공학박사 학위논문

증강가상현실과 실존 오브젝트를 활용한 현실과 가상환경 통합 입력 플랫폼의 구현과 검증

Integration of Reality and Virtual Environment: Using Augmented Virtuality with Mobile Device Input

2020년 2월

서울대학교 융합과학기술대학원

융합과학부 디지털정보융합전공

신 종 규

증강가상현실과 실존 오브젝트를 활용한 현실과 가상환경 통합 입력플랫폼의 구현과 검증

Integration of Reality and Virtual Environment: Using Augmented Virtuality with Mobile Device Input

지도교수 이 교 구

이 논문을 공학박사 학위논문으로 제출함

2020 년 1 월

서울대학교 융합과학기술대학원

융합과학부 디지털정보융합전공

신 종 규

신종규의 공학박사 학위논문을 인준함

2020 년 1월

위 원 장 이 중 식 (인)

부위원장 이 교 구 (인)

위 원 서 봉 원 (인)

위 원 이 준 환 (인)

위 원 남 주 한 (인)

Abstract

Integration of Reality and Virtual Environment: Using Augmented Virtuality with Mobile Device Input

Jongkyu Shin
Department of Transdisciplinary Studies,
Program in Digital Contents and Information Studies
Graduate School of Convergence Science and Technology
Seoul National University

Interactive virtual reality (VR) experience has become more common and widespread these days, with the development of head-mounted display (HMD) technology in recent years. The interaction inside the virtual environment is enabled by capturing inputs from users usually by mechanical hand-held controllers. Mainstream HMD-based VR platforms come with these trackable multiple degrees-of-freedom (DoF) controllers with a high level of versatility. This enables a tremendous diversity in input types and forms to provide more interactive and immersive VR experience. However, these devices lack in cross-platform compatibility as the protocols are non-standardized. Thus, it leads to a steep learning curve. The trouble in

learning these alien devices results in non-intuitiveness and confusion to users. Also, the precondition of immersive virtual reality platform with HMD is that it fully occludes the vision of the real environment while users are participating in the VR. The virtual isolation inevitably arises with this condition, keeping two different worlds separate and not integrated, leading to potential risks to users.

In this study, we propose a method to use a more familiar, and a more intuitive interface as an input device for VR by importing objects from the real world and use it as an input device. Using augmented virtuality (AV), a real texture of outside object and user's hand is overlaid onto the virtual synthetic scene and users are able to interact with the device while they are still wearing the HMD device. In order to properly import a real-world object into a three-dimensional virtual environment, we develop a platform and conduct a series of user studies with various tasks to implement an efficient visual representation method yet to maintain the level of immersion with minimized vision focus swap inside the virtual environment. Through our investigations and findings, we establish an AV platform with guidelines providing an accurate visual representation with minimized user confusion and a high level of usability.

The results demonstrate that the proposed method significantly improves and gives promising effect for better virtual reality input experience. Also, properly blended visual representation with synthetic background and physical object with its original texture reduces the physical side of the virtual isolation caused by the occlusion. Hence, the proposed

method is expected to be used as tool for the growing research field of mixed reality and augmented virtuality.

Keywords: Virtual Reality, Input Interface, Human-Computer Interaction, Mixed Reality, Augmented Virtuality

Student Number: 2014-30810

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Motivation

Virtual reality (VR) technology enables to deliver visual and aural computer-based synthetic simulated experiences that is similar to or completely different from the real world. The technology is applied in many areas including engineering, entertainment, education and in any areas that require experiences that is hard to archive in real life. Recent development of consumer-based Head Mounted Display (HMD) systems enabled users not only to easily access to the visual experiences of synthetic visual scenes, but also to actively participate in the virtual reality (VR) environment with a great level of immersion. Combining with modern computer graphics, previous generation devices such as Oculus Rift DK[137] series, and the low-cost smartphone based systems such as Google Cardboard VR[49] or Samsung Gear VR[101] focused on delivering stereoscopic imagery to users yet had limited ways to gather input entries from users and let them interactively participate in VR scenes with interaction. To interact or manipulate with the VR, indirect manipulation methods including gaze input or traditional universal input devices such as joysticks or keyboards with proxy input interface UI were utilized as there was no better way. There were many attempts to recover this limitation, but none of them was considered as the standard method and widely adapted in consumer-based devices. As a workaround for manipulation / interaction in VR, there came current

generation HMD platforms such as HTC Vive and Oculus Rift equipped with tracking capability. They can track multiple dedicated objects for the platforms including the HMD. Thus, they came out with packages including their trackable hand-held mechanical controllers which have multiple degrees-of-freedom (DoF) as shown in Figure 1.1. Most controllers come with multiple buttons and touch pads, supporting different types of fingertip inputs. Also, they have 6-DoF which means the platform recognizes the x, y, z positions as well as the pitch, roll and yaw values of the controllers. This technology fills the missing key to the definition of VR mentioned by Burdea G et al.[19], which is the communication. According to their statement, the synthetic virtual world is not static, and it responds to users input, with interactions enabled.

The trackable controllers with multiple input ways create a great opportunity for diverse interaction methods. For instance, a single HTC Vive controller has 8 clickable physical buttons, one axis-separated movement buttons with ability to custom-map all of them. But this opportunity for unlimited ways of custom mapping also resulted in a steep learning curve in usability as they are not standardized and inconsistent throughout different platforms. Often these complexity results in lack of intuitiveness, confusion in control and unnatural interaction for users[11].

This direction of HMD platform development is the corresponding response to the proliferate demands for deeper, and more immersive virtual reality experience with a higher level of sensory deception. Moreover, this path leads to the right end point of the Mixed Reality spectrum introduced by

Figure 1.1: Mechanical multi-DoF controllers

(Adapted from reference [137][138])

Milgram and Kishino[88] shown in Figure 1.2, which is the Virtual Environment. It is an artificially created world with capturing zero contents from the real world. The purpose of this environment is to give users a complete immersion towards virtual reality, trying to stimulate all human sensory with synthetic stimuli at the same time. On the other hand, the isolation problem inevitably arises as users get extremely difficult to realize the context of the reality, nor communicate visually or aurally with outside in real-time. This concept doesn't allow users to see or hear things from outside while they are immersed in VR; Conversely speaking, they are isolated in the VR. Because modern HMD platforms usually target end-users for consumer group, this could be a serious problem and considered as inconvenient or dangerous[84], as those platforms are likely to be installed and utilized in our everyday life environments such as home and office where multiple obstacles do exist.

Figure 1.2: The Mixed Reality Continuum (Adapted from reference [88])

Augmented reality (AR) and augmented virtuality (AV) are parts of the Mixed Reality spectrum and they both compromise in terms of combination between the reality and the virtual environment[87]. These two terms differ from their based cornerstones. AR applications are built upon a real environment whereas the AV is the opposite. AR represents computer-generated virtual elements on to the real background image, while AV overlays specific contexts such as objects from the real world onto the virtual scene. To maximize the benefit of integration and utilization, these methods in-between two extreme points of the spectrum show a safe, and well embedded results. AV enables the virtual environment to have a synthetic background while interactable physical elements such as object or people are dynamically integrated into the VR without any risks of collision and safety issues.

The reason for the current VR platforms do not target these balanced methods in spite of abovementioned advantages and takes the risk of isolation is that because placing both elements from the real world and virtual reality often breaks immersion[38]. Users have to swap their vision focus in order to

properly acquire information from two different environments and this tends to reduce performance and may increase cognitive load[53]. As detection and processing of the image from the outside is not an easy technique to accomplish., technical barriers on recognizing and segmenting outside context plays a great role for negatively effecting the level of immersion. When combining two different worlds into a blended visual representation, an equal level of image quality is crucial for consistent immersion. However, despite the challenges, there are areas that require the information of outside world while being immersed in VR and blending these two worlds can benefit from more possibilities.

1.2 Topics of Interest

The purpose of this dissertation is to propose a method for reducing the physical gap between reality and virtual reality by using an active object as a medium as shown in Figure 1.3. Blending two different worlds using object has advantages as it minimizes the immersion interference towards VR and can provide a more natural, intuitive input control for virtual reality by encountering a user-friendly physical input device. Also, importing objects to VR enables direct manipulation[31] and discards the necessities for proxy inputs on VR scenarios that require outside context such as context-aware entertainment or simulation applications.

To implement a natural, intuitive input control platform for virtual reality with properly blending the real world with the virtual environment, we focus

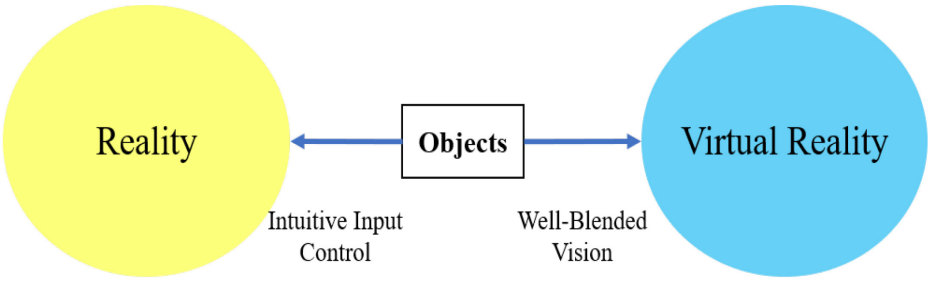

Figure 1.3: The focus of this study - using objects as mediums between reality and virtual reality

on the research field of the augmented virtuality (AV). The overlaid real object is an active device, which is determined by the existence of its own power source. As the power source can be utilized to easily transform the communication or trigger signals for input, we first implement with an active object, then find possibility to investigate further.

The feasibility of the platform is validated through a series of user studies. Within the implementation, we compare the proposed method with conventional platforms to quantify how the method affect the overall virtual reality experience as well as measuring usability with a variety of scenarios and interaction types. In this dissertation, to overcome the usability in input in virtual environment and isolation challenges in VR, we propose an approach that enables to users to directly interact with object outside VR yet maintain the level of immersion with well blended worlds. We utilize an object as a medium to link the virtual world and the real world, and use it as a more familiar, more intuitive input interface than regular controllers that come with the HMD platforms. To profit the possibilities of easily modifiable

environment, we follow the form of augmented virtuality. This approach has the advantage of establishing a flexible virtual environment yet considering the relevant data of the real environment for filling the physical gap between two different worlds. Blending both worlds can benefit for more expandability and ensure a more intuitive platform by using user-friendly input devices for virtual reality. The results demonstrate that the proposed method significantly improves and gives promising effect for better virtual reality experience without isolation, hence, to contribute to the growing research of augmented virtuality. The goal of this study is ultimately extended to properly linking two different worlds using objects as a medium.

The contributions of this work are summarized below:

Through augmented virtuality, we implement the technique for overlaying real object onto the virtual scene to integrate reality with virtual reality. We design and implement the platform, followed by experiment design and user studies. We validate the platform through the comparison process and fine tune the method to establish a more precise overlaid object for virtual environment. Implementations are revised step by step with discovering limitations as AV research area is quite new up to recently, and during the process we address issues and important factors for an AV platform with accurate visual representation and a high level of usability.

We apply our idea on active objects which can be considered as things with dedicated power sources. Through experiments and discovering insights

from our method, the results demonstrate that the proposed methods significantly improve and gives promising effect for better virtual reality input experience with less isolation.

1.3 Outline of the Thesis

In Chapter 2, we first present the scientific background and related work. we address issues that current HMD-based virtual reality platforms have and focus on previous studies resolving addressed limitations by blending the reality with the virtual environment. Implementations and researches across mixed reality is explored, especially on the term augmented virtuality, which is augmenting synthetic elements onto the real world. We also discover methods that were previously conducted in attempts to provide of inputs in virtual reality experience to users in various forms and shapes.

In Chapter 3, we explain our findings from the first attempt to overlay an active real object onto the virtual scene. An active object in this context was a touchscreen based mobile phone, as it is a more widely adapted, more user-friendly interface compared to mechanical controllers of the HMD platform. We present a method to overlay the object, make it interactable, design UI and gather results via user study consisted of various input tasks and survey. As issues were found on the first attempt, we conduct additional experiment with modified implementations to address the culprit of our found limitations, targeting to improve the platform as a method to import objects

onto virtual reality with intuitive control and high accuracy with minimized user confusion.

In Chapter 4, we pay attention to importing active object properly into virtual reality with a low risk of immersion breaking in VR experience. We change the image representation technique in the system, and modify the properties of outside object with resolved issue which was found in the previous study. We implement the improved method and apply features to conduct experiments to find out the feasibility for the system to support natural touch input manipulation on touch devices for HMD-based virtual reality. We first test the system with previous method, then the experiment is followed by the comparison with bare-eye conditions was conducted in order to explore users' perception towards the system factors of intuitiveness when controlling the object with or without the HMD. User survey using Presence Questionnaire (PQ) was conducted in order to explore the important factors for a highly immersive augmented virtuality platform. Results show that our system minimizes the perception gap against input scenarios in natural state, and provides decent performance in user input with a proper visual image adjustment.

In Chapter 5, we conclude the dissertation with summarizing with our findings in the process the study, and state the contributions of the research. Future work is represented with our insights gathered throughout the implementations and user studies.

Chapter 2. Background and Related Work

In this chapter, we introduce the challenges that current HMD-based virtual reality platforms are facing. We address several issues within the platforms and target our research area for our proposed method to be applied. Previously discovered studies for resolving addressed problems along the area of visual representation and inputs in the virtual environment are introduced in next subsection. The image representation technique in VR can be blended with reality for additional immersion and expandability of the platform for better user experience. We go through visual representation techniques and input methods in VR, explain previously explored ways and indicate current challenges throughout the methods. Subsequently, we explain approaches that involves real objects in all mixed reality scenarios, followed by the summary addressing augmented virtuality as the cornerstone for a potential interactive VR platform.

2.1 Challenges in HMD-based VR

2.1.1 User Protection

The HMD-based virtual reality systems are known as the fully immersive platforms for VR experience, whereas other systems which utilize two-dimensional screens or projection devices are known as non-immersive or semi-immersive platforms, respectively. The distinctive difference of the fully

immersive VR methods is that they provide the most direct vision experience with binocular images by separate LCD screens placed in front of the user's each eye. Then the sense of immersion is determined by multiple parameters including the resolution, the refresh rate, the field of view (FoV) and brightness of the screen[33]. Unlike non-immersive and semi-immersive structures, HMD-based platforms have unique structure as users are required to wear a special, heavy machinery on to their heads with tethered cables for immersion. Thus, certain level of safety and health consideration should be ensured for both physical and psychological user protection to avoid harm and injury. Physical issues mainly come from the unique designs of the headset, as all of them have visual displays which are closely coupled with eyes for stereoscopic image representation. Additionally, the complex structures of the headset results in bulky in terms of size and weight, as the weight of head mounted displays including see-through AR glasses in current market varies from 350 to 550 grams. Prolonged usage of the HMD could cause damage to the tissues of retina by the electromagnetic field and synthetic lights from displays. Additionally, it could also cause discomfort, even damage to neck and spine caused by the unnatural posture resulted from wearing a heavy device on to the head[56]. The design of headsets is lacking in ergonomic factors up to this day.

The blocked vision also acts as a potential risk for physical issue, as when users are wearing an HMD, they are functionally blind in real life. The chances of colliding into obstacles such as tables and chairs are high as they move around inside the virtual environment and move their body parts for

interaction, while holding the controllers in their hands. Unlike previous generation devices, recently developed systems support the six degrees-of-freedom of the player. Within the dedicated area, users are able to explore the place with their own physical movements. However, the risk of collision injury even gets higher as there are still cables attached to the HMD device, meaning there are possibilities for those cables to be kinked and act as the obstacles themselves. Manufacturers are also acknowledged with the problem and some does include integrated camera on their HMD device for monitoring outside context[138], but the lack of feedback features for danger still leaves potential risk in such scenarios.

For the psychological issues that exist in current HMD, cognitive aspects in perceptual shift / disorientation, changes in perceptual judgment and change in psychomotor performance were explored by Howarth[57]. Stress, addiction and mood change are the factors that represent behavioral aspect of psychological issues. Wilson[126] suggested other behavioral effects including hallucinations, dissociation, lateralization and retreat from reality as the results of prolonged usage of VR headsets. All psychological concerns explored by scholars mainly focus on longevity in terms of usage, due to the reason that VR tend to deliver provocative contents via the platform.

2.1.2 Cybersickness

Cybersickness can be defined as a physiological issue that occurs across virtual visual experience. Symptoms from slight headache to an emetic

response could happen among users after the exposure to VR. Number of possible theories exist for the cybersickness[61][69][73], but the most common and popular theory can be described as the sensory mismatch or the sensory conflict. Meaning that, unlike natural experience in the real world, the experience in VR can be split or partial for each sensory. For example, a user in VR headset can virtually move inside the VR using the controller, while the actual location of the user remains the same. This leads to a sensory mismatch between the visual cortex and the motor cortex of the user, and prone to cause cybersickness symptoms.

