction, instead of 3D selection. It allows users to manipulate virtual 3D objects with their bare hands (i.e., no hand-mounted tracking devices) by recognizing pre-defined gestures. It extracts the hands' 3D features and renders a virtual hand in local reference coordinates of the AR space. The system, however, focuses on manipulations in near-space. Therefore, the users' control space is limited to the recognition area of the depth camera.

GoGo interaction, which was suggested by Poupyrev [5], is a more intuitive way of freehand manipulation. It inspired our idea of changing one's arm length at will. 3D displacement of a virtual hand avatar is non-linearly amplified by that of the real hand so that user can easily reach otherwise unreachable areas. The system, however, uses a fixed mapping function that can possibly lead to difficulties in precise manipulation of objects in the far distance. Like the ray casting-based system, the movement of the virtual hand avatar in the far distance can be too amplified to handle delicate tasks. We therefore propose a switchable GoGo interaction system using simple hand-trigger gestures. Thus, users can handle virtual objects using either normal mode or boost mode.

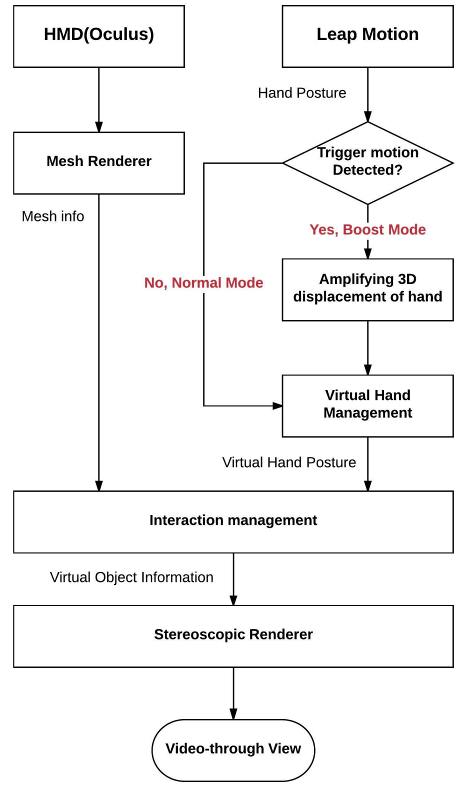


Figure 2: Overall system process flow

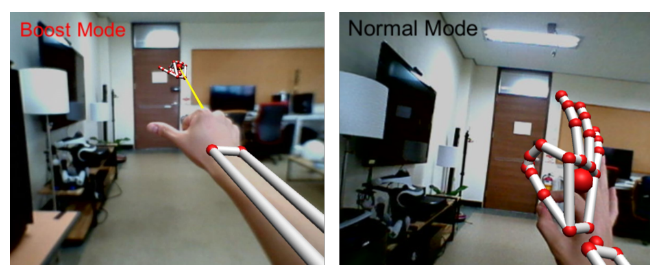


Figure 3: Designed gestures

3 Proposed System

3.1 System Design

The proposed system focuses on modifying GoGo interactions with simple trigger gestures allowing users to easily manipulate otherwise unreachable virtual objects using virtual hand avatars. The high-level system process is shown in Figure 2.

Leap Motion is a system that first tracks hand positions and trigger gestures using a detection module. The module then determines the displacement of the virtual hand using the real hand's trigger gesture (e.g., normal or boost). After this virtual hand management is complete, interaction between virtual hands and virtual objects are then managed and combined into a video-see-through view, which provides a 3D perspective via a stereoscopic renderer.

Simple trigger gestures allow switching between normal mode and boost mode in a head-mounted display (HMD) environment. We designed two trigger gestures, but were somewhat limited because of some tracking errors that occur in Leap Motion. Even though Leap Motion shows moderately good performance in hand-tracking, it throws errors when the hand rotates or when the fingers get too close together. Therefore, we attempted to find the simplest gesture with the most precise level of detection for our test. Note that a trigger motion should be a gesture that is not common to daily hand use, so that unintentional switching of modes does not occur.

As shown in Figure 3, we programmed boost mode to correlate to raising the thumb while the rest of fingers maintain a fist (i.e., 'thumbs-up'). Whenever a user makes this gesture, boost mode is activated, resulting in amplified virtual hand-movement. Otherwise, normal mode is the default setting.

The 'OK' hand gesture (Figure 3) is designed to reset the virtual hands position. When the 'OK' hand gesture is detected, 3D displacement of the virtual hand returns to default, and is immediately mapped to the real hand's position. This functionality is most helpful when users lose track of their virtual hands, or when they wish to return their virtual hands to their near-front proximity after manipulating virtual objects in the far distance.

3.2 Virtual Hand Management

As stated previously, virtual hand management is divided into two operational modes: normal and boost. In normal mode, the rotation and displacement of the virtual hands overlay the user's actual hands. The virtual hands move just as much as the real hands move. In short, a user's hand movements are mapped one-to-one to the virtual hand in normal mode. Actual hands and virtual hands, however, do not need to be co-located; they can be projected away from the user's real hands.

In boost mode, rotation of the virtual hand behaves the same as the real hand, whereas 3D displacement of the virtual hand is amplified. If the actual hand moves 5 cm in a forward direction, then the virtual hand will move a nonlinear amplified distance in boost mode. Boost mode, therefore, extends the users' control space, allowing them to easily manipulate otherwise unreachable virtual objects.

Determining the proper amplification function is an important requirement. Implementing a simple linear function does not work well [5], because if the value of the function is too low, users will face difficulties reaching virtual objects in far distance. Users will need to repeat their reaching actions to iteratively arrive at the distant virtual objects. In contrast, if the value of the function is too large, users can easily reach virtual objects in far distance, but cannot precisely control their virtual hand, since it will travel with exaggerated or unexpected acceleration. Thus, the amplification function should be carefully considered. Consequently, we chose a quadratic function for the best performance, shown as Equation 1.

$$g(x) = \begin{cases} \alpha X + \beta X^2 & if\ Boost\ mode \\ X & if\ Normal\ mode \end{cases} \tag{1}$$

Higher dimensions of this function create an over-amplified result, which can lead to the difficulties described above. We need our system to cover the entire range of an AR classroom, whose range is around 6 m. We therefore manually coded parameters for the function ($\alpha = 3, \beta = 5$), because the maximum range of hand movement is generally up to 0.8 m from the body.

4 Evaluation

4.1 Implementation

We implemented our system using the Oculus Rift Development Kit 2, with an Ovrvision Stereoscopic RGB camera. We also used Leap Motion for hand tracking. We mounted the system on the HMD so that the real hand can be tracked whenever it is within the HMD's field-of-view. The system was developed on a Unity 3D engine, and for our prototype application, we used an open-source Hololens application called 'Galaxy-Explorer.'

We chose GaS for comparison, because it is generally used as a freehand interface in AR environments, such as Hololens (Figure 4). We set up GaS with a user interface similar to HoloLens. Gas' main feature enables users to select and manipulate virtual objects with a gaze-enabled cursor while performing a clicking gesture with the free hand. For a fair comparison, we implemented GaS with Oculus Rift DK 2, just as we did our proposed system.

4.2 Task-based Evaluation

To evaluate our work, we designed a set of 3D docking tasks (Figure 5) during which users must move virtual 3D objects from a starting position to a target position in a specified virtual space. We designed six tasks with ground-truth positions, shown in Table 1, arranged along iterative distances with obstacles.

The independent variables in the experiment are the BoostHand and the GaS techniques, which were repeated twice. We performed a total of 24 trials (2 techniques x 6 tasks x 2 sets). The dependent variables are the distance error and completion time. The distance error is the Euclidian distance between the virtual hand's release position and the target position in the virtual bounding area.

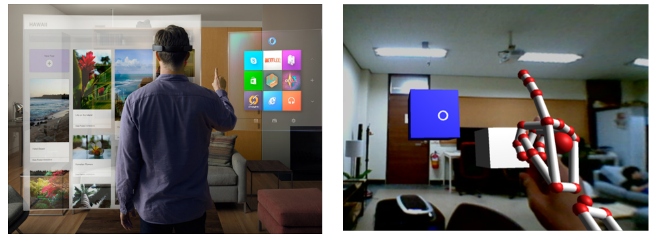


Figure 4: Gaze-assisted Selection(GaS)

Additionally, we surveyed participants' subjective ratings of accuracy (i.e., "How accurately did you complete the task?"), speed (i.e., "How fast did you complete the task?"), easiness (i.e., "How easy was it to complete the task?"), and perception (i.e., "How visually understandable was it to complete the task?"). Each rating used a five-point Likert scale, where '1' was most negative and '5' was most positive.