The severity of the symptoms varies upon the user's physical condition and depends on the quality of the video in the virtual environment, but there are some factors that have been identified to contribute to symptoms. The factors include time lag, field of view (FoV) and *vection.* Time lag is the sense of experience when delay occurs between the certain execution of action and the response from the image representation. The latency could come from head movements, response upon certain interaction (e.g., pressing a button inside VR) and the delay from the LCD display panel itself, as certain types of LCD matrix have latency in fast switching scenarios. Additionally, poor calibration in platforms or bad optimization in visual representation performance also take a role for the time lag. FoV factor is likely to create issues when combined with three factors including inter-screen distance (ISD), lens inter-ocular distance (IOD) and users' inter-pupil distance (IPD) as shown in Figure 2.1. The ISD refers the distance between the centers of separate LCD screen inside the HMD, IOD refers to the distance between the optical

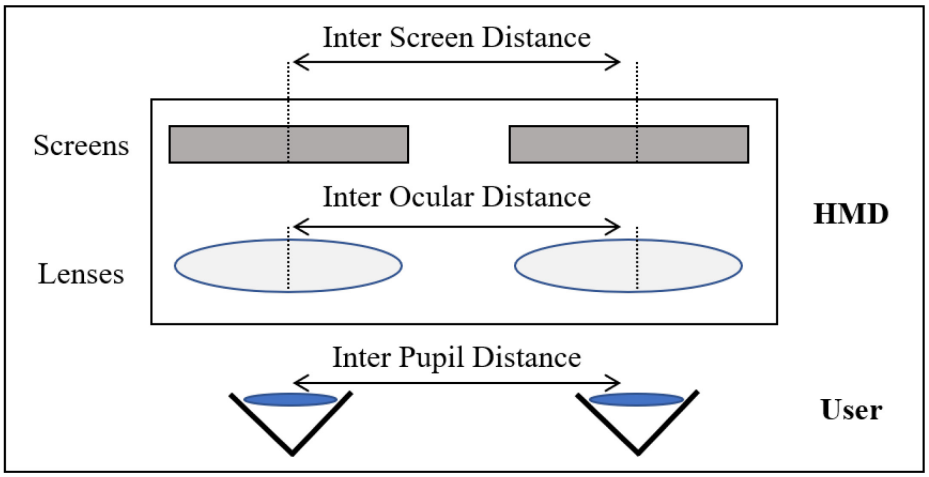

Figure 2.1: Description of three variables (ISD, IOD and IPD) in stereoscopic visual imagery representation via HMD

lens center between two lenses inside the device and IPD refers the distance between the pupils of the user. Mismatch of these three factors may create visual discomfort, transient heterophorias or muscle imbalance in the eyes of the user[95]. While modern HMD devices support calibrations with variable IOD (called as “IPD” calibration in most commercial platforms), improper adjustments create an off-center axis in visual representation, and leads to potential factor to create visual related cybersickness symptoms. *Vection* is cognitive factors for self-motion perception in the virtual environment. It is a feeling that the person is moving without physical movement, caused by a perception of movements of other things. As it is difficult to maintain the natural correlation between vision senses and the motion balance of human body in HMD-based virtual reality, this illusion occurs and confuses the user

and creates a possibility for cybersickness symptoms.

To measurement of symptoms is a crucial point for both system and content developers in order to maintain a certain level of safety for all users. Kennedy et al. introduced a survey-based measurement called the Simulator Sickness Questionnaire (SSQ) for measuring cybersickness symptoms for users. The questionnaire consists of 16 different symptoms listed in three clusters, which are nausea, oculomotor, and disorientation. The SSQ is usually used in developing scenarios for testing out a new product or content, during user involved experiments. Similar questionnaires to the SSQ have been developed to measuring cybersickness in virtual environment. Nausea specified method developed by Muth et al.[89] focuses on detailed symptoms on nausea profile (NP), categorizing the symptoms into somatic distress, gastrointestinal distress and emotional distress. Much simplified versions of questionnaires are introduced by Keshavarz and Golding[70]. Keshavarz's Fast Motion Sickness method measures the score by asking general subjective feelings from zero to 20 during the experience while the latter method called Motion Sickness Susceptibility Questionnaire measures discomfort across general simulator platforms.

As we all know, questionnaires are very subjective measures that only rely on user's feelings during the experiment, and it is generally difficult to maintain certain level of safety due to the variations of users' skills and physical differences unless we monitor a large scale of people. Other than subjective methods, more objective oriented methods include using bio signals for establishing automated real-time measurement platform, to be

utilized as early warning platforms to ensure VR users are always under suitable condition. Electroencephalogram (EEG) and electrocardiography (ECG) signals are used in these researches. Chuang et al. found out that a higher level of cybersickness symptoms increased activation of alpha (8-13Hz) and gamma band (>32Hz)[32]. The tendency and variation for EEG and ECG signals differentiate in different cybersickness levels, and more precise prediction can become possible after gathering a large size database of the signals. Inevitably, the system becomes bulkier due to additional sensor arrays. To resolve the bulkiness of the system, the nGoggle[85] was a prototype introduced for a portable HMD with integrated EEG electrodes within the headset. But these platforms still cause inconvenience in terms of usability as the apparatus setup for these systems require users to wear the electrode cap and the HMD at the same time.

Measuring and classifying cybersickness is difficult as the clear answer for the symptoms is not yet discovered and the root cause of emerging cybersickness is still hidden. Also, the level of experiencing symptoms hugely vary from person to person. The combination of answered questionnaires and bio signal data can provide a lot of information about the main cause of the cybersickness.

2.1.3 Isolation

Typical recent HMD based virtual reality experience blocks out the reality and replaces it with a virtual world. While users are being immersed in the VR

wearing the HMD, they challenge the isolation from the outside, real world. The isolation problem first arose as the user had to stay in one static place when wearing the HMD. As there was not any method to consider the physical location of the user and reflect the data with the virtual environment, the limitation was clear that the users are strapped in one static place (e.g., sitting on a chair) to experience the visual imagery. In order to overcome this problem, the industry and academia focused on integrating the HMD-based virtual reality experience with indoor positioning technology. The first basic approach was to provide tracking of the HMD itself. Introduced by Shin et al.[105], the ultrasound-based indoor tracking was combined with the HMD. Within the designated area, they attached a custom ultrasound transmitter on the top of the HMD and receivers on the ceiling of the infrastructure. Calibrating the existing area and gathering the absolute location coordinate values, the location of the user in virtual environment was simultaneously updated as the person moved around the area. This changed the static HMD-based virtual reality experience to an experience which supports translational movement (a.k.a. “free-roam” VR), but the inability to track body parts still kept the platform to have limitations for expanded interaction with the outside world. Similar concept but more advanced and precise method from industry was first introduced by HTC and Oculus, with IR-based indoor tracking. This method was considered as more expensive compared to the ultrasound-based platform but had less hassle as they did not require installation of receiver beacons on the infrastructure and also supported tracking additional handheld controllers. With this platform, users now have ability to explore the physical

area instead of just staying in one static place. Additionally, giving inputs and interacting with the virtual environment became possible with the trackable controllers. Such systems are considered as the platforms that purely focus on the term, “virtual reality” on the Mixed Reality Continuum introduced by Milgram and Kishino[88]. Even though the experience became more active and more immersive, the experience was kept isolated in the virtual environment. Also, as bi-directional interaction became possible and movement is supported in the virtual environment, the risk of collision became another problem within the experience. Because the HMD device is designed the occlude both eyes of the user, it is nearly impossible to interact with an object placed outside, and even could create a dangerous scenario as the user cannot avoid collision with nearby obstacles. The result was a very controlled environment with strict boundaries, making users only to stay in the trackable physical space and the system was not able to respond to sudden environment change of the real-world context. Some approach such as Virtuix Omni[139] introduced a 360 degrees, omni-directional treadmill working as a contraption for safe moving, instead of utilizing the actual physical space. Users wear special anti-friction shoes on the treadmill and naturally move / run to explore inside the VR. As a result, a safe VR platform with translational movement is established with a safety barrier, but still users get locked in the virtual environment. The visual information is all based on synthetic computer image, and any static or dynamic status change of real-world environment including objects are never reflected to the inside vision of the HMD.

VR platforms are often utilized as socialization tool to treat disability and

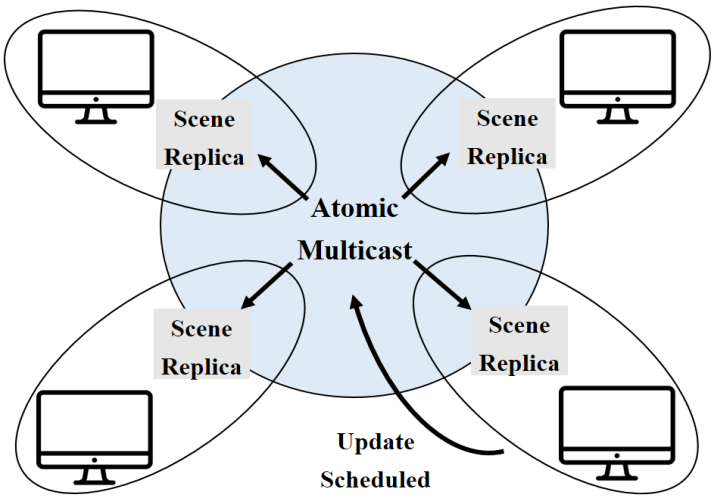

Figure 2.2: The Active Replication Model in CVE architecture

help rehabilitation for certain groups of people using virtual characters inside the virtual environment. In applications which require human to human interaction, the opponent or the collaborator in the VR depending on the story of the context, is established either via telepresence using network or by a computer intelligence disguised in a synthetic human figure (e.g., virtual humans[44]). This way of implementation is often applied on games, education and training purposes in remote-location socialization scenarios. Ironically, the physical isolation of the HMD user still exists in the real life as the whole virtual experience via HMD takes place within the single user's vision and excludes other people physically from sharing or participating with. To overcome this social isolation within the co-location, a large amount of research focuses on implementing virtual reality platforms that support collaboration in co-located scenarios. Collaborative platform refers to a

system that simultaneously supports the existence of multiple people inside the same virtual environment with added interaction. Whether the interaction is between the players or with the virtual environment itself or supports both, the key element to these platforms is a shared experience. In terms of the collaborative virtual environment (CVE) research area, collaborative HMD-based virtual environment usually follows the active replication of the consistency model classification. As shown in Figure 2.2, the active replication model represents the equal, synchronized same data across users. With this model, it becomes possible to correctly view the shared viewpoint in the VR for multiple users simultaneously.

In co-located collaborative virtual reality platforms via HMD devices, supporting multiple users in the same location are mainly categorized into two implementation methods depending on the types of interaction between the users. First one is the interaction between the two or more HMD users, and the second type targets the interaction between the HMD and the non-HMD user. The first method comes in systems with more complexity than the latter as the platform requires additional external monitoring devices to detect each player individually and represent their image on to each players vision either in a real texture or a synthetic virtual avatar form. The system has to be precise enough to track accurate either users' whole body or partial parts (e.g., hands), while maintaining low latency in order to minimize discomfort and confusion in peer-to-peer interaction scenarios. Several efforts target co-located multiple HMD user scenarios, enabling interactions between the users [40] or with the environment[8][77]. However, up to this day it is difficult to

find methods that supports physical contacts in between users as in order to archive the implementation which enables actual body contacts of HMD users, the whole body must be tracked at a precision level of less than one centimeter. The challenge still exists with a technology obstacle for maintaining safety and avoiding collision between users.

On the other hand, collaborative VR platforms supporting both HMD and non-HMD user could focus more on interaction with the virtual environment or between the users, as they had less obstacle for precisely tracking and representing both users at the same time. The interaction with the environment can be interpreted as the symmetric coordinated action, which users work together on equal footing, whereas asymmetric action is the interaction between users without any specific timeline order (e.g., both are fighting with swords) In terms of sharing vision, the leading and main viewpoint of the virtual vision is usually the one inside the HMD user, and often the external screen placed on the HMD itself (oculus) or outside in the real environment to represent the visual status of the inside world[59][100]. Gugenheimer et al. [51] introduced a platform that could perform both symmetric and asymmetric coordinated actions between HMD user and non-HMD player. The visual cue interface for the non-HMD person was the projection vision on the floor and additional mobile hand-held display for monitoring inside virtual vision. In Addition to this concept, a safety barrier implementation to avoid contact between the users were done by Yang et al.[131] by creating a coarse synthetic visual barrier around non-HMD user to give visual warning to the HMD user.

Many researchers have explored in methods to minimize both physical and social isolation factors of current VR platform. However, staying blind in virtual reality behind the HMD has its limitations and the necessity for more socialized experience involved other blending techniques between reality and virtual reality.

2.1.4 Indirect Manipulation

One of the unique facts for input / manipulation in current HMD-based VR platform is that the direct manipulation of objects or interfaces represented in virtual environment is not possible. For example, when we see a menu UI in virtual reality, it is not possible to use our fingers to select it and when we see a ball inside VR, it is also not possible to grab it using bare hand. The only way to control and enter input inside VR is using proxies for manipulation. The hand-held controller which current HMD platforms are equipped with could be categorized as the proxy interface for VR. They are not only used to position and select objects inside the virtual environment via ray-casting technique but also used on controlling devices on a specific scenario, such as pulling a trigger for a gun in the game. All input methods are driven by those controllers by using their versatile ways of input methods, but still may lack in intuitiveness for several reasons. First, as it is still using an additional proxy input method on to a natural input and second, current VR controllers cannot provide kinesthetic and tactile feedback levels compared to those in real world. When operating certain objects in VR, the knowledge about how things work

in the real world and how to interact with it defines the way of controlling.

In 3D spaces, multiple degrees of freedom in control should be guaranteed for objects for natural interaction. They include translations (2-DoF), rotations (1-DoF), and scaling (1-DoF). In the experiment of Knoedel et al.[71], they compared the direct and indirect methods in object manipulation in 3D spaces. The direct manipulation method let users control the object by placing their hand directly over the object whereas the indirect method used a proxy touch pad device to control the target. The result indicated that, direct manipulation is better in performance and completion time, whereas indirect method lacked in efficiency and precision. As presented in several researches, the misalign and the difference between the experience in real life and virtual world affects negatively on immersion and intuitiveness of the control.

2.2 Visual Representation in Virtual Reality

With the development of computer-based graphics, the virtual environment has become more immersive as it could provide visual experiences with the real-like synthetic imagery. The introduction of virtual reality interfaces enabled people to experience things that could not happen or are nearly impossible to archive in real life. By the definition, virtual reality can be defined as a computer technology to create a three-dimensional simulated environment. It mainly focuses on visual information combined with modern computer graphics, and the display types for the imagery generally varied from two-dimensional LCD screens and spatial projector settings.

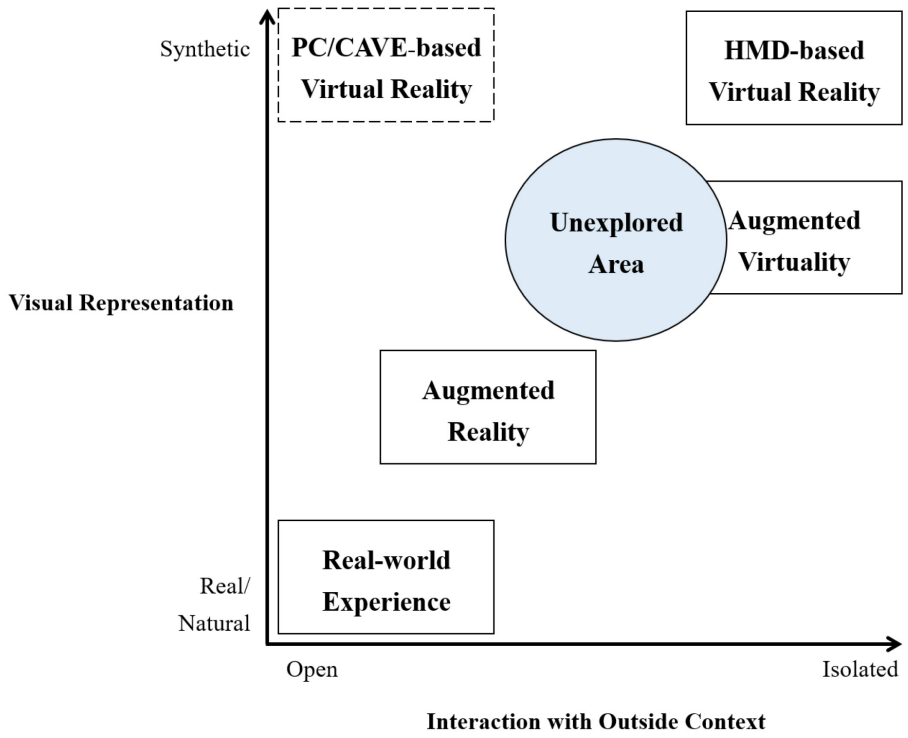

Figure 2.3: Taxonomy of technologies among the Mixed Reality spectrum

After the release of low-cost HMD devices, true 3D image representation became possible by creating stereoscopy with separate displays for left and right eye. The devices became widespread to ordinary consumers, and the most recognizable component of term “VR” has been known as the HMD-based visual / aural simulation platform for years.

However, as these platforms had limitations coming from the blocked vision, the isolation issues mentioned earlier still exists in the systems. Figure 2.3 shows the various methods of visual representations in all mixed reality spectrum and the targeted unexplored area of this study. Numerous hybrid

approaches have been addressed to resolve the isolation problems and represent virtual elements efficiently and safely while maintaining the level of immersion and presence.

2.2.1 Overlaying Virtual Elements onto Reality

To overcome the isolation problem that exists in the HMD-based virtual reality and expand the immersive virtual environment experiences, researchers focused on a different approach by blending both reality and virtual reality[84]. The term for these methods is called the Mixed Reality technology. The MR is the blending of real and virtual elements to provide a new visualization and environments. A widespread subcategory of the research is known as Augmented Reality (AR). This is a technology that overlays virtual elements onto the real-world surroundings. Even though the reality is shown through a display device that is filled with streams of the images from a camera instead of seeing it through bare human eye, it does not block the outside world and the outside context stays as the background layer of the visual information. On top of that, computer-generated virtual elements provide an additional layer of visual information, which enhances the real-world experience with additional information in many areas such as games, simulations, engineering and education[136][24]. AR targets to integrate both reality and virtual reality, not just as a simple display of artificial data on to real vision. Methods utilize not only visual information but also haptic, auditory and olfactory. On the visual term of the AR, placing the virtual

element on the three-dimensional real environment is the biggest feature. Virtual Fixture[97], developed by Louis Rosenberg in 1992, is known as the first immersive augmented reality system ever made. Users were to control robot arms while wearing special optics on their eyes and wearable devices on to their bodies. With the help of the optics, the vision aligns the users' arms and robot arms to be perceived to exist in the same location. Then overlay of virtual image shows in the optics for support remotely manipulated tasks. The virtual sensory overlay works as fixtures to improve the work performance of the user.

For decades, AR focused on different applications that could improve our everyday life by augmenting additional visual information to our eyes. The image representation of AR application can be done in three ways. First method is the video see-through which replaces all visual information of the user with real-time video feed of reality. As the reality is digitally converted into pixel data, it is known as the easiest method to implement as the background can also be modified if needed just like the intended virtual overlay. Most of modern smartphone based AR applications in many different categories such as games, maps, and other areas takes advantage of this method, for placing different kinds of virtual objects by registering fiducial markers on to the algorithms of the applications, gathering all real imagery by the device camera and blending the imagery with virtual elements, then showing the final imagery through a digital display. Although it has a limited in field of view (FoV) and relatively low resolution of image quality of reality, with the expansion of smartphones and the ease of implementation made this

way of AR very popular.

Another way for image representation in AR is the optical see-though method, which overlays holograms of virtual objects onto real environment. This often requires head-worn devices, hand-held displays and spatial setups where AR overlay is mirrored either from a planar screen or through a curve screen[9].

This includes the first known HMD system of Ivan Sutherland in 1965[111] and previously mentioned Virtual Fixture. The idea of this method adding virtual objects holographically through transparent mirrors, while the real-world background image is not intervened, letting human bare eyes can recognize naturally. It provides safety as power fail does not affect any occlusion in vision, and cheaper for setting up optical devices for virtual overlay. However, it has spatial with FoV problems, cumbersome setups for camera calibration and often occlusion in overlay images which utilizes projecting machines instead of close-by optical wearable devices.

The final method for displaying virtual elements in AR is projecting the virtual overlay onto screens without any necessity of additional device for human eyes[12]. Suited for covering large area for multiple people, similar to Cave Automatic Virtual Environment (CAVE)[34] platforms which uses projectors to show virtual objects in a spatial setup. Projection AR systems exclude representing the whole imagery as virtual but focuses on overlaying synthetic objects onto the area using projectors. Just like CAVE applications, usually more than one projector is used for multi-wall projection, and the trouble in installing / implementing can be improved in Zhou et al.'s[136]

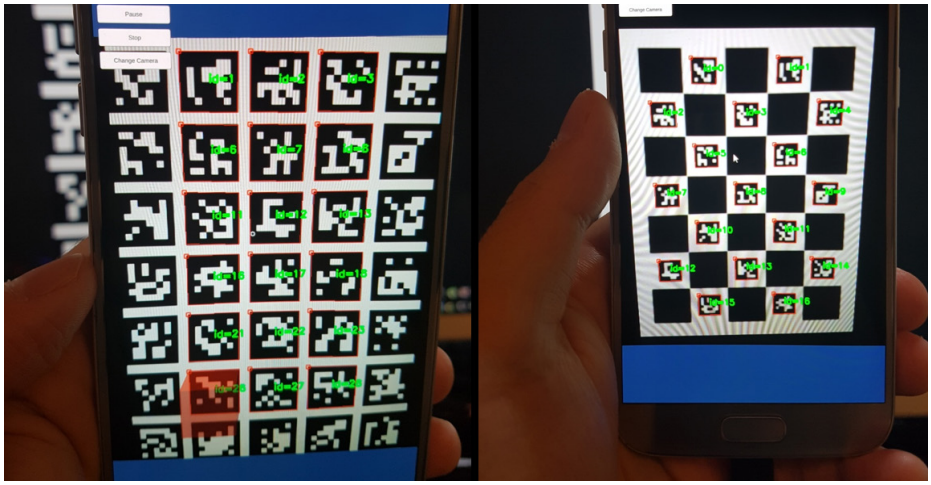

Figure 2.4: An example of ARUCO marker detection method

method using smaller projectors for easier installation. Another research including the CAVE include Mono-a-Mano[10] method, which overlays virtual elements directly on un-calibrated area then dynamically calibrating the overlay position by tracking the depth of the place and user hand simultaneously. The method is well suited for large exhibitions and places as it is only option available to cover larger areas and provide sharable AR experience simultaneously to multiple people compared other representation techniques, limitations still occur as it requires calibration of projector devices, occlusion prevention is crucial for reflective materials for virtual overlay and it is difficult for outdoor use as the method requires a precise control of light brightness for avoiding blurry image representations.

The key for all kinds of AR applications is overlaying the virtual element on the intended location of the real-world. Early studies including Virtual Fixtures[97], ARQuake[114] had very limited FoV for convenience in

tracking and error control. Computer vision-based tracking algorithms were involved in most of AR applications for locating the right place for displaying the virtual object, using such as feature extraction or marker detection and often involves Simultaneous Localization and Mapping (SLAM)[116] or Parallel Tracking and Mapping (PTAM)[121] techniques to locate the intended area for overlaying virtual elements. The tracking divided into two main categories, which are marker-based and markerless tracking. Marker-based method utilizes a specific paper-based binary marker such as QR code or ARUCO[48] marker placed in the real space for the camera to detect its existence, specific bits and orientation. In Figure 2.4, an example of marker-based AR application using ARUCO code is shown. A paper printout of a maker is placed on the table, and the AR application detects its location and orientation then places a virtual cube on top of it. Related researches[41][67] in AR area systems successfully demonstrate registering virtual objects on top of cardboard markers.

Compared to markerless method, marker-based AR applications require relatively less sophisticated tracking algorithms as the makers used in the scenarios have a clear boundary to recognize and strict rule to follow. On the other hand, markerless AR method requires more complex computation for context recognition and have a higher barrier environment compared to marker-based environments. As real-world region of interest (ROI) should be interpreted not just as pixels with color, but also data with a height, width and depth for accurate classification. This often results in delay in the process, making the implementation harder for responsive interactable platform.

Workaround researches to minimize the delay problem often involves hybrid approach, such as using GPS signals as shown in work of Santos[102]. In his method, in order to compensate the error that appears from low accuracy of pattern matching is aided by GPS signals to efficiently display the virtual elements with precise detection of real-world context.

2.2.2 Overlaying Real onto Virtual Environment

Augmented virtuality (AV) is a subset of VR technologies which is placed next to AR in the Mixed Reality Continuum. AV is referred as to have a "window" inside the virtual environment to see the outside real world and mostly implemented by using at least one external cameras on HMDs or outside environment[13]. The augmenting approach is similar to those implemented on AR methods, but it differs to have background as digitized virtual graphics instead of real textures, which usually requires close-by display devices such as HMD or special optic devices for segmenting reality from the bare eyes of the users. Many researchers suggested different ways to seamlessly integrate the virtual on the top of the reality. Numerous applications under this term was developed for games, simulations, engineering and education, with interactivity.

The AV did not get much attention as a significant virtual environment experience compared to AR. The reason for that is that there were more technical obstacles to break to set the VR into the baseline environment. Those obstacles include a low-resolution display, a tethered VR device

platform which works as a limitation for the usage in outdoors, and difficulty in tracking real objects.

The approach for real image representation in AV platforms differs based on two factors which is amount of blending and the moment of representing. The levels of blending reality onto VR was first introduced by McGill et al.[84]. Full blending, partial and minimal blending is the levels of the mixture between the reality and virtual reality. As full blending shows no virtual elements on the display, partial blend shows key objects via real vision whereas minimal blend shows very small area of reality through a small window. Metzger[86] suggests the inset blending, which is overlaying a small window of reality at a fixed location in virtual environment.

For the second factor for AV, which is the moment of initial image display is categorized into two groups; user-initiated and inferred. User actions such as pressing a button or grabbing an object can be as a trigger signal for the user-initiated process. The moment for visual import is requested by the user at the VR scenario, whereas inferred method requires system itself to attempt to predict inquiry at right timing. The prediction can be calculated by scene change, measuring task solving timing points. In McGill' and Budhiraja's work[18], a user survey was conducted to see the user preference on the initial moment representation method, and participants preferred the user-initiated method over the system prediction as they could express their intentions more efficiently.

The early approach for implementing AV platforms started as using external camera installed on the top of the HMD device. The SIMNET[21] is

one of an early work that combined a real element into a virtual scene. A head mounted camera was placed on the HMD to provide a window to the outside from inside virtual environment, giving users an ability to visually perceive objects placed outside their HMD. Steinicke et al[103] used chromakeying technique to import user's hands into the scene by also using the camera outside the VR interface. This method is a switched way for capturing real image as the direction of camera is reversal compared to the previous method. This method was further investigated by Tecchia et al[112] by adding additional depth tracking capability to the platform. They used a motion capture platform to track multiple markers installed on the HMD and user's fingertip rings to create an interactive platform that could lead users to manipulate virtual objects with their actual hands. With modern development of computer platforms and the widespread of low-cost mobile based HMDs, studies to utilize the AV for application were presented for various scenarios including video games[30] and education[113]. However, most of studies took an approach to focus overlaying whole player body instead of utilizing objects in outside context[90].

Recent solutions often involve Leap Motion controller[140] which is originally designed to track users' hands and fingertips, by measuring wavelength of infrared (IR) waves with binocular cameras with IR LEDs. Its original purpose was to track human hand and fingertips and represent virtual hands based on the tracking data in VR environment, but the raw image from the cameras can be utilized to augment real image to AV scenarios even if it only provides black and white images. A consumer device from Microsoft has

also similar purpose but expandable to track the whole body, is used in some researches[91] for augmenting real vision in VR.

Similar approach to our work in which is to utilize outside object as medium to order to smoothen the isolation problem in virtual reality are the SDSC[36] by Desai et al, and NRAV[2] by Alaee et al. Both implementations import a mobile phone to a VR scene and let users recognize the outside context with the real object. In the research of Desai et al, they used Smartphone Detector based on a Statistical Classifier (SDSC) method with a LeapMotion controller. The approach resulted in high accuracy in detecting smartphone within the field of view (FoV) of an HMD user but had inability to show the screen in real-time without a minimal delay as they were capturing the stream of screenshots instead of live stream of the visual data as the image representation. In NRAV system, they also represented the real image of mobile phone into the virtual scene to smoothen the isolation in VR and let users utilize their phones. In this research, the phones were used with the same purpose as it had in real life, in the case of adding an attempt to utilize as an input device for VR.

All researches mentioned above have been shown at least one method to augment certain physical object located outside into VR using external monitoring devices with or without tracker combinations. It is proven by earlier researches that the AV area has a potential to provide an immersive virtual reality experience by using the advantage of modifying synthetic background, yet limitations still exist in the methods. First, it lacks directly importing the precise original texture of the physical object into VR. Even if

the physical object is used as a simple input device in[132] VR, the texture of the object is replaced with synthetic image, as extracting the object texture alone from the outside image is a challenging task. Second, some methods involve importing information displays for providing additional augmentation for VR, no real-time communication was available, making it impossible for the usage for input devices for VR. And finally, due to resolution problems in most cases, interacting with the object was nearly impossible while the interface has a high-density UI such as soft keyboard.

In this study, we focus more on importing objects from outside for the purpose of using them as input devices in the virtual environment. And also, we enable real-time interactivity with the mobile device by implementing efficient image representing method with minimized delay and confusion. A method for efficient and precise image representation in AV environment would also be discovered by various experiment setups and tasks.

2.3 Inputs in Virtual Reality

The input in VR takes an important role for interactive, a richer sensory experience to users. In the study of Burdea and Coiffet[19], the three I's which are interaction, immersion and imagination are mentioned as key elements for virtual reality experience[17]. Among the three factors, interaction can be established by capturing users' intentions at right moment then reflect the signal to control the virtual environment. Input must be supported in some way in order to accomplish the immersive scenario.

Following recent advances in VR technologies, many researchers have been conducted and commercial systems have been released various interaction techniques for enabling different functions and types of inputs for VR. In this section we group the methods into two, those that does not require manipulation devices and those actively utilize mechanical devices to operate the environment. In the following sections we explore diverse approaches for solving limitations in input in VR and other Mixed Reality realms.

2.3.1 Methods with Passive Expression

Passive expression of input in virtual environment involves methods to monitor user simultaneously for capturing pre-defined status change that would be utilized as trigger / input signal for manipulation in VR. Those expressions include body movement, gesture, eye-tracking and voice recognition[6]. Maggioni[82] introduced hand gesture input system by using outside RGB camera to monitor user's hands and segment hands image, then estimate the gesture to be utilized in 3D environment. This hand tracking approach was followed by adding depth to the awareness[96][119] and more precise tracking even with finger tracking. The idea of capturing the hand posture and gesture was mature enough to establish a solid input platform, but most systems at that era were difficult to implement and lacked in accuracy[76]. Finger touch input was also used as input signals in MR scenario[129][123][72] by enabling fingertip gesture recognition when a user approaches a surface. These methods rely on pre-acquired background models

for the finger meeting surface making it difficult for variable moving surfaces. Not only hand but extending to virtual arm approach was also used[93]. These techniques were based on the metaphor of being able to change arm length under user's intention. Expanding the tracking method from single body parts to full body was also explored by Latoschik[75] and Caserman[23] by representing full-body avatars in virtual environments. Under tracking full-body platforms, human articular surface tracking works as the basis of all motion recognition. The main focus of this area was extraction of skeleton and as the depth camera from Microsoft supported certain limitations in tracking numbers of articular surfaces, recent methods are usually consisted with 25 joints[78]. Tracking the body parts and full-body methods divide into either they are using the passive marker (e.g., ARUCO) or not. Markerless tracking systems usually utilize binocular cameras or hybrid cameras with IR sensors in order to retrieve the depth data of the image in 3D environment. Tracking full-body in scenarios of exergames resulted significant evidence of motivation and fun[74], but the sudden tracking loss, unreliable data[45] was a challenge to overcome. In those passive methods, the recognition process was calibrated through segmenting the body parts from the real image and classify them through definitions after the tracking process[130].

The methods to track small body parts(i.e., hands, fingers) from third person perspective were not able to be used in scenarios in which required a high precision level of operating small objects, as they generally respond to distinguishable big hand movements or big delta values between the image frame maps for posture detection. Also, vision-based tracking approaches had

latency problems as they had long process of posture classification. Ranging from seven to approximately 300ms[75][37][66][63][104][115][64], the delay in synchronizing the user movements in real world with those in virtual environment was carefully considered among all studies to minimize the risk of cybersickness symptoms in VR[29][109] and not to deteriorate the sense of immersion[46]. Additionally, the lack of haptic / tactile feedback in passive expression methods created problems as the latency combined with vague feedback often led to user confusion and low level of user usability as the motions were almost mime-like.

Gaze input is another input method without physical device, often involves eye-tracking[92]. This interaction technique utilizes user's direction of gaze as the input signals. Deecker and Penny[35] identified six common input information types for graphical user interfaces (position, orient, select, path, and text entry) and in the gaze input method, one gaze would operate both terms; position and select. Longer fixation to an object performs the select. This simple method was also implemented to modern smartphone-based HMDs, on to systems even without the eye tracking support. A simple workaround for this to work was to set the center of the screen as the initial position cue, then utilize rotational values of the device as a directional modifier, which are acquired by 3-axis gyroscope sensor of the device. This allows to substitute the actual gaze direction with head movement, creating a simpler implementation with a restriction barrier for errors for random eye movements.

While utilizing the head-movement as workaround for gaze direction, the

predictive model from Fitt's method[43] which is actively used in other areas of human-computer interaction is used for measuring the time performance of gaze input methods. The model is summarized as showing the direct proportional correlation between the movement time (MT) and distance traveled and opposite with the target size (Equation 2.1).

$$MT = a + b \times \log_2 \frac{2D}{W} \tag{2.1}$$

Where intercept a and b are coefficient determined by the properties of the input device, while:

$$D = Distance\ Traveled$$
$$W = Width\ of\ the\ Target$$

Several studies[106][16] utilized the index to measure the correlation between the time for selection and the difficulty in the gaze input system.

The limitation of gaze input combined with current generation HMD could be mentioned as losing accuracy overtime. As the systems utilizes relative head positions using 3-axis gyroscope sensors, recalibration is required in certain periods of time due to slippage of the sensor values. Maintaining center becomes harder after usage due to noise from outside world. Other methods utilizing eye-tracking methods mainly suffer from delay, lost gaze position, and out of envelop[108]. Out of envelop refers the wandering eye that moves to the target in a curved path, prone to error eye

movement that is out of the prediction trajectory model.

Other passive input approach includes the speech command system[22]. An example of the system is a web-based police training tool[58] to use the voice commands as input for controlling a virtual robot in VR. As the accent would vary from user to user, it requires a pre-recording session for system to learn and classify exact commands for voice recognition. Although the large vocabulary continuous speech recognition (LVSCR) decoder used in the research is based on word N-gram and context-dependent HMM and claims to be able to perform in real-time, misinterpretation or noise often made the system unstable for controlling input. In these word-based speech input systems, only simple chunks of pre-selected words could be utilized as inputs and required controlled environment as the noise worked as a critical failure point. Different options for input initialization time point including automatic command interpretation (words spoken without confirm action), press confirm (confirm action after command auto-interpretation) and press to talk (interpretation initializes after confirming) were tested to minimize errors.

The passive input expression methods in VR mainly focused on detecting the trigger points by letting machines or algorithms automatically using event classification methods in visual / aural signal processing technologies. Often these automated systems interpreted human input signals wrong, and this lack of accuracy. The additional latency resulting from calculations for the action classification in abovementioned methods led to a low level of usability and lacking intuitiveness in control for VR.

2.3.2 Methods with Physical Devices

Posture and gesture tracking platforms that require outside cameras for monitoring often suffered from the occlusion. This led to a loss of tracking while recognizing the motion of the user and eventually made the platform unreliable. As the problem was coming from a passive, third eye perspective way of monitoring, numbers of studies focused switching the perspective into first person view by utilizing physical devices designed to be attached on the user or to be held by the user, rather than utilizing monitoring devices placed outside of the players.

As the technology evolved from two-dimensional (2D) based virtual reality to modern three-dimensional (3D) based virtual environment, the demand for input devices exclusively for 3D also enlarged. However, until up to recent years, there seemed to be a congestion in technology due to lack of standardizations of the devices in consumer market. People still utilized 2D based traditional input devices such as keyboard, joysticks, mouse and gamepads for the manipulation / interaction devices in the virtual environment. To break these limitations and to provide more immersion different techniques have been introduced for over two decades to support specific motions with multi-DoF in 3D world including translation, rotation and scaling (TRS).