For the docking tasks, each user was required to grasp a sky-blue target object by its center, position it inside a green target position, and then release it. The target object turns red during a target position match condition so that the user can easily determine an appropriate release opportunity.

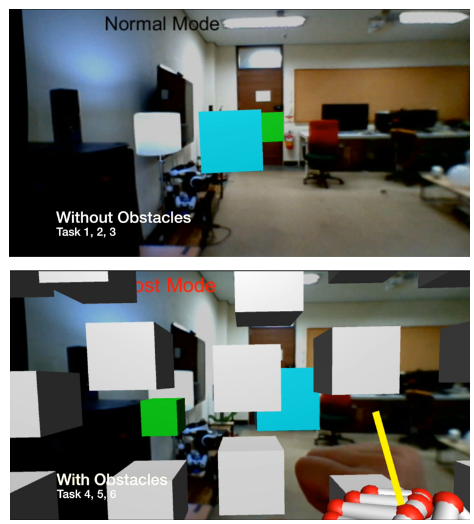


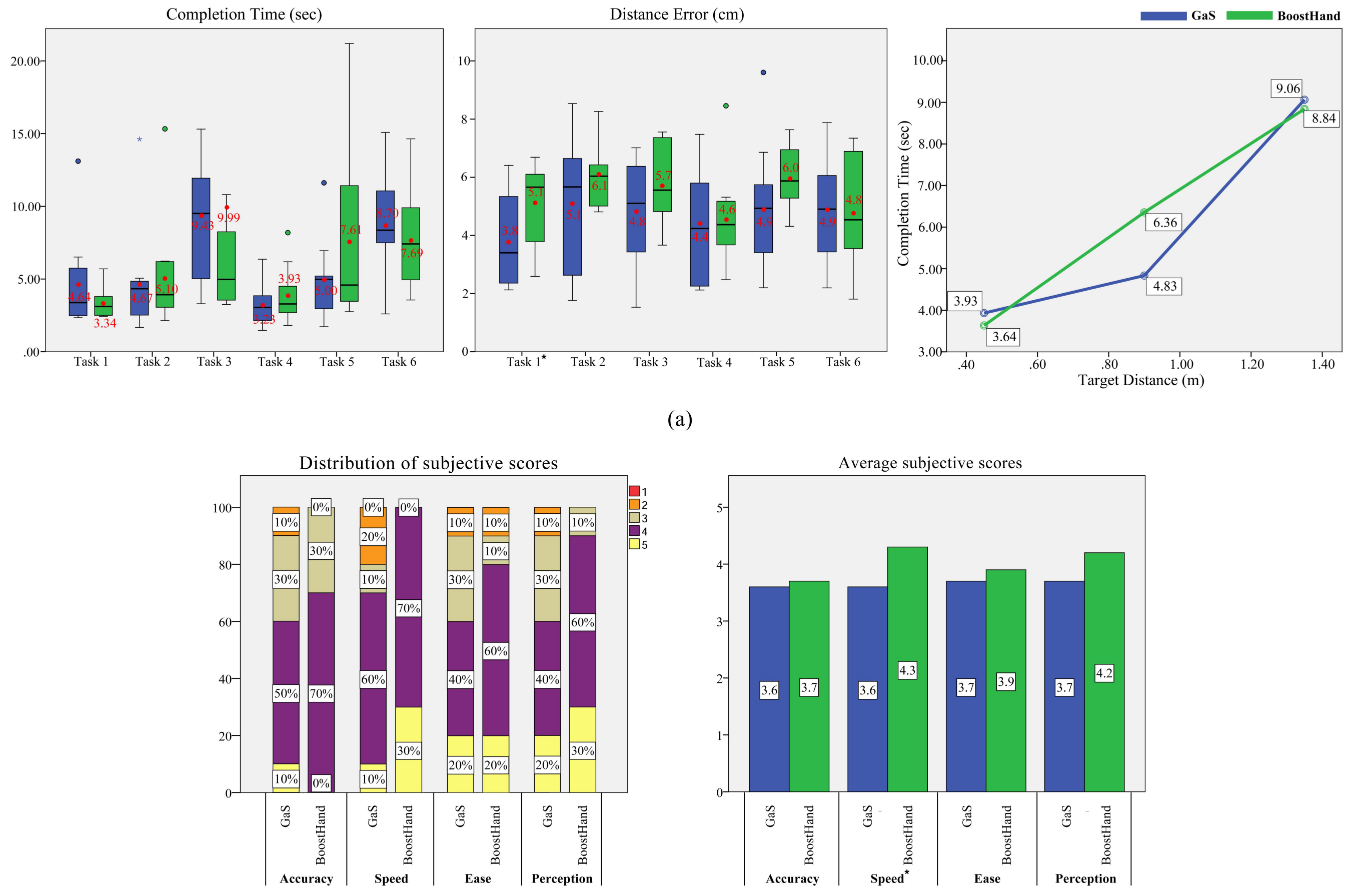
Figure 5: Environment for user task evaluation (top) without obstacles, and (bottom) with obstacles

Table 1: Index of target position in distances and obstacles

	Task1	Task2	Task3	Task4	Task5	Task6
Target Pos.	0.45M	0.9M	1.35M	0.45M	0.9M	1.35M
Obstacles	X	X	X	O	O	O

4.3 Result of Task Evaluation

A total of ten people participated in our evaluation (6 males and 4 females; mean age: ca. 28). Of these, ten had experience with AR software, such as commercial AR applications. None had ever used our system. All ten participants were right-handed. Participants were given a demonstration of each visualization, and were then



(a)

(b)

Figure 6: The user task evaluation results. (a) Result of objective evaluation: the box and whisker plot show minimum-to-maximum, first and third quartiles, and the second quartile (median) with the band inside the box; the red dot shows the mean value; and * notes the significant level for p<0.1. (b) The number of subjective scores and average subjective scores: * notes the significant level for p<0.1.

given a chance to practice with a virtual block. The user evaluation took about 20 min. For each trial, we asked each participant to select a given object and to move it to the target position as quickly as possible.

We used a one-way independent sample t-test to compare the two kinds of techniques (normal and boost) with significance level $p = 0.1$. Figure 6(a) shows the experiment results. There were no significant differences in completion time for the 6 tasks (Task 1: $F = 4.484$, $p = 0.252$, Task 2: $F = 0.088$, $p = 0.568$, Task 3: $F = 1.799$, $p = 0.908$, Task 4: $F = 0.594$, $p = 0.375$, Task 5: $F = 6.259$, $p = 0.243$, Task 6: $F = 0.033$, $p = 0.546$). However, the BoostHand was 1.3 seconds faster in Task 1 and 1.1 seconds faster in Task 6, over the GaS (Figure 6(a-1)). Except for Task 1 ($F = 0.655$, $p = 0.060$*), there were no significant differences in distance error across 5 of the tasks (Task 2: $F = 7.265$, $p = 0.243$, Task 3: $F = 0.483$, $p = 0.236$, Task 4: $F = 2.139$, $p = 0.873$, Task 5: $F = 3.153$, $p = 0.200$, Task 6: $F = 0.396$, $p = 0.869$). However, the BoostHand had 1.2 cm shorter distance error in Task 6 than did the GaS trial (Figure 6(a-1)).

Figure 6(a-3) shows that the BoostHand is faster than the GaS for task-completion at both the closest and the farthest target positions. Additionally, BoostHand shows better completion time per target distance at shorter target distance without obstacles and at longer distances with obstacles.

To analyze the users' subjective ratings with the Likert scales, we used the Mann-Whitney test. Figure 6(b) shows the results of the subjective ratings. The BoostHand technique has more positive ratings than the GaS in all 4 criteria; accuracy ($M = 3.7$), speed ($M = 4.3$), ease ($M = 3.9$), perception ($M = 4.2$).

In the post-experiment interviews, one participant mentioned that GaS seemed difficult for quick task-completion because it did not have a "stretch arms" capability, such as boost mode. Also, another said that GaS was intuitive to use, but the repetition of stretching his arm caused pain. Also, most participants complained that GaS made them dizzy, because it excessively relies on a gaze point. Alternatively, most participants said that BoostHand was easy to use after they got used to it, because they could move their virtual arm with little movement of their real arm. However, one of participants mentioned that he needed more cues, like sounds for interaction. Another said that when his virtual hand went farther, it was hard to determine its position.