Device-oriented input methods showed different types and forms in implementation including hand-worn devices[42], hand-held wand style controllers, and non-universal unique structures for specific contexts[15]. Input methods involving physical devices mainly focused on egocentric view

of the user, as it enables the perspective match from outside to inside as well as a direct manipulation. Hand-worn and hand-held input devices are usually universal controllers for VR contents, meaning that they provide diverse input methods compared to structures designed only for a single, specific scenario.

The concept of hand-worn methods is using natural hand as intuitive input device for VR. In order to gather intended movements, different combinations of sensors were used in many researches. Most researches and commercial systems utilize inertial measurement unit (IMU) to measure the specific force of the moving hand and to place relative position. The movement of fingers are tracked by flex sensors by tracking the amount of deflection and bending data. Some approached extended the tracking range up to the upper limb of the body by attaching additional flex sensors on elbows and shoulders[54]. Even though most of them provide tracking capabilities for representing the hand positions inside the VR and were able to gather inputs, they lacked in term of sensory information from the touch or physical contact. As the human action of touching corresponds to a bi-directional sense, hand-worn input devices with haptic feedback ability were introduced[122].

The wand style controllers are refereed as the standard physical input controllers for commercial HMD systems including the mobile based simple HMD platforms recently. As mobile HMD systems are focusing on the simple setup without tracking and convenient stereoscopic vision delivery, the wand controllers included in these packages only support 3-DoF, meaning that they only work as pointer without supporting any interaction or object reaching in the virtual environment.

Recently released sophisticated PC-based HMD platforms include their dedicated hand-held controllers in their packages. As most of these platforms now support motion tracking by indoor positioning systems (IPS) by using array of IR based transmitters / receivers, their controllers are also tracked in the installed environment. This enables 6-DoF of the controllers, as cartesian coordinate data of the controllers is known inside the virtual environment. Unlike previously mentioned 3-DoF controllers, these platforms can let users to give inputs or interact with the objects inside the virtual environment as well as to have the functionality as pointers. To ensure the expandability and variation in input, they are equipped with multiple clickable buttons, axis input touchpads and integrated with countless number of sensors for acquiring positional / rotational values of the controller. For most of 3-Dof and 6-Dof hand-held controllers, ray-casting technique is utilized to point and select an object. The direction of the ray is specified by the user's hand. In the case of combination with physical controller, the point of the ray (virtual laser) is attached to the very end tip of hand-held devices.

Additional issue with these physical controllers include that these systems are not interoperable among different platforms or contents, and often claimed to be too complex, acquiring a high level of learnability and not being user-centric[11].

2.4 Objects in Virtual Environments

In mixed reality, typically in AR scenarios, as real objects are visible to users

manipulating or operating the object could be considered as a very natural action. However, it becomes a different story in the realm of recent occluded HMD platforms, as the visual isolation occurs in the platform. With the precondition of the HMD based virtual environment, which is a blocked vision, it is not able to monitor outside. Thus, it can be defined that the nature of the system is not designed to encounter physical objects within the context[83]. The inability to encounter physical objects often causes a lack of presence and perception as it is impossible to match the existence of objects inside and out. Even if a touch or contact with virtual elements occurs, the sensory feedback is unable to be delivered to the users. To amplify the immersion by improving sensory perception, numerous researches on aligning the existence of the virtual object with the real object have been explored. Generally, the object existence aligning method in virtual reality can be categorized into two groups[110]. Those that involves physical objects, and those that simulates the touch or contact perception by using passive haptic feedback devices without the actual object itself.

Actual utilization of physical objects involves tracking methods such as vision monitoring from the outside for detecting the target and represent it inside the virtual environment session. The purpose of the physical object for these methods are either to utilize a proxy device to deliver feedback and enable tangible interaction, or to use the outside object as it is. Proxy objects are designed to fill the missing gap between object and user hand, to provide shape, texture and tactile feedback when user has touched or grasped certain object in virtual reality. Zhenyi et al.[52] introduced haptic proxies with the

assistance of motion tracking platform to track the object. The objects were called robots and it was designed in different shapes for providing similar representation with the object inside the virtual environment. As the platform was able to track multiple objects, users were able to use the robots while being immersed in VR. However, as the type of objects represented in virtual reality can hugely vary from different scenarios, researches on scalable tangible objects were introduced. Zhao et al.[135] presented a scalable physical object consisted of magnet based square shape blocks. The users were able to assemble the blocks to match the shapes of objects inside the VR, but the implementation lacked in spatial resolutions as the blocks used had square shapes, and each was too big for the precise perception of the object shape. VirtualBricks[4] is another device that is scalable, modular system that could provide physical manipulation in VR. Custom designed LEGO bricks were used to deliver similar tactile feedback to users by assembling modular bricks to mimic the shape and scale of the virtual element. The higher accuracy in shape perception was archived as it had higher resolution in terms of modular brick size. Additionally, the modular system had a dedicate channel for data transmission for object status change. By this, it could maintain a lower level of error in input compared to other vision-based tracking systems. But the ability for motion tracking was limited to single-axis rotation and single-axis linear translation. Aside from scalable proxies, Shifty[133] is a hand-held rod-shaped dynamic passive haptic proxy that has variable weight inside the rod with pulley. By moving the location of the weight inside the rod, the internal weight distribution was able to be modified

to improve perception of the virtual object. The object was represented as long rod-shaped things in VR, such as a sword and a baseball bat.

On the other hand, Paperstick[47] is an example of utilizing an outside, offline object without any feedback implementations. It is imported into the VR scene and used as a control device, but the texture of the object seen from the VR remains synthetic, lacking in real-image visual cues and having a limited ability to deliver visual feedback to users. Yoshimoto and Sasakura implemented virtual reality tower defense game by assigning real objects with visual markers as “tower defense guns” and letting users modify the trajectory of the gun in the game by rotating the actual hexagonal object[132]. The rotational values of were tracked by the external camera, but initial visual cue was ignored in the monitoring session leading to a fact that users could not recognize the initial position of the object outside their field of view unless it is not held in their hands. Additional lack of hand tracking and seldom loss of tracking rotation created low accuracy in the platform.

The methods in mimicking touch feedback without physical objects mainly focus on the quality of feedback to users using different methods. The goal of these systems is to deliver a real like tactile perception to users by passive haptic implementations. These approaches sometimes utilize attachment devices to user hands[141][142] to provide proper simulated feedback to users while they are interacting with object inside the virtual reality. Passive haptics using force feedback from electrical muscle stimulation has been introduced by Lopes et al.[79][80]. Instead of stimulating hand using gloves, the electrodes are applied on users’ triceps muscles for

stimulating sensory with force feedback when user encounters virtual object. Results in the experiment showed that it improved perceived realism and also showed the potential expandability to be utilized in MR scenarios.

Body movements are also involved in some methods to attach sensor arrays on the body part of the user to provide feedback[125]. However as mentioned earlier, those methods cannot encounter physical object and their perspective and their tactile still remain in the virtual world. Especially on gloves that utilizes grasp gestures[28][27] suffer from mismatch from expected haptic feedback and the actual sensory input as those methods cannot mimic all kinds of feedback from a real experience of object hand manipulation. Additionally, methods using EMS may cause muscle fatigue and the actuation of user hands is typically limited to a single dimension of translation.

In this study, we accomplish methods to overcome the two structural problems that current object involvement in VR platforms have, which is blindness and simulated touch perception. By opening a window to the existing HMD device and efficiently represent the object portion of the image, authentic tactile feedback can be delivered. From this, direct manipulation becomes possible with a clear recognition of the outside context with a reduced risk in physical portion of the virtual isolation.

2.5 Summary

Current direction of VR development focuses on enlarging the immersion and

real-like experience in the name of “immersive virtual environment experience”. As the demand for the platform mainly comes from VR games and simulations, it could be reasonable response from the academia and industry resulting in more methods for more sensational and dynamic factors. However, it is inevitable for a regular user to encounter real world during the experience and ignoring the virtual side impinges on overall virtual reality experience for users. The cost for switching between the virtual reality and reality is high, as users must take off their HMD devices in order to interact with real world objects. Simple actions taking a glance into a mobile phone, taking a sip of coffee are not simple anymore in VR, it becomes expensive tasks in terms of the switching costs because taking off the immersion device breaks the presence of virtual environment.

In order to create a virtual reality platform with natural interaction without breaking the presence factor, we believe that interacting with outside objects while being immersed in HMD VR is a crucial key factor. An Augmented Virtuality system that allows a ubiquitous, seamless communication via an object is most promising solution to the current problems of VR platforms.

Previously introduced methods in AV term lacks in three conditions. First, they could not import the original texture of the object into the virtual environment, or even if they could they lack in image quality. Most methods substitute synthetic texture to represent the object inside VR. This also led to a problem that users’ hands are not visible inside, making it difficult to let users recognize the initial position of the object. Second limitation is the lack of

abilities for simultaneous data transmission with outside objects. As gathering signals for status change of the outside object was not possible, it was impossible to use the object as interactable device. Objects with LCD screens which were utilized in some methods, were only able to work as additional information displays for users. And finally, most methods could not respond to the necessity for precise input control. The existence of resolution and latency issues made limitations for methods to be utilized in scenarios which required to control high-density interfaces, such as soft keyboard layout UI. In this dissertation, we implement our augmented virtuality method to use an active object as an input device for virtual reality, using it as a medium for linking the real world and the virtual reality. We compare our method by evaluating the effectiveness in terms of performance and presence in virtual environment, aiming to establish guidelines for the design of AV environment and explore gaps in the current literature. The remainder of this dissertation presents the investigations of our methods and findings.

Chapter 3. Importing Active Object into VR

3.1 Introduction

In this chapter, we describe our method to import a real active object into virtual environment and the process of testing the system with user studies. The proposed method was developed under the term Augmented Virtuality (AV), which refers to the condition that physical object in the real world is visible inside the virtual reality. The AV has an advantage as it could use the flexibility and the expandability of the virtual environment while still being able to encounter physical objects placed outside, preventing the isolation of occluded virtual environment. For the first attempt in implementation, an additional RGB camera was attached on the top of the HMD and used as outside object detector, and when pre-defined object appears in the Field of View (FoV) of the user, the object was pattern-matched and then segmented then overlaid in the virtual scene while users are immersed in VR, wearing HMD. While using the pattern-matching technique, the pre-defined object can be anything, if it has a texture recognizable by the algorithm. This time, considering interface familiarity to users, ease of communication and convenience in development with the VR engine, we chose an “active” object with its own power source, an Android-based mobile phone. Using this method, we provide a more user-friendly input with ease of learning curve and more intuitive control, as well as reducing the isolation problem as the representation of outside world opens and windows for users while they are

immersed in VR. After presenting the results, discovered limitation is addressed, then we present our methods to further investigate the problem to resolve the existing issue. A separate preliminary test aside from the main experiment was conducted in order to figure out the problem. The results from the test imply that the issues are not only solved but shows a potential factor to establish a better platform with a significant improvement.

3.2 Platform Design

We explain our overlay method system with details in this section. First, we briefly describe the system design of the proposed method. In the following subsection, we explain the process we went through for implementing the pattern-matching technique to recognize a target object. Finally, we describe the specific configuration and how it integrates all together.

3.2.1 Architecture

Figure 3.1 illustrates the overall architecture and elements of the system. Our system mainly consists of four elements which are the HMD, the 3D rendering engine, the RGB camera and finally the target object, which is a mobile phone in this prototype. The RGB camera is attached on the top of the

HMD as shown in the right side of Figure 3.1. The camera mounting position is calibrated to be aligned with the center of the HMD, for the horizontal field of view (FoV) of the RGB camera would stay inside of the HMD FoV all the time. This enables to deliver the image of the target object

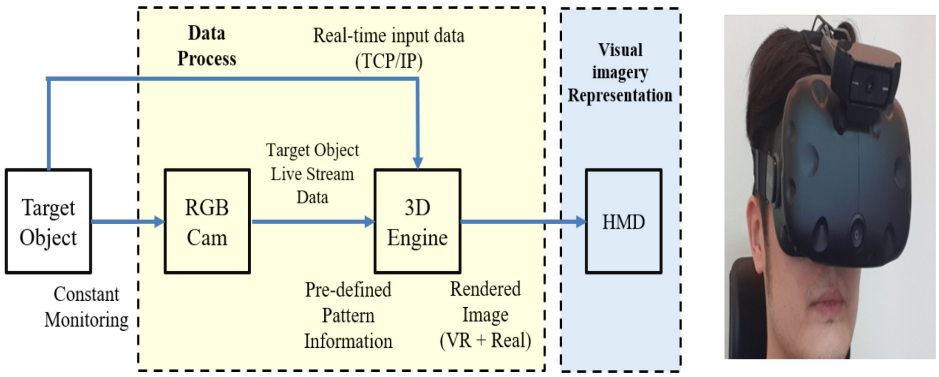

Figure 3.1: Overall architecture of the system (left) and the HMD with the mounted camera(right)

to users even in scenarios which users turn their heads side to side to change facing directions. Also, the vertical angle for facing direction for the camera is set roughly to 130° degrees downwards as we assumed that the target object will be placed on the under area of the HMD where the users' hand would be placed. The RGB camera is a Logitech C920 webcam which has a vertical FoV of 43.3° degrees, and even if the placement of the camera sometimes fluctuates due to different style of HMD wearing positions upon different users, initial test showed that there wasn't any serious issue reported related to this slight angle misplacement. The pre-defined target object texture data is stored in the 3D Engine, which is Unity 3D. In the engine, pattern-matching scripts are set to constantly check on every frame if the pattern exists in the current frame. If detected, the system overlays the detected object into the current playing scene inside the FoV of the HMD, otherwise it does not show. When represented in the VR scene, users are able to see the actual phone and interact with the device. A Samsung Galaxy S9 Android phone acts as the

target object in this method. The real-time input data from the users is simultaneously transmitted to the Unity 3D engine by a custom Android app.

3.2.2 Visual Pattern Matching

We used OpenCV[1] for processing the simultaneous visual imagery of the target object. OpenCV is an open-sourced library functions mainly aimed at real-time computer vision. Unity 3D mainly supports development environments based on Microsoft C# or Java Script. Since OpenCV does not natively support those languages mentioned, a modified wrapper version on C#, OpenCVforUnity was used. Among various pattern matching algorithms such as ORB[99], SIFT[81], SURF[7] and AKAZE[3], we decided to utilize the ORB as it is known to be fast and rotation invariant[65]. ORB is a fusion of FAST[98] keypoint detector and BRIEF descriptor with modifications to improve the performance[20]. It uses FAST to find keypoints at first then to find the top N points, a Harris corner measure is applied. As FAST does not detect rotation, it computes the intensity weighted centroid of the patch with located corner at center. The direction of this vector from corner point to centroid gives the orientation. An ability for fast detection with rotation invariance and partial scale invariance was efficient enough to be applied to our method balancing between the performance and the delay. We modified the maximum number of features to be retained, which is the nFeatures and is set to 800. In the very edge of keypoints, four corners (top left, top right,

[1] OpenCV. https://opencv.org/.

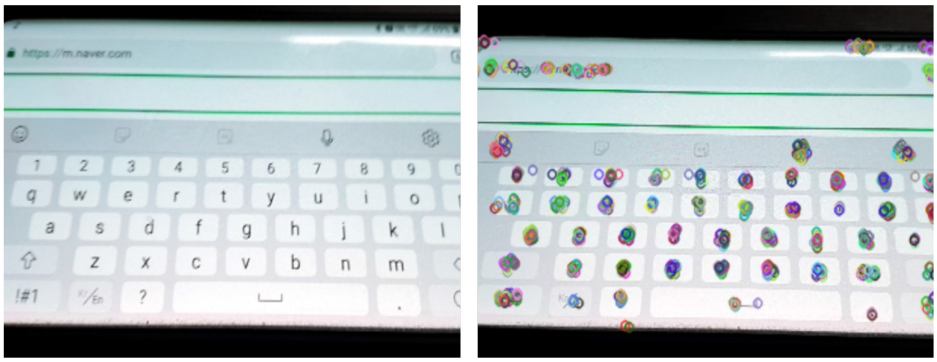

Figure 3.2: An example of a keypoint detection original (left) showing detected keypoints (right)

bottom left and bottom right) are created and put into a new image matrix of the size of the target object then represented as a rectangle with cropped RGB input matrix of the current frame. The rectangular image of the target object is represented in the VR scene. Figure 3.2 shows an example of a detected keypoints in pattern matching technique, whereas Figure 3.3 shows target object, with the trace of the target using the recognized patterns. The actual trace visualization is not visible to users in real usage scenarios. Unlike other AR based implementations that mainly focus to track users' hand or arm exclusively, our method did not consider those elements and focused only on the target object.

After the initial implementation, an additional idea was considered. Since there are no visual cues while "holstering" the phone (e.g., pockets, outside the camera FoV) and to help the users to see the object always straight, a perspective wrapping method was applied. This enables to see the straight top-down view of the image regardless of the rotational angle of the target.

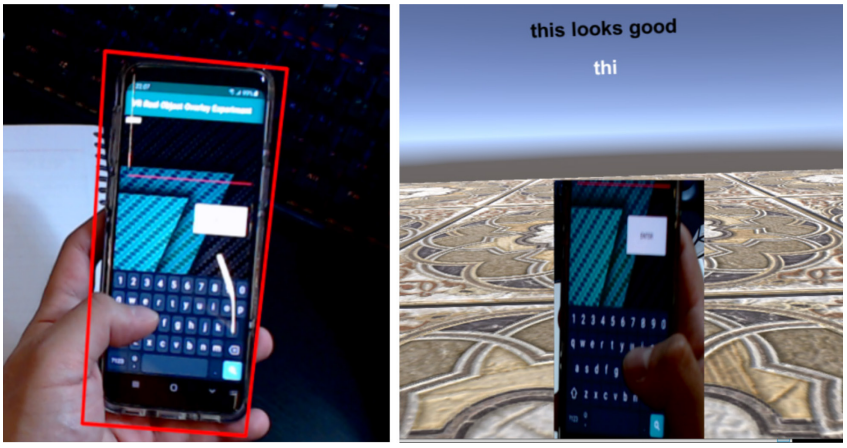

Figure 3.3: An example of a target object in sight with trace visualization (left) actual representation in VR scene (right)

However, there were some issues with this method discovered on the first pilot test. As the algorithm continuously checks the pattern and wraps perspective every frame in Unity 3D, we found out a problem that in some ambiguous environments with much of visual noise, the image represented kept flickering and twisting. The users reported that this problem caused discomfort and made them hard to concentrate on executing inputs. And also, they claimed that auto-correcting the rotation causes more confusion. This method was later discarded in the main user study and the method was switched back to the previous ORB based pattern-matching detection.

3.2.3 System Integration

All virtual imagery rendering, real world overlaying and input control were

managed on Unity 3D engine. The communication between the Unity engine and the Android device was implemented using TCP/IP protocol. Communication scripts for both platforms have been created in order to transmit input data from the device and show received data on the engine in real-time. With all components mentioned above, users are able to manipulate an external mobile phone as controller within a VR context. The live image feed from the camera was done at 720p (1280 × 720 resolution). We used 720p instead of 1080p (1920 × 1080 resolution) even the camera itself was capable of a higher resolution, due to delay caused by slow data pipeline of the Unity 3D. As we were utilizing default WebcamTexture API to deliver outside image to inside, it was difficult to maintain minimal frames per second (FPS) of 30 when using 1080p. Thus, to maintain fastest performance with maximum readability, 720p was selected. However, the methods to resolve this issue do exist, which will be discussed later in discussion. Nevertheless, the image at resolution 720p was still able to deliver an enough level of readability for users to recognize contents while wearing the HMD.

As mentioned earlier, a mobile phone is considered as an object without a texture in general, because usually it is not represented with continuous or repetitive visual patterns on its screen surface. To utilize the keypoint-based pattern matching technique in this situation, we designed the screen contents to have recognizable textures by the RGB camera. Then, fixed the LCD screen brightness of the target device to minimize visual noise. The pattern for the target object was pre-captured before deployment and saved in Unity 3D engine. In addition, a feature to save and utilize multiple snapshots of

different keypoint patterns was implemented to cope with scenarios with multiple objects, each with different texture patterns. During the VR play, all the pre-recorded patterns can be utilized inside the virtual environment and they are able to be switched in-between scenes to recall / detect the corresponding pattern for the context.

3.3 Task Design

The tasks for the experiment were consisted of three different sets. Each task was designed to investigate the usability and the performance of the proposed method upon different scenarios in VR which involves different types of UI elements and input styles. The structure and the placement of the UI elements were designed identical across the baseline and two overlay conditions inside the virtual environment to minimize a risk of perception gap and confusion across participants caused by the platform difference. In addition, we added slightly different variations and randomly picked performing orders for the same task during the experiment, to maintain a balance throughout the repeated measurements. In the following subsections, we describe the components and structures of the tasks in detail.

Menu Selection. Menu selection task was designed to utilize one hand. In a static physical location (sitting down on a chair), participants manipulated the menu UI accordingly to given instructions. As shown in Figure 3.4, we utilized commonly used software input components of graphics user interface

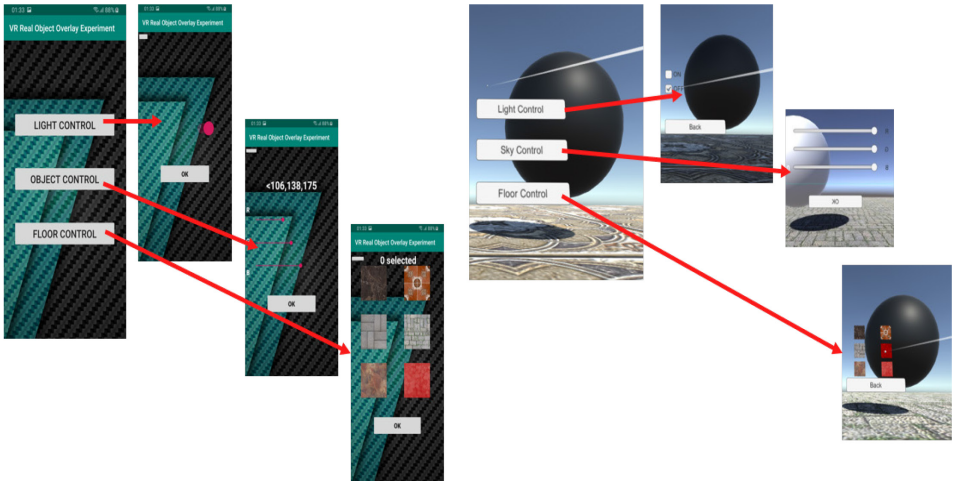

Figure 3.4: UI structure of menu selection task (left: Overlay, right: Baseline)

(GUI) when creating the task. And also, the menu had a hierarchy, as the main menu is separated into submenus. When the virtual scene started, the participants first entered to the main menu where three choices could be made to navigate to different submenus. The submenus had three small tasks: 1) Toggle switch manipulation for binary selection (on / off), 2) Three slider UI elements to give different number combinations as input and 3) Six clickable buttons with numbers for providing sequential press order. The participants completed all three small tasks, and an instruction for a task order was given to the participants one at a time in a random order.

The toggle switch was used for turning on and off the light of the virtual environment. The instruction was either to turn on or to turn off the lights in the virtual scene. The three sliders combination was represented as an “object color changer” to control the color of a sphere in the scene. Each slider had a range from zero to 255, and all together they worked as an RGB color

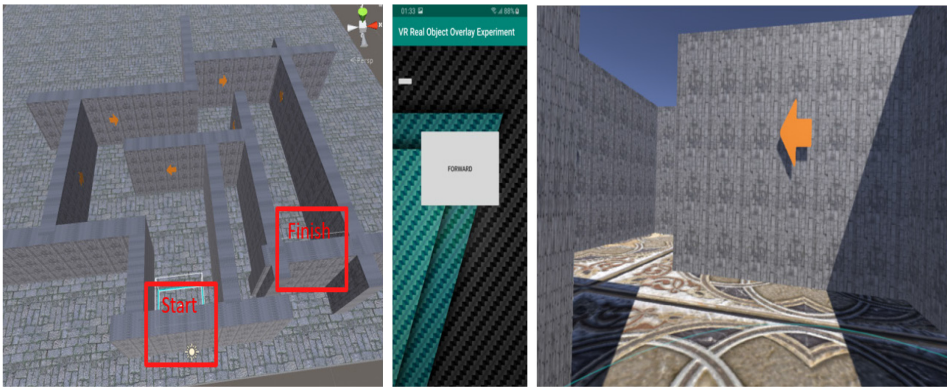

Figure 3.5: Visual representation of the Navigation task

modifier. Specific values of red (R), green (G), and blue (B) were shown in the instruction and the participants had to match the given RGB value combination by moving each slider with their input. Finally, in the six sequential buttons task, the participants were told to select series of buttons to a given number order (length of three). The time for each task completion, with the total time of duration was recorded. For the baseline condition which utilized dedicated hand-held controllers, ray-casting was used for positioning the desired UI element, and a physical button click for object selection. For conditions with our proposed method, Android touch UI elements were used.

Navigation. Navigation task was also designed to utilize one hand. In this task, the participants stood still in a static location and moved the first-person perspective 3D player by using controller to solve a small maze in VR scene (Figure 3.5). There were a starting point and an ending point in the maze, which involved six direction changes to complete. The navigation hints for solving the maze were shown in the VR scene, represented by arrows pointing

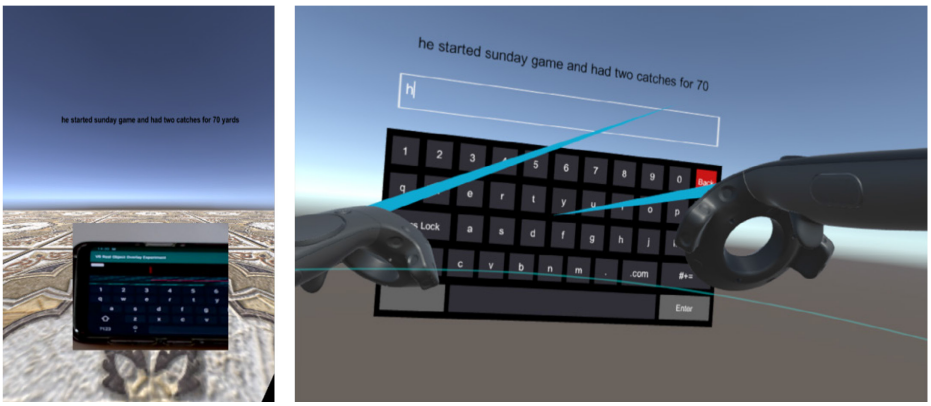

Figure 3.6: Visual representation of the Text Entry task (left: Overlay, right: Baseline)

the right direction. The total time of travel from the start to the end was measured. There was only a forward button for both the baseline and the overlay conditions, and the heading direction was determined by the facing direction of the participants. This is a scenario which the participants do not require a constant visual cue their control devices once they first get used to. This simulates a traditional direction input device such as a joystick or a keyboard with a simple mapping set such as WASD, which are easy-to-reach, and always in user's hand. In this one-dimensional input task, this could be executed by a simple movement of finger by pushing the same button (the baseline: physical trigger button and Overlay: Android touch UI button) repeatedly.

Text Entry. Text entry task was a two-hand operation, and the participants were asked to type to five stimulus sentences one at a time. A different set of

five sentences were randomly drawn for each condition. As the visual feedback to the participants, the response text was shown in the VR scene with the stimulus sentence (Figure 3.6). The current text view was placed below the stimulus element. For the baseline condition using Vive controllers, a keyboard 3D UI element with QWERTY layout was constructed. And to operate the keyboard, we chose the ray-casting technique. The rays came from both controllers to point the keyboard element and the participants used trigger button to select the highlighted key. For Overlay conditions, we used Google keyboard for Android combing with a custom app delivering text input in real-time via TCP/IP. Automatic auto-correction feature was disabled for this app. The measurements were word per minute (WPM) values. Five consecutive characters were counted as a word, including spaces. Error rate was measured as character error rate (CER), which is the minimum number of character-level deletion, insertion and substitution operators required to match the response text to the stimulus text, divided by the total numbers of characters in the stimulus. Stimulus sentences were acquired from the mobile phrase set[117].

3.4 Experiments

3.4.1 Conditions

To test and examine the feasibility of our proposed method in terms of usability and performance as an input system against existing VR dedicated

hand-held mechanical controllers, we conducted various experiments with different conditions followed by a variety of input tasks described in the previous subsection. First, three conditions for the experiment were created as follows:

- **The Baseline:** The Baseline condition was created by using HTC Vive VR controllers. Participants used either a single or dual controller to complete the experiment, according to the task they were given.

- **Overlay Normal:** The real image of the detected object (mobile phone) was delivered to the VR environment when detected. Participants then manipulated their mobile device to complete the task using one or two hands depending on the type of the task they were performing. The size of the object shown through the HMD, was set to be perceived as a real-world size.

- **Overlay Large:** There was no difference in keypoints nor registered patterns from the pattern-matching technique between the Overlay Normal and Overlay Large conditions. The only difference between the two was the represented scale of the overlaid image to participants. The scale was set to 150%.

In the Overlay Normal condition, the size of target object was calibrated to be perceived as a real-world size whereas the Overlay Large condition was set to have 150% scale compared to the previous condition. The reason for

supplementing an additional condition in different scale of represented image was to further investigate the influence of the visual image size represented to users in terms of task solving performance, readability and usability. For the final visual representation for the participants, HTC Vive HMD was used for all conditions.

3.4.1 Participants and Procedure

We recruited 15 participants in this study and performed repeated measures for three conditions. The participants were 12 male 3 female, aged between 24 to 35 years. None of them claimed to be a serious VR HMD user, but they had experienced the virtual vision through the HMD at least once. All participants had zero experience of using VR dedicated hand-held controllers except one. Also, all of them were familiar with modern smartphone usage, and accustomed to using QWERTY desktop keyboard as typing interface. Before the experiment, all participants were briefly told about the operating methods for both controller and overlay object as the VR controller. Then they were asked to wear HMD and stare at a demo VR scene which was irrelevant to this experiment conditions to adjust inter-pupil distance (IPD). Regular HTC Vive controllers and overlay object were shown at the same time in the same scene, but participants were not allowed to interact, just to grasp the visual representations of the two.

The experiments were within-subjects design. Thus, the executing order of the tasks as well as the specific instructions were different in menu

selection task, orientation was modified for navigation task, and different sets for stimulus sentences at similar length were selected for the text entry to maintain the balance across all conditions. For all input process including menu selection and typing in across different conditions, the participants were asked to make inputs as quickly and as accurately as possible. For the navigation, the participants were asked to avoid any collision with the wall. After completion of each condition, the participants took five minutes of break while filling out a simulator sickness questionnaire (SSQ)[68] and system usability scale (SUS)[60] as feedback for each condition. At the beginning of each condition, all participants were asked to hold the controllers / mobile phone in their hands.

3.5 Results

For statistical analysis, we performed Friedman tests with an initial significance level at $\alpha = 0.05$ to find statistically significant differences among three conditions (the baseline, Overlay Normal, and Overlay Large). The post-hoc analysis was then carried out using Wilcoxon Signed Rank Sum test with Bonferroni correction to investigate a statistically significant difference between each condition.

3.5.1 Measurement Data

Figure 3.7 shows the average task completion time values of the participants

for each task as well as the total time of duration from each condition. In the first task (menu selection) with the total time, A Friedman test showed that there was a statistically significant difference among those three conditions, χ^2 (2) = 20.8, $p < 0.001$. The post-hoc Wilcoxon tests showed that the baseline had significantly slower seconds (median = 59.79, SD = 11.7) in completion of the tasks compared to the our method Overlay Normal (median = 40.39, SD = 8.79, z = 3.57, $p < 0.001$) and Overlay Large (median = 39.71, SD = 10.46, z = 3.26, $p < 0.001$). No significant difference was found between two Overlay conditions.

Coming down to the small tasks, first noticeable result was shown in the RGB color slider combination task. A significant difference had been found in the task, χ^2 (2) = 20.13, $p < 0.001$. Comparing the baseline (median = 36.84 SD = 8.15) to each Overlay Normal (median = 21.47, SD = 6.94, z = 3.37, $p < 0.001$) and Overlay Large (median = 20.46, SD = 5.78, z = 3.56, $p < 0.001$), our method showed faster completion time.

The results in the first task suggests that our system provides a better task completion performance as the task requires more visual attention and gets more complex. In addition, another difference was also found on the toggle switch task (χ^2 (2) = 12.4, $p < 0.005$). The difference between the baseline and Overlay Normal was reported as significant as z = 2.69, $p < 0.005$, whereas other comparisons between the conditions showed no significance. We assume this is due to a high IQR (Inter-quartile Range) found on toggle task of Overlay Normal condition. The median values for the baseline was 5.24 seconds (SD = 1.06), for Overlay Normal was 3.81 seconds (SD = 1.06) and

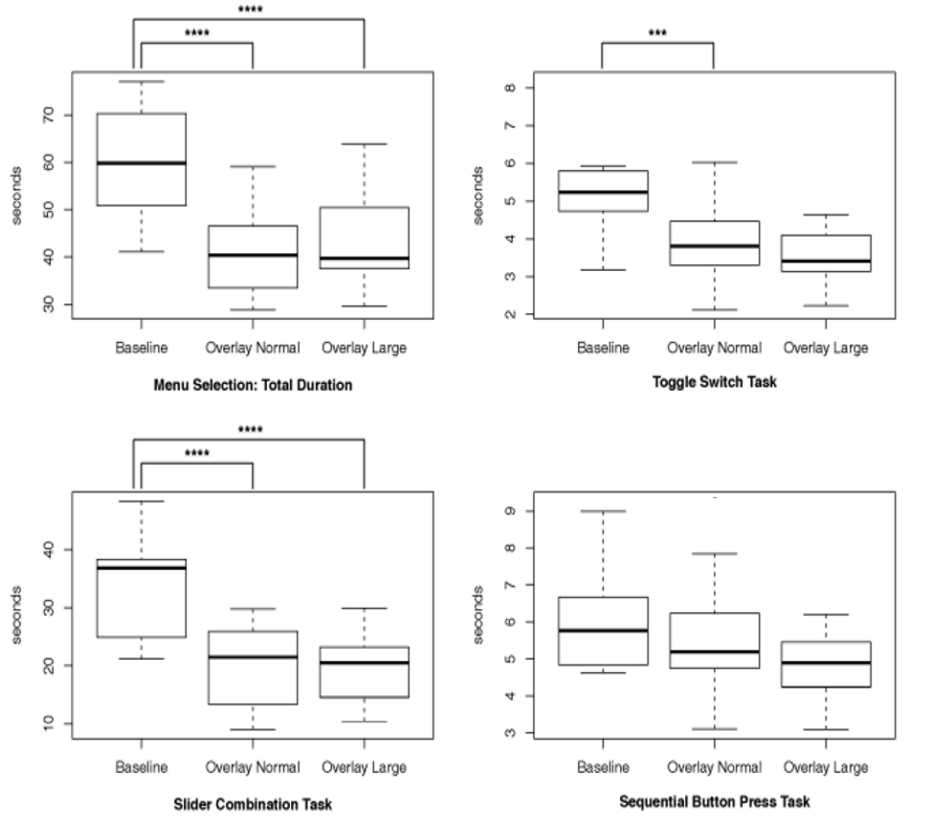

Figure 3.7: Menu selection time among three conditions

(* p<0.05 ** p<0.01 ***p<0.005 **** p<0.001)

3.41 seconds (SD = 1.52) for Overlay Large condition.