5 Discussion and UX Scenario

5.1 Discussion

In our task-based evaluation, the proposed system shows better performance in terms of completion time and distance error for manipulating virtual objects at a long distance in environments with obstacles. The primary reason for our system's better performance is that, with GaS, users can select only the nearest objects. Thus, users are forced to remove obstacles that obscure the target object prior to performing their given task. It also takes more time to select an intended target object when only part of the target object can be seen. In contrast, BoostHand (our method) allows selection of a virtual object anywhere in 3D space using a virtual hand. So, if the user can see where the target object is, it does not require any additional performance costs (e.g., removing obstacles) for precise selections.

Moreover, tracking error differences are not significant, but are meaningful, since the tracking errors caused by Leap Motion largely influenced our system performance where it did not with GaS. We implemented a GaS user interface similar to Hololens so that rotation of the target object only occurs when the user manually selects vertices and rotates them. In contrast, since change-of-wrist orientation is directly applied to selected objects in our system, tracking failures with Leap Motion led to unexpected changes in the selected virtual objects. In fact, participants often got confused when tracking errors occurred. We attribute that phenomenon to the primary reason our system did not show significant error reduction in our task-based evaluation.

In post-experiment interviews, many participants stated that the proposed system was more convenient, since BoostHand allows selection and translation of virtual objects with fewer movements. BoostHand also provides a delicate mode of operation in normal mode, regardless of the location of the virtual hand. Thus, we expect this technique to be useful in a variety of applications, such as an AR classroom.

Our future work will focus on improving accuracy of the system in two ways. First, Leap Motion tracking failures can be reduced by utilizing additional data. Our tracking errors mainly originate from wrist orientation. Therefore, by employing additional data from an inertial measurement unit (IMU) sensor in a smart watch (frequently used these days; does not require complex algorithms), we can resolve most of those errors. Secondly, a limitation of our system in which it cannot precisely select and control targeted virtual objects in certain situations, such as within dense or occluded environments, can be resolved by adding snapping and by implementing transparency. Snapping automatically guides the user's virtual hand avatar to a controllable virtual object when it is within proximity. Transparency allows the user to see through certain objects by rendering them translucent if they occlude other virtual objects.

5.2 UX Scenario for AR Class

We developed an UX scenario for an AR classroom using BoostHand. With the BoostHand system, a lecturer can handle any virtual object in an AR classroom simply by standing at the lecture desk and moving a hand, and can easily utilize supplementary materials in the form of virtual objects, from anywhere. Students can see the lecturer's virtual hand moving around the AR classroom and its actions (e.g., pointing, writing on the board, or relocating interactive materials). For example, the instructor can deploy a virtual solar system over the students' heads while pointing to the sun using BoostHand, causing an explanation banner to pop up at the lecture desk (Figure 7). The same scenario works in the opposite direction. Students can manipulate virtual materials using their own virtual hands while the lecturer and other students watch the action in real time. In this scenario, a networked virtual environment is assumed to be well-constructed for collaborative learning.

Figure 7: An example of an AR classroom, using BoostHand

6 Conclusion

We propose BoostHand, a freehand, distance-free object-manipulation system that supports simple trigger gestures using Leap Motion. We implemented a virtual hand interaction system using simple trigger gestures so users can easily move their virtual hands into the far distance and manipulate virtual objects. We conducted simple experiments to examine performance of our proposed system against the GaS, which is presented via a Hololens UI. Our experimental method leveraged task-based evaluations and post-experiment interviews. Results show that our system provides faster and more accurate handling of virtual objects in the far distance, even in environments with obstacles. Apart from far-distance manipulation, GaS shows better performance due to the tracking errors thrown by Leap Motion, in addition to detrimental effects of object rotation.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.2017-0-00399, Development 3D 360 degree VR content creation technology using multi-view camera)
This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning

References

[1] F. Argelaguet and C. Andujar. A survey of 3d object selection techniques for virtual environments. *Computers & Graphics*, 37(3):121–136, 2013.

[2] T. Ha, S. Feiner, and W. Woo. Wearhand: Head-worn, rgb-d camera-based, bare-hand user interface with visually enhanced depth perception. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pp. 219–228. IEEE, 2014.

[3] H. Lee, S.-T. Noh, and W. Woo. Tunnelslice: Freehand subspace acquisition using an egocentric tunnel for wearable augmented reality. *IEEE Transactions on Human-Machine Systems*, 47(1):128–139, 2017.

[4] M. R. Mine. Virtual environment interaction techniques. *UNC Chapel Hill CS Dept*, 1995.

[5] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pp. 79–80. ACM, 1996.