In the results of the maze task, we could not find any statistically significant difference among three conditions. The median completion time of the baseline was 36.01 seconds (SD = 9.12), whereas Overlay Normal condition showed 41.66 seconds (SD = 8.01), and 42.47 seconds (SD = 7.68) for the Large condition. Once the controller is held in participants' hand, a constant visual information was not necessary to move around the maze. We

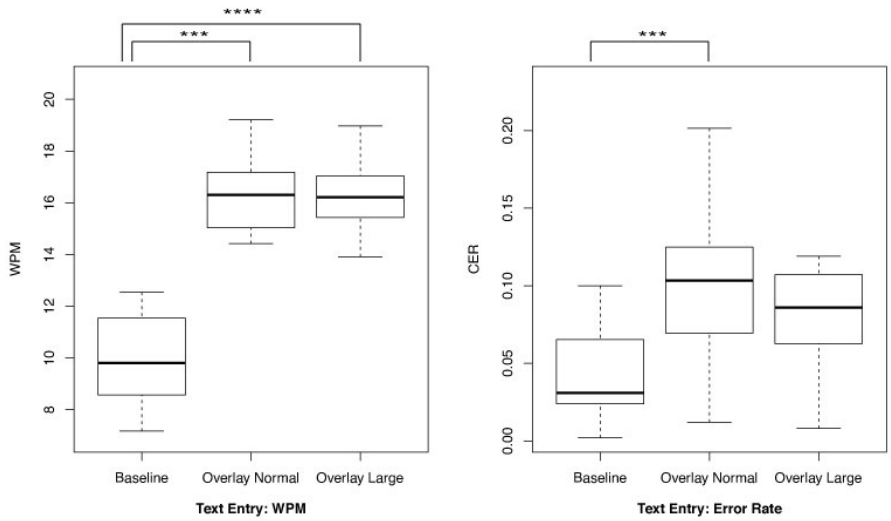

Figure 3.8: WPM (left) and CER (right) scores across conditions

(* p<0.05 ** p<0.01 ***p<0.005 **** p<0.001)

believe that is the reason for no difference. This is the kind of task that could be executed with the VR isolation which leans 100% to the virtual environment. The result suggests that our system did not show any advantages over the baseline on the tasks that does not require the intervention or interaction with the real world.

For the sequential button press task, we could not observe any reported statistically significant difference. The measurements were median at 5.77 seconds (SD = 1.28) for the baseline, 5.19 seconds (SD = 1.58) for Overlay Normal and 4.89 seconds (SD = 1.25) for Overlay Large. The task itself showed that it was too short to discover any difference nor participants reported awareness of difference among three conditions.

The results from the next text entry task is shown in Figure 3.8. The medians of Word Per Minute (WPM) and Character Error Rate (CER) values recorded from text entry task is shown in each plot. A Friedman test found a statistically significant difference among three conditions of WPM results (χ^2 (2) = 12.93, $p < 0.005$). The median entry rate WPM for the baseline was 9.98 (SD = 1.7), 16.31 (SD= 4.06) for Overlay Normal, and 16.21 (SD = 2.56) for Overlay Large. Post-hoc test found a significant difference between the baseline and Overlay Normal ($z = 2.7$, $p < 0.005$). There was also a significant difference comparing the baseline with Overlay Large ($z = 3.37$, $p < 0.001$) condition. No significant improvement in WPM score was found in Overlay Large condition over Overlay Normal. Even though no significant difference was found on those two conditions, the result imply that the Overlay method in general enables a better text entry performance over the condition with hand-held controllers.

In the results of error rates in the text entry task, another Friedman test was conducted with the data of the error rate measured with CER values. The results also showed that a statistically significant difference exists within three conditions (χ^2 (2) = 6.93, $p < 0.05$). The post-hoc Wilcoxon tests found that there is a significant difference between the baseline and Overlay Normal ($z = 2.69$, $p < 0.005$). The median CER for the baseline was 0.031 (SD = 0.04), 0.1 (SD = 0.04) for Overlay Normal, and 0.08 (SD = 0.05) for Overlay Large. Results suggest that our methods enabled participants to archive a higher WPM score, whereas also had a higher risk to make an error during entry according to the CER values. A possible reason to explain this result would be

the lack of the depth information of the real image inside the VR scene, which will be discussed later Section 3.6.

3.5.2 User Feedback and Survey

SSQ and SUS survey data was gathered from the participants from the survey they took on each five minutes break in between experiment conditions. Developed by Kennedy et al., the SSQ consists of 16 symptoms in three distinct clusters including nausea, oculomotor, and disorientation. Participants rated each symptom on a 4-level Likert scale ("None=0", "Slight=1", "Moderate=2" and "Severe=3). Overall SSQ score was calculated with combing all three clusters with corresponding weights. Only post immersion data was gathered this time. The results are shown fin Figure 3.9. It shows that the average total score is significantly lower for our both methods (Overlay Normal: mean = 24.68, SD = 21.52, Overlay Large: mean = 18.94, SD= 24.625) than the baseline with controller (mean = 49.61, SD = 43.87). A Friedman test result shows that a statistically significant difference among SSQ values from three conditions, χ^2 (2) = 18.926, $p < 0.001$. The post-hoc analysis presented that the baseline had significantly higher SSQ total scores compared to Overlay Normal ($z = 2.27$, $p < 0.05$) and to Overlay Large ($z = 2.66$, $p < 0.005$). The same pattern in statistically significant differences existed throughout the subscales of the SSQ.

To compare the conditions in terms of usability, SUS survey was conducted. SUS was created by Brooke[60] and provides a "quick and dirty"

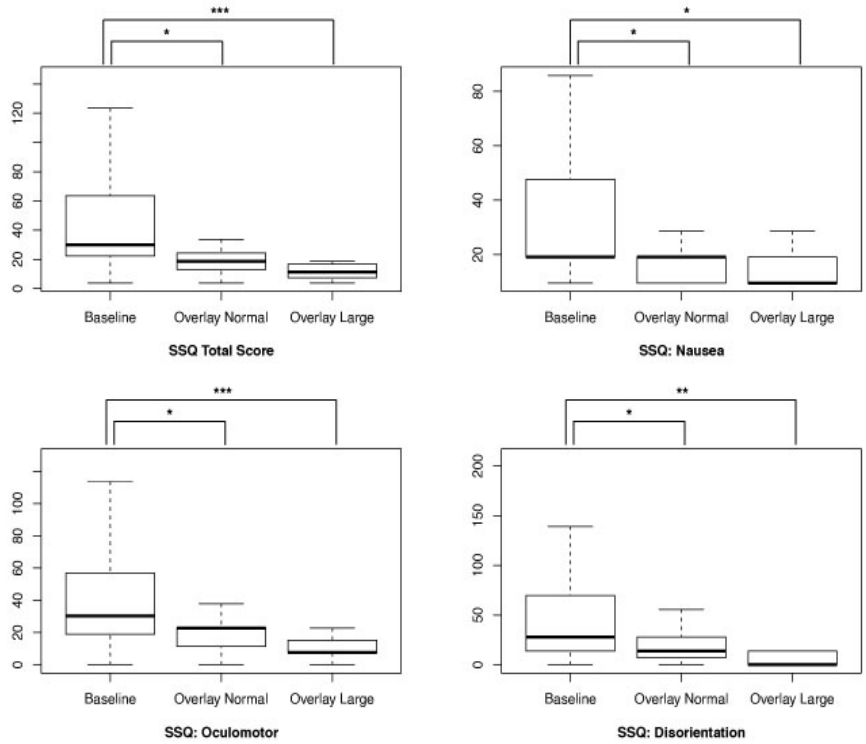

Figure 3.9: SSQ Scores among three conditions

(* p<0.05 ** p<0.01 ***p<0.005 **** p<0.001)

reliable tool for measuring the usability. It is consisted with 10 questions with a 5-level Likert Scale from strongly agree to strongly disagree. A Friedman test on the SUS showed that there was a significant different among conditions, χ^2 (2) = 18.429, $p < 0.001$. A Post-hoc test found that comparing the baseline (mean = 44.33, SD = 15.76) with each Overlay Normal (mean = 78.33, SD = 9.33, z = 2.8, $p < 0.005$) and with Overlay Large (mean = 80.17, SD = 11.11, z = 2.75, $p < 0.005$) was significantly different. Based on the research by Brooke, a SUS score of a 68 would be considered as an average.

Therefore, the result suggest that our system provides a better usability above average level, with considering all input types we have involved in this experiment.

No participant mentioned about the object detection accuracy. While monitoring data in real-time, the object monitoring session can lose its track caused by a high level of occlusion created in between the target object and the participant's hand, or with a random visual noise at the specific frame. This can cause a sudden disappearance of the target imagery in the VR scene if the track is not immediately resumed within a second, which are consisted of 30 frames. However, we assume this positive feedback was due to a simple filter to maintain the sight (keeping the window open) at the last seen point for three seconds in case when the tracking was lost. This made users to naturally move their holding positions in seldom "lost" scenarios, then the system was able to resume the tracking. Apparently flickering caused by the tracking loss often occurred during the experiment, but survey showed that it had a small to none effect on the overall usability of the system.

3.6 Stereoscopy in Real Images

After the experiment, we discovered that the current implementation had a limitation when delivering real image to users. The camera we used was not a stereo camera, rather a regular single-lens, RGB webcam. Thus, it was impossible to represent the image with stereoscopy. The result would be overlaying a flat 2D image without depth information onto a stereoscopic

virtual rendered image. There wasn't any serious issue reported while just observing the outside world or when manipulating the phone menu UI which had noticeably huge margins between interactable elements. However, when the participants started the text entry task and focused on the software keyboard which had near-to-no margin between the keypads, the issue had been risen. The participants reported that readability was fine, but the notion that there was a difference between the planned finger landing point and the actual landing point, often caused a confusion and a slight difficulty to focus. We assume this is caused by the lack of the depth information of the image, as a mismatch is likely to occur between the visual and motor cortex. The actual landing point of users' fingertips on the screen have high risk of not being aligned as they expected, as there is no depth representation. This explains the higher CER on both Overlay Normal and Overlay Large conditions on the text entry task compared to the baseline condition. We used Google keyboard as the software keyboard for the Android device, and the physical gap among different keys is less than 2 mm on the mobile phone we used. It turns out that this was a risky choice to use, as a great level of precision is required in order to accurately operate the keyboard, even with natural eyes. Against our expectation, the scale factor in the representation did not improve as we could not find any statistically significant difference between the Overlay Normal and Overlay Large in WPM and CER results. As some researchers mentioned similar findings in AR study[55], alignment of imagery may be a crucial factor in HMD-based environment when delivering visual information to users, as the precondition of the HMD is that it always provides stereoscopy

in virtual imagery by using its two separate screens. Aligning the real visual representation with the virtual information is required to minimize confusion and improve presence while being immersed in virtual reality.

In order to further investigate this issue, we designed a preliminary test for measuring touch alignment in two different conditions: with or without stereoscopy in real image representation in VR.

3.6.1 Preliminary Performance Test

To conduct the comparison test of touch input accuracy and user perception upon different visual imagery representation, we prepared two conditions with the monoscopic and the stereoscopic image representation. In this preliminary performance test, the only variable factor was set to image stereoscopy. The monoscopic condition utilized system setup from the previously mentioned system with regular single-lens webcam device, whereas the newly approached stereoscopic condition utilized ZED mini stereo camera which has two lenses designed to deliver offset in imagery, capturing two streams of images at different angles and each frame simultaneously. The final resolution of represented outside imagery inside the HMD was equally set to 720p (1280 × 720 resolution) which was the same resolution compared to the previous method. Even though it was possible for the new method to implement the visual representation with a higher resolution, we decided to match the previous conditions to minimize the difference resulted by resolutions. The pattern tracking method we have utilized in previous sections was disabled to

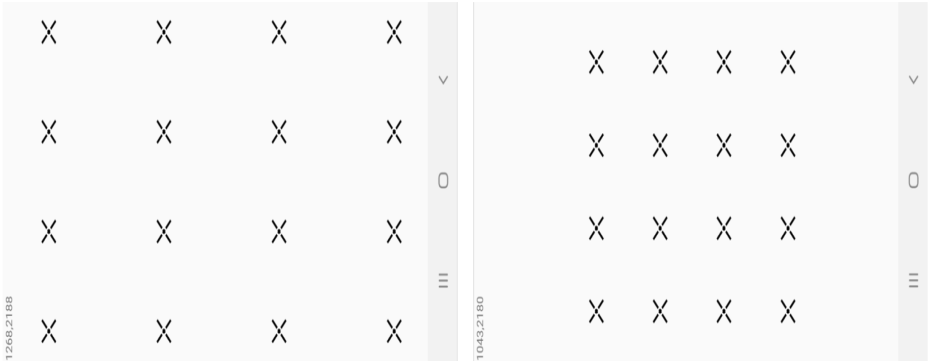

Figure 3.10: Crosshair setup for touch input accuracy test: Coarse (left), dense (right)

ensure the result was not affected by any delay or flickering from the tracking algorithm. Nor the stereoscopic method utilized any tracking algorithms in any form. As a result, a permanently opened “window” of outside context was shown on top of the virtual elements through the HMD. By utilizing a different camera to monitor outside, now we had two separate devices that have same variable factors, which is the inter-pupil distance (IPD) on both stereo camera and the HMD. The adjustments were calibrated prior to the experiment as we found out that the mismatch of IPD variables in two devices may result a distortion of the imported image.

For the task design, we created a UI interface on the Android phone with multiple touch points in both coarse and dense conditions regarding the margins among the touch points (Figure 3.10). As shown in Table 3.1. the total size of the canvas on the LCD was 2960 by 1440. Within the whole canvas, the UI on Android device was setup to have 16 touch contact points marked with visual crosshairs in a 4 × 4 grid spaced evenly across 2250 ×

Table 3.1: Canvas size and margins in pixels for each condition

	Margin Between Points	**Total Space in Pixels**
Canvas	-	2960 × 1440
Coarse Condition	X: 750, Y: 450	2550 × 1200
Dense Condition	X: 416, Y: 266	1250 × 800

1200 pixels in coarse interface and 1250 × 800 pixel in dense interface. The margins among target touch points were 750 pixels for horizontal 400 pixels vertical in the coarse interface, 416 pixels horizontal and 266 pixels vertically on the dense interface. Participants were required to place their fingertip on the surface of the Android device corresponding to the displayed points. The most upper left point was considered as the first point, and the numbers increased in transversal order, as the most lower right point was numbered 16. All participants were requested to touch all 16 points in the UI in a numerical order as accurately as possible. The system recorded all detected actual points of touch contacts with their pixel coordinates including x and y axis, allowing us to see study the accuracy and variations of the result.

We recruited 15 participants in this study and performed repeated measures for three conditions. The participants were 11 male 4 female, aged between 25 to 35 years. The experiment was conducted in repeated measures, letting all participants experience both conditions including monoscopic and stereoscopic real image inside the virtual environment. Prior to the experiment, participants were informed with the objectives of the task, and given at least

30 seconds to get used to the environment with the HMD before the initial touch began. Any auditory or haptic feedback upon a keypress from the mobile phone or the HMD device has been disabled to gather raw finger landing locations without any assistance.

3.6.2 Results

As we recorded the x and y pixel coordinates of all points at the moment that participant pressed the touch screen, a scatterplot is shown in Figure 3.11, visualizing all 16 points of two different tasks with the data from 15 participants. 240 points per conditions were gathered from the user test, consisting a total of 960 points.

For the coarse task with larger margins between the touch points, monoscopic touch condition showed an average pixel error (RMSE) of 63.78 pixels on horizontal axis, 33.56 pixels on vertical axis and 50.96 (SD = 4.77) pixels for combined axes. On the other hand, stereoscopic condition resulted in 55.87 on x axis, 34.53 on y axis and 46.44 (SD = 5.24) for combined. Statistically significant difference was not found between the conditions.

In the results of dense task, monoscopic condition showed an average error of 113.06 on horizontal, 51.56 on vertical axis resulting 87.87 (SD = 13.16) pixels for combined axes. Stereoscopic condition resulted in 62.93 pixels on the x axis, 28.47 pixels for the y axis and 48.84 (SD = 15.01) pixels for considering both axes. The difference between the conditions showed a statistically significant difference at $p = 0.001$. While the stereoscopic

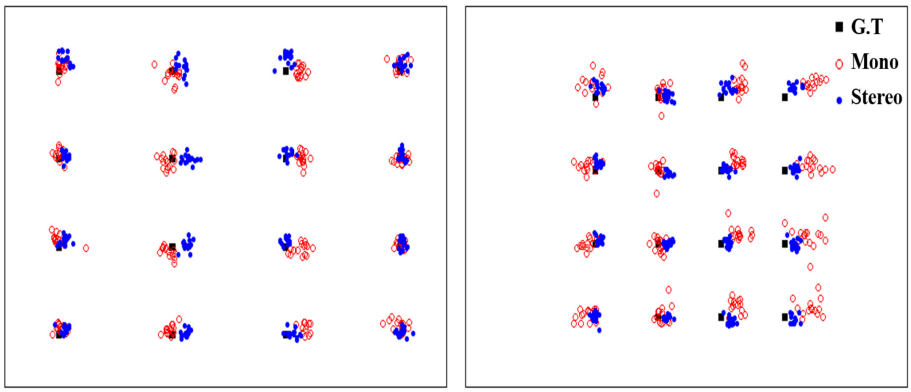

Figure 3.11: Touch points input results in both coarse (left) and dense (right) condition

condition maintained consistent accuracy in finger landing points without any statistically significant difference across the coarse and the dense tasks, the monoscopic condition showed an increasing amount of RMS error from coarse to dense interface. Similar behavior was observed in the text entry task from previous section with a high rate of CER. The result proves that our suspicion that the culprit might be from the lack of stereoscopy. The correlation of RMS error values with the different axes including x and y was considered not significant as the actual horizontal length of the canvas was more than twice long compared to the vertical length.

As shown in Figure 3.11, the distributions of touch points on the monoscopic condition scatters more on dense task than on coarse task, especially on the far left and far right column. Even though the x and y gap between the crosshairs in dense conditions was far wider than those in soft keyboards, we could observe uneven distributions on those areas. We suspect

that this came from the combination of system and psychological factors, meaning that participants' intentions for not to touch the wrong point was added on top of the limitation of the monoscopic visual representation itself. Also, the actual length of the 50 pixels was measured at approximately 4 mm on the device we utilized in this experiment.

Overall, the points from the stereoscopic condition relatively maintained consistency in distribution throughout the tasks, whereas monoscopic system could not.

3.7 Discussion

Throughout the experiments, we could see that our method is capable of delivering a virtual reality experience with a capability of encountering and interacting with a physical object placed outside world. Also, our method can let users use mobile phone from the outside world as an input interface with a decent level of usability. We found that our proposed idea can provide more intuitive and more efficient input experience in virtual environment compared to the conditions with existing hand-held VR dedicated controllers. The performance showed better results both on simple graphic user interface and on more sophisticated UIs for typing. Users also reported a lower level of cybersickness symptoms that could affect the negative aspects of the virtual reality experience. The advantage of the system became more observable in the scenario which required a higher level of precision in controlling objects. However, the issues found from the study revealed that the system began to

lack in terms of stability and performance when the task became more complex. From the results of the first experiment, we observed that the participants had confusion and higher error rate when micro-managing the interface with low margin among boundaries in the UI for the input. We examined the culprit for this issue to discover the underlying problem factor and came to suspect two major factors: stereoscopy and latency. Both factors are a huge factor to define the level of virtual reality experience[62] and combination of slight lack of those together acted as culprit.

The limitation of lack of stereoscopy was further investigated with additional implementation and modification, followed by the preliminary accuracy test. To provide stereoscopy for the real imagery we utilized a binocular camera instead of a regular single lens camera for aligning the image representation. With the additional depth information in the final image, we could see improvements in the results of the test. We found out that the modifying the real imagery with stereoscopy to match the type of the virtual imagery represented in the HMD, dramatically improves the accuracy in participants' fingertip landing points. With the matched IPD on both binocular devices, the implementation leads to less confusion by letting users perform accurate touch inputs.

In terms of the latency problem, due to limitations of the framework we used, we could not maximize the resolution with the fastest framerate of the imagery provided to participants. During the implementation process we found out that even though ORB is a fast and efficient algorithm for pattern-matching, it did not shine when combined with previously mentioned data

pipeline limitation of Unity 3D and the unexpected low performance of the library we used. The OpenCV wrapper we used for Unity and C# did not support multi-core CPU processing nor GPU computation. As we deployed a powerful PC system with a 16-core CPU and a high-end Nvidia GPU, the computational calculation only relied on utilizing 2 or 3 threads of the CPU. This ineffective computation inevitably led to some latency in final representation of the image, and the pipeline problem forced us to stick with low resolution of 720p instead of utilizing a full potential. We believe that this issue is solvable, by utilizing different workaround approaches in the usage of the data pipeline of the engine. As the real visual data stream handler can be established separately from the 3D engine, we plan to deliver the real image in a more efficient way to the engine by processing the high-load calculations outside the 3D engine then directly delivering the real image into the virtual environment in the future work of this study to meet the utilizable latency level[1] in VR.

Overall, our initial idea of using real-world objects as an input device showed a potential to be applied on areas where virtual reality needs to consider and recognize real-world context, such as virtual based simulation, education and medical applications. HMD-based virtual reality technology is already widely applied to such scenarios, to take the advantage of synthetic background of virtual environment as building real environments for such applications can be expensive and often dangerous[14]. However, as direct physical object manipulation is not supported on recent HMD platforms, indirect manipulation methods such as using proxy menu interfaces and

involving proxy objects are used in current applications. To provide a precise and intuitive control for learning purposes, those proxy controllers work as barriers for natural interaction with objects, affecting negatively on presence and learnability. Even though our implementations for interactable physical object in virtual reality only focused on a mobile phone in this study, same algorithms for tracking can be applied to other objects in terms of types and forms.

3.8 Summary

In this chapter, we presented a method to capture a user-friendly object from the real world and import it to a VR scene and use it as an input device in the virtual environment. Our method augments a real element, which is a mobile phone in this study, onto a virtual scene and let users to have ability to manipulate it as an interaction / input device for virtual reality. The advantage of the proposed system is that it allows users to maintain a good level of usability in terms of controlling inputs in virtual environment by using a more familiar and more intuitive protocols compared to existing methods with complexity. We developed and implemented the method which fully supports the purpose of our idea, then conducted different usability experiments inside the virtual environment including menu control, navigation controller and text entry. After that, we came with a new experiment in aligning the vision in the real object representation in VR in order to further investigate the limitation we found during the previous experiment. Data from the preliminary test

showed that we have identified the culprit for the high error rate in previously conducted experiment and we leave a potential mark for future improvements for the platform. Overall, results from the experiments show that our system has a significant advantage compared to the systems with complex hand-held controllers in multiple scenarios, display a little to no difference in user awareness compared to bare eye interaction scenarios and show a feasibility to be utilized as an augmented virtuality method.

Chapter 4. Precise Object Representation for AV

4.1 Introduction

In this chapter, we present a method to import an active touch screen object into virtual environments with proper visual adjustments and investigate the calibrated image representation process in VR, in order to minimize the gap between the interaction in real world and those in the virtual environment. As Augmented Virtuality (AV) is the term for importing objects from the real-world to the virtual environment and requires complex methods to implement, it is not yet a widely explored area in the research field of virtual reality and virtual environment. Thus, it is hard to follow the guidelines for a proper establishment of the platform with accurate visual display and a good level of usability for users. Visualization, image representation, immersion and presence factors could hugely vary even with a small misleading adjustment within the implementation. When designing an AV platform with interactivity, it becomes more difficult to deliver immersive virtual reality experience to users without breaking immersion or creating confusion. Currently developed methods have limitations as they only focused on delivering outside object as an additional visual information to HMD users, not considering the actual contact with physical objects. In this chapter, we implement the platform to deliver additional information display from the real-world into the virtual environment with properly adjusted factors of the representation, and conduct experiments to test the feasibility of the system.

In order to properly import a real-world object into three-dimensional virtual environment, a stereo camera was attached on the top of the HMD to capture simultaneous real imagery with depth information. The image representation method was designed to obtain close-by object then represent it directly into the VR engine to reduce latency and to increase the quality of the final image representation with segmented background. This enables efficient and fast real image representation in VR, as well as supporting other near-by object awareness to users while being immersed in VR.

We evaluated the platform via two different sections of user studies. In the first section, we measured the method in terms of performance as a VR input device, comparing with method that represented monoscopic imagery of the real objects while participants were wearing the HMD for both conditions. Discovering the differences in touch-input awareness of the users between the proposed method and the bare-eye condition was set as the second section. Touchscreen-specific tasks including pinch / zoom and scroll select gesture were given to the participants to interact while they were with / without the HMD. Results shows that our method provides better input performance, as well as showing no significant difference compared to bare eye conditions. In addition, our platform shows great potential for virtual simulation or training application establishment as the system removes necessity for proxy input interfaces for such scenarios.

In the following sections, we present the process we took through in order to provide natural touch interface interaction with real object inside the virtual environment.

4.2 Methods

In this section, we describe our implementation for a natural touch interface in virtual reality using an active device with precise visual representation. Previous attempt of establishing an AV platform with real-object interaction support had some issues. Image alignment was one of the issues, meaning that the types of images shown through the HMD between real object and virtual element differed to each other. As the precondition of visual display of HMD is basically a binocular vision with two separate LCD screens, we found out that the real image should also be following the type, in order to accomplish proper interactive platform with accurate visual representation. The second issue was resulted from the image detection and representation technique. The pattern matching technique utilizing ORB in combination with OpenCV caused delay and also was not able to deliver the users' whole hand to the virtual environment. Caused by abovementioned limitations, previously introduced platform often showed poor results and confusion especially on scenarios which required a more precise control of the object. In the following subsections, we describe a different approach we took in this chapter for establishing the overall system and specific implementations step by step.

4.2.1 System Architecture

The basic structure of the platform has four components including the HMD for final visual representation, binocular camera for outside detection, Unity

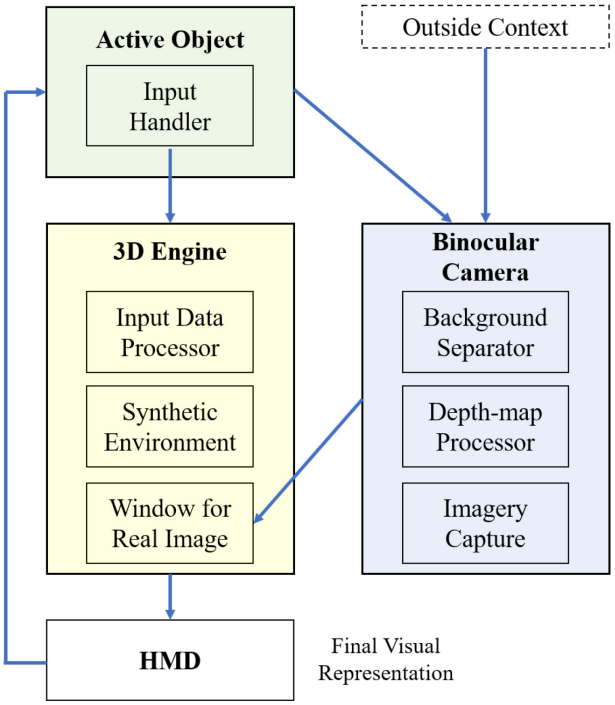

Figure 4.1: Components and structure of the system

3D engine for processing inputs from the object and the target touch interface mobile phone for interaction.

The basic structure and the data flow of the system is shown in Figure 4.1. The camera is installed on the top of the HMD facing down to monitor outside object movements. The FoV of the both camera and HMD were aligned to have the same center point for supporting head movements of the user. As the target appears in FoV of the binocular camera and also reaches a close distance of our configuration from the lenses, the object imagery is imported directly into the virtual environment regardless of types or patterns. This means that the body parts, hands in this case for manipulating the mobile

phone, also appears with the object with segmented background of real imagery. A Samsung Galaxy S9 Android phone was used as the touch input interface for VR. Real-time inputs gathered from the device was transmitted to Unity 3D engine for representing immediate responses onto the virtual environment using TCP/IP protocol.

The synthetic visual information was rendered using Unity 3D. The stream of real imagery from the binocular camera was processed outside the 3D engine with on separate threads of the CPU, as we have experienced the limitation of the 3D engine in terms of handling video streams using internal APIs in previous chapter of the study.

4.2.2 Vision Alignment

As the HMD has two separate displays for left and right eye and the visual image represented is rendered twice with offset, it was obvious that same procedure was required for real image representation. As mentioned earlier, the previous attempt had limitations as the system had to overly a flat 2D image onto 3D display, and this caused confusion and higher error rate in task completions especially on interfaces with higher density among input boundaries (e.g., soft keyboard). To overcome this issue, we replaced the previous regular webcam to a ZED stereo camera which had two separate camera lenses. The ZED device is composed of stereo 2K cameras with dual 4MP RGB sensors. It has a maximum field of view of 110° at low resolutions and can stream uncompressed video at a rate up to 100 FPS in WVGA format.

The camera was calibrated with configurable variables, which in this case were the pixel number and the focal length of the camera in pixels depending on the resolution settings. The field of view of the camera device was calculated as follows (Equation 4.1):

$$fov = 2 \times arctan\ (pixelNumber/(2 \times focalLength) \times (\frac{180}{pi}) \quad (4.1)$$

In summary, the resolution for the imagery was set to 1080p (1920 × 1080) with vertical FoV of 42° and horizontal FoV of 69° for the binocular camera.

Also, to maintain a higher level of accuracy in representing display and minimize potential image distortions, the IPD of both the camera and the HMD were calibrated together for more precise representation at 63 mm and 64 mm, respectively. While carrying out the initial test of our new method with stereoscopic imagery rendered on the virtual scene, we found out that the fingertip landing point confusion is drastically decreased. The position of the camera was set to capture image from the underside of the HMD, assuming that the target object would be placed on the area where users' hand would be placed. The vertical FoV location of the camera often fluctuated due to different styles of HMD wearing positions upon different users, results from the initial test showed that there were not any serious issues related to camera FoV angle misplacement. The image is then transferred to the 3D engine for representation.

4.2.3 Image Representation

The method of using the pattern matching technique for representing the target object after detection created latency in the process as shown in the previous chapter. The amount of delay caused by inefficient calculation and complexity was approximately around 30 ms to 50 ms, which was quite noticeable by users. According to Abrash's research[1], a latency level of less than 20 ms at motion-to-photon (MTP) is considered as an acceptable level for proper virtual reality experience with a enough level of presence. The term motion-to-photon latency is the time need for a user movement to be fully reflected on the display screen. The latency is more distinguishable especially on the virtual reality platforms which utilize close displays for eye (HMDs), as the whole FoV is covered by the screens. Factors such as screen pixel switching time, GPU, CPU and game engine is all closely involved in the data pipeline of the HMD-based VR platform in sequential or parallel manners. Since different components in the MTP pipeline all add up individual latency values to the total amount of delay, minimizing the latency on the additional real-vision process was crucial fact to improve the overall virtual reality experience. Especially on the AV term, the problem might become more serious compared to issues in other regular VR scenarios, as additional data steam handler of outside context inevitably creates latency.

The latency in representing real image caused by the pattern matching technique also often led to poor results and confusion in the user test conducted previously. In this chapter, instead of the pattern matching

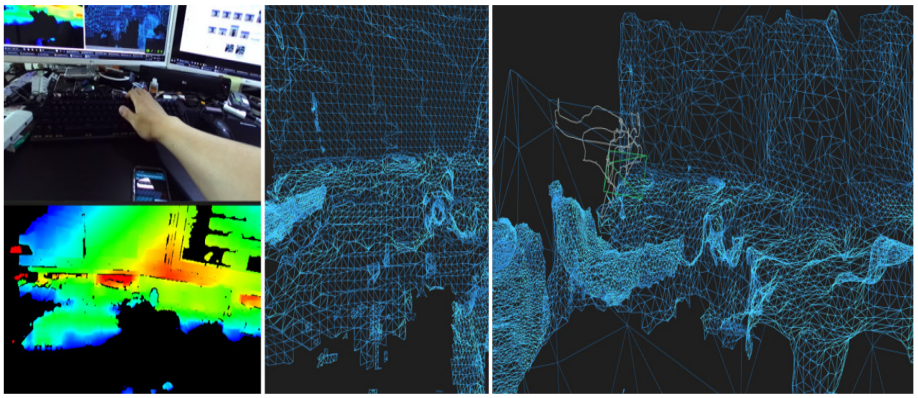

Figure 4.2: Utilization of real-time depth map

techniques such as ORB or AKAZE, we utilized an active real-time depth map to detect close-by objects to user then overlay segmented image onto the virtual scene. To archive best optimization for the usage, the camera parameters were tweaked in various ways. The resolution was set to 1080p at 30 FPS for both left and right cameras and maximum depth map range was limited to 45 cm to improve the performance. As the external cam was only set for capturing near-field objects, maximum distance for detection was calibrated to monitor objects within the reach of the users' arm. An example of background calculation of active depth map is shown in Figure 4.2. Notice that any object other than mobile phone can also appear in the scene without any tracking or detection algorithms.

The latency on processing with the depth map at 1080p resolution was measured approximately at 7 ms with Nvidia 1060 6GB (1280 CUDA cores) GPU. Combining with two factors including the resolution and the depth map

Figure 4.3: An example of image representation with another offline object

distance, an example of the final overlay representation with stereoscopy and modified segmentation blending is shown in Figure 4.3. Additional ability to capture the users' whole hands was also established as the system now represents real object depending on the distance between the object and the camera lenses. When represented in the VR scene, users were able to see the actual phone as well as their hands holding it and interact with the device.

All imagery rendering including the overlay image and the virtual background shown through the HMD were managed on Unity 3D engine. The communication between the input signal from the Android device and feedback from the Unity engine was implemented using TCP/IP protocol. Communication scripts were developed both on Android and Unity in order to transmit input data from the mobile device and show received data in real-time. With all component integrated together, users can manipulate the mobile phone from the real-world with its original texture and use it as an input controller while still being immersed in VR wearing the HMD. Additionally,

Table 4.1: Experiment task groups and measurements

Task Group	Task Name	Measurements
1) Comparison with monoscopic platform	Text Entry	WPM, CER
	Presence Test	29 items in Presence Questionnaire (PQ)
2) Comparison with bare-eye interaction	Gesture Inputs	Input time lag, Error rate

added stereoscopy provided accuracy when interacting with outside objects, by aligning the perspectives of the represented images. Expected finger landing points and the actual landing points would less likely to mismatch with the new platform.

4.3 Experiment Design

The purpose of the evaluation was divided into two groups (Table 4.1). First purpose was to figure out how newly designed method compares to the system which cannot provide stereoscopic image representation, in terms of the input performance and perception levels in control. The first section was consisted of one VR task, and a questionnaire to gather user feedback. The second objective was to compare the system with bare-eye natural conditions, in order to measure the intuitiveness of the system in natural touch input

scenarios. We designed different tasks for users to perform in order to evaluate each purpose of the experiment.

4.3.1 VR Text Input

To evaluate the input performance of the platform compared to the method utilizing single lens camera with flat 2D real image representation, we first set up our experiment environment as the virtual reality text input platform. The VR text input task was a two-hand operation, as participants were holding the mobile phone in their hands, they were asked to type five different stimulus sentences one at a time. Five sentences were randomly drawn at each trial out of 30 sentences phrase set. The stimuli sentences were shown in the VR environment and the visual feedback for the real-time typing content was shown both in the VR and mobile device.

For the software keyboard, we chose Google keyboard for Android with a specific keyboard UI skin that has no boundary margin among the keys as shown in Figure 4.4. This was set to measure the performance and accuracy of touch points of the system on extreme circumstances, which requires a high level of precision in control for a proper text input. Additional features of software keyboard such as auto-correction and swipe-to-input were disabled to minimize any interruptions during the experiment. A custom Android application with above-mentioned software keyboard layout was configured to transmit entered inputs to the 3D engine using TCP/IP protocol. The measurements were word per minute (WPM). Five consecutive characters

Figure 4.4: Specific keyboard UI used in this study

were counted as a word, including spaces. Error rate was measured using character error rate (CER), Stimulus sentences were drawn from the mobile phrase set[117].

4.3.2 Comparison with Bare Eye

The second purpose of this study was to investigate how fluently participants could use their touchscreen mobile phone with our method. If the current configuration provides enough accuracy in terms of fingertip landing point and minimizes confusion that is caused by misalignment in vision representation, we set our hypothesis as the touch input performance with our method would not differ from the performance with the bare-eye natural interaction, without using the HMD. We decided to conduct another portion of the experiment to compare the proposed method with the natural interaction, HMD-less scenario.

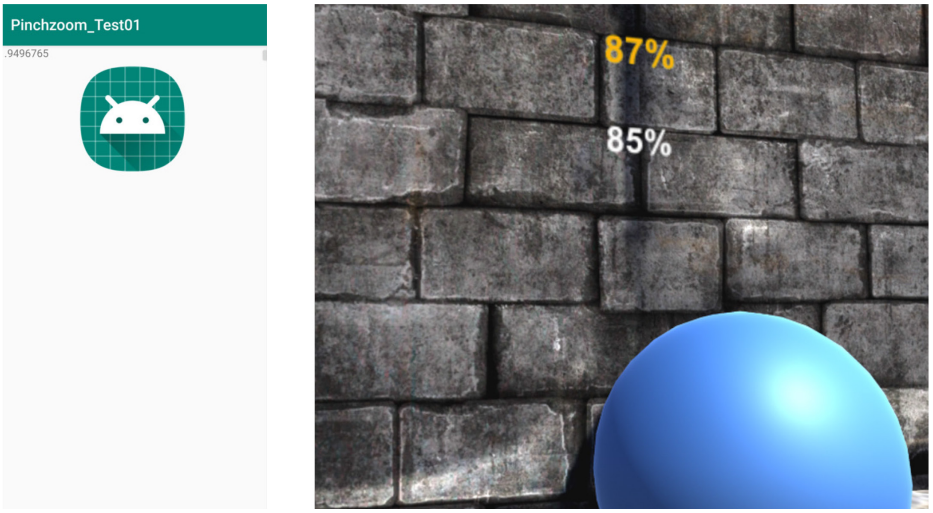

Figure 4.5: UI interface (left) and target object in VR (right) of the pinch-spread task

We wanted to see how our proposed method can be precise and intuitive in terms of controlling the touch interface. The core of gestures for touch commands are divided by the numbers of fingers in usage[118]. According to the Touch Gesture Reference Guide developed by Villamor, there are basic, object-related, navigating and drawing actions in the categories of the touch gesture. In this task design, we chose most used one gesture from the object-related actions, and another from the navigating actions. Two selected actions were the pinch-spread action with two fingers and scroll-select action with a single finger. As tapping with a single finger is also considered as the one of the most frequently utilized action in touch interfaces, we excluded that specific action because the tapping action was already included in the second task, as users must perform a tapping action after the scroll. Also, we wanted

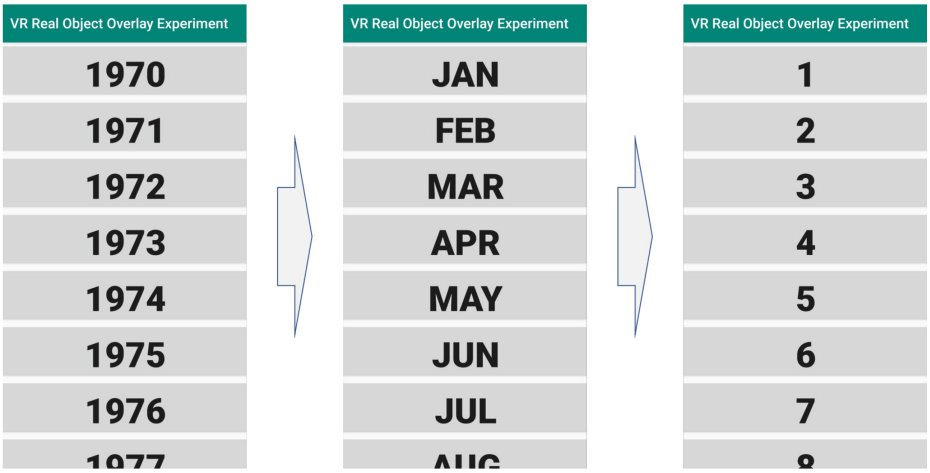

Figure 4.6: The sequence of UI interface in scroll-select task

to minimize the overlap with the previous task and experiment which was conducted in previous chapter of the study. Thus, aside from tapping gesture with one finger, two touchscreen specific tasks were set to measure the fluency on operating touchscreen gestures. First involved the pinch-spread interface for zooming in or out and the second utilized scroll actions to select certain section of the interface. The control UI design for the Android device is shown in Figure 4.5. The same UI interface was used for both conditions with the HMD and with the “natural” bare-eye condition.

Both gesture tasks were set to have random quiz-answer form. The pinch-spread task was set to modify the scale of a sphere in the VR scene. Random target scale values of the scale were given as quizzes and the participants tried to match the absolute value by either zooming in or out by pinching and spreading two fingers. The range of target scale value was set to 10 to 200, and on the interface part, the initial number was set to 100 for both enabling

zooming in and out scenarios staring from the center point. For the UI part on the Android mobile, there wasn't any UI visible element on the screen as the purpose of this interface was only to gather two-finger gesture actions to control elements in the virtual environment. The measurements for this task was completion time and error. The error was calculated by comparing the difference between the given target value and the entered value.

For the scroll-select task, random value of year / date / day values in ISO 8601 format (e.g., 1999-03-11) were given as instructions for participants to provide the corresponding answer. Starting from the year select screen, month and day select screen appeared consecutively upon users' final finger tap input as shown in Figure 4.6. Participants controlled the touch interface to scroll up and down using their fingertips to select then transmit the corresponding value to the 3D engine.

Five trials were given to each participant for each gesture task, entering a total of ten entries for the answers for this section of the experiment. To reduce the risk of familiarity bias, the UI elements on the Android controller were designed not to accept input modifications since the first touch. As the initial touch was started, the answer input session automatically started, and the release motion of finger immediately triggered the closure of the current session and called for the next question. The measurements were operating time and the accuracy. Any values different than the given values were considered as errors.

4.3.3 Presence in Virtual Reality

The presence in virtual environment refers to the illusion that user is actually believing that they exist in one place or environment even in scenarios which the physical location of a user is in somewhere else. The term derived from the original “telepresence”, is a phenomenon enabling interaction with the environment and people outside their physical world via technology. It is a strong subjective experience that is very critical to measure the real-like factor in virtual environments, which are filled with synthetic visual and aural information to trick the users’ perception.

Unlike watching pre-rendered 3D based movie or multimedia contents, the synthetic visual representation in HMD-based virtual environment is rendered according to the user’s head position and gaze direction in real time. The presence is considered as an important factor for establishing an immersive virtual environment as the information has to be updated simultaneously upon users’ unexpected movement, without patterns.

Where there is a disagree on the theories for explaining the cause of the presence, which is a psychological and a heavily subjective feeling[107], a widely accepted theory is that the presence occurs in virtual reality experience when both involvement and immersion arise in the perception of the user.

Involvement is a psychological state of attention on a set of stimuli or related activities and event. It can be also described as the level of users’ willingness of participation towards the virtual environment. Depending on individual perception on the level of significance, different levels of

involvement occurs in the experience. The perception level in involvement requires both physical and perceptional state of conditions, including the visual content quality and the usability of the platform. Any discomfort or cybersickness symptoms occur can diminish the level of involvement. Immersion is a subjective feeling that perceiving oneself to be included in with an environment providing a continuous stream of stimuli and experiences. Virtual reality that could provide high immersion would result in a high level of presence. Presence is the result coming from the combination of involvement and immersion towards the virtual content.

Based on their empirical and theoretical research focusing on the theory of involvement and immersion, Witmer and Singer introduced Immersive Tendencies Questionnaire (ITQ) and Presence Questionnaire (PQ)[127][128] for measuring the immersion / presence level of certain virtual environment contents. ITQ questionnaire measures individual tendencies for immersion, usually prior to PQ questionnaire to predict the PQ values of the user. The early theoretical research sought to define presence and related terms, and identify factors that might add to or detract from the presence experience. As the researched continued throughout the years from 1994, revised version of the PQ questionnaire is published. The version 2.0 of the PQ questionnaire suggested 32 items in seven clusters, whereas version 3.0 suggests 29 items on six or four clusters. The latter study involves a more sophisticated analysis for every factor to correlate the total score of PQ. The cluster components for measurement clusters on each questionnaire are shown in Table 4.2. ITQ questionnaires are useful for predicting the quality of the content including

Table 4.2: Measurement clusters for different and PQ models

Presence Questionnaire (PQ) 6-factor model	**Presence Questionnaire (PQ) 4-factor model**
Involvement	Involvement
Audio Fidelity	Sensory Fidelity
Haptic / Visual Fidelity	Adaptation / Immersion
Adaptation / Immersion	Interface Quality
Consistent with Expectations	
Interface Quality	
-	

involvement, immersion and presence before the actual exposure to users [127]. PQ is can be utilized after the exposure to the virtual environment and measure the quality, usability and the completion degree of the virtual content in seven specific clusters.

In this experiment, as our purpose was set to measure the presence level of the virtual environment with our added implementation, the post-exposure test was required.

As our experiment targeted to investigate the user sensory perception

factor and from the fact that our proposed method could not provide any responses from audio fidelity, we decided to utilize the PQ test with 4-factor model excluding the sensory fidelity cluster, in a seven level Likert scale form. Therefore, three clusters including involvement, adaptation / immersion and interface quality was measured for final PQ score representation. Also, as our platform enabled users to touch objects, item 13 in PQ “How well could you actively survey or search the virtual environment using touch?” was activated. And also, manipulation was enabled in the platform, the associated item number 17 “How well could you move or manipulate objects in the virtual environment?” was used. Two specific items in PQ are generally not used for correlating the total score of PQ as most systems do not permit two actions.

4.4 Experiments

Experiments were conducted with a sequence of two different sections this time. In the first section we measured the proposed method in terms of performance as a VR input device, comparing with the previous approach using single lens camera for image import without stereoscopy. Since the worst error rate occurred on the finest resolution task from the previous work, the text input task was conducted again with a denser keyboard layout in first section of the experiment to investigate the difference. Afterwards, discovering the touch-input awareness of difference compared to bare-eye interaction was set as the second section. Touchscreen specific tasks including pinch-spread and scroll-select tasks were given to participants to interact while they were with or without HMD.

4.4.1 Conditions

Different conditions for experiments were prepared in this study. Separate configurations of conditions were established in order to measure different aspects of the implementation and matched with different combinations.

For the VR text input task, the number of conditions were set to two. The two conditions are listed as below:

• **Mono Overlay:** The real image of the mobile phone was delivered to the VR environment using the single lens webcam (Logitech C920). The size of the object shown through the HMD, was set to be perceived as a real-world size.

• **Stereo Overlay:** The mobile phone image and texture was delivered into the VR with stereoscopic imagery and real-time active depth map. Not only the mobile device but the user's whole hands were represented inside. The scale of the image was also set to similar to a real size.

For the touch gesture action interaction task, two conditions were set up in order to measure the user fluency on manipulating touch screen devices with finger gesture inputs depending on the conditions. The conditions are listed as below:

• **Bare-Eye:** This condition was set as the baseline of the experiment in this task. In this condition, users were asked to control the mobile device without wearing the HMD, meaning that vision display was set to a 2D flat screen monitor instead of stereoscopic HMD. The monitor we used in this condition was a 32-inch LCD screen with LED backlight, with resolution of 1920 × 1080. The monitor was placed in front of the participants, showing the VR content in a full screen mode.

• **Stereo Overlay:** Stereo Overlay condition was set as the same condition as the Stereo Overlay condition of the previous text input task. The mobile phone was visible inside the VR with its original texture.

The conditions were paired for each cluster of tasks, as the purpose for each task slightly differed. Participants performed two clusters, with four conditions.

4.4.2 Participants and Procedure

We recruited 15 participants 8 male, 7 female aged between 21 to 33 years. Among the participants, none of them claimed to be a heavy HMD user but had VR experience at least once. All participants were familiar with using modern smartphones. No participant had trouble manipulating the QWERTY layout of software keyboard or giving hand gestures for the touch input. Before the actual experiment initialized, all participants were informed with the purpose of the study and the way to operate the platform. After the

briefing participants wore the HMD device and adjusted the posture for wearing the HMD device. Showing a demo representation process was followed for gathering initial presence of the virtual environment. During the initial HMD test, the IPD and the camera exposure was calibrated to each participant for accurate display of visual elements without any disturbing noises. The two IPD variables from the HMD and the binocular camera were calibrated together in order to minimize any distortion of the imagery.

We performed repeated measures for this experiment. The executing order of the task group were randomly drawn for each participant to maintain balance in experiments. The tasks were conducted in two clusters. First cluster included two conditions of text input task (Single Overlay with HMD / Stereo Overlay with HMD) and the latter included touch-gesture input task (Bare-eye without HMD / Stereo Overlay with HMD). For both tasks, participants were asked to enter inputs as accurately and as fast they could. After each participant finished two tasks, PQ questionnaire was given for comparing the presence levels and perception levels for Mono Overlay and Stereo Overlay and conditions.

4.5 Results

4.5.1 Text Interface Results

For statistical analysis, we performed Wilcoxon Signed Rank Sum tests with Bonferroni correction to investigate a statically significant difference between each section for two tasks. Figure 4.7 shows the WPM and CER results from

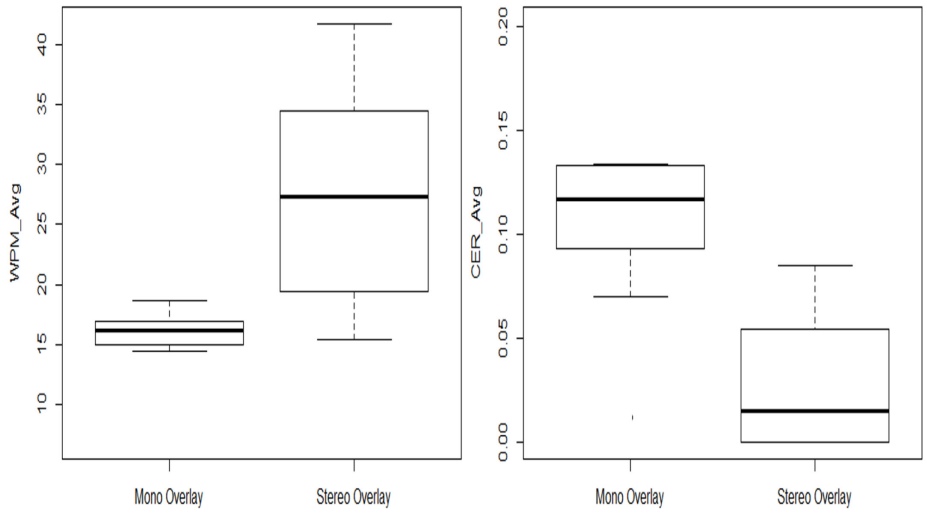

Figure 4.7: WPM and CER scores between Mono and Stereo Overlay conditions ($p < 0.05$ and $p < 0.01$)

the text entry task. The results from the Stereo Overlay method archived significantly ($p < 0.05$) higher WPM values compared to entry results from the Mono Overlay condition. The Mono condition (median = 16.22, SD = 3.58) showed slower input results to the Stereo condition (median = 25.34, SD = 9.37) in terms of WPM.

In terms of CER, there was also a significantly difference ($p < 0.01$) between the conditions as the Mono condition (median = 0.117, SD = 0.054) showed a higher error rate compared to the Stereo condition (median = 0, SD = 0.034). Comparing the CER value of the Stereo condition with previous Baseline condition (Mono Overlay) did not show any statistically significant difference.

The most frequently mentioned feedback from users during the text interface task was that the resolution and latency. Improved image resolution

definitely could deliver a better-quality image inside the HMD and helped the results with a clearer vision of display.

For the latency part, as the proposed method is capable of showing a remarkable level of latency reduction compared to the previously suggested method, this also helped as a factor to result a better input performance and a lower error rate. Even though the resolution helped the performance, we suspect the latency as a primary fact for an improvement as we could not see the effect of size difference in image representation in previous study.

The results imply that our new method with an aligned stereoscopy and less latency in image representation shows a more robust, and a safer way of importing an interactable object from the real-world.

4.5.2 Touch Interface Results

For the results of pinch-spread task we could not find statistically significant difference between the Bare Eye (median = 5.3, SD = 0.57) and Overlay (median = 5.4, SD = 0.28) conditions. The average input response time for the Bare Eye was 5.32 seconds per quiz and 5.48 seconds for Overlay. The percentage error rates for those conditions were 0.4% and 0.38%, respectively. For the scroll select test, the Bare Eye condition showed little better input performance results (median = 4.54, SD = 0.84) than the Overlay condition (median = 4.70, SD = 0.78), but we could not find any significant difference between the results. Since error rate on this task was measured none, we could not compare the error rates from the scroll input.

Although these touchscreen tasks were combinations of simple gesture inputs, users claimed that they did not suffer from confusion or difficulty in terms of providing inputs using hand control. Some participants claimed about the difference in field of view (FoV) between the bare eye and camera-based vision, but the issue did not affect the result to have statistically significant difference between conditions. Additional feedback from the users included that they could not feel any noticeable inconvenience during the text input, they still could feel a little delay in input feedback but was manageable, which did not affect them to make errors.

4.5.3 Presence Test Results

The results from PQ was inspected in aspect of three specific clusters, involvement, adaptation / immersion and interface quality (Figure 4.8). Sensory Fidelity cluster was excluded from the study because the 4-factor PQ only suggests auditory feedback for the specified cluster on three question items.

Comparing the conditions of the mono overlay and the stereo overlay condition, the total score of PQ showed that the stereo condition shows a significantly ($p < 0.01$) higher (median = 50.82, SD = 1.53) score compared to the mono overlay (median = 48.73, SD = 4.81). In the first specific cluster of involvement, no statistically significant difference was found between the mono overlay (median = 26.18, SD = 2.35) and the stereo overlay (median = 28.63, SD = 1.03). Significant improvements were found on both two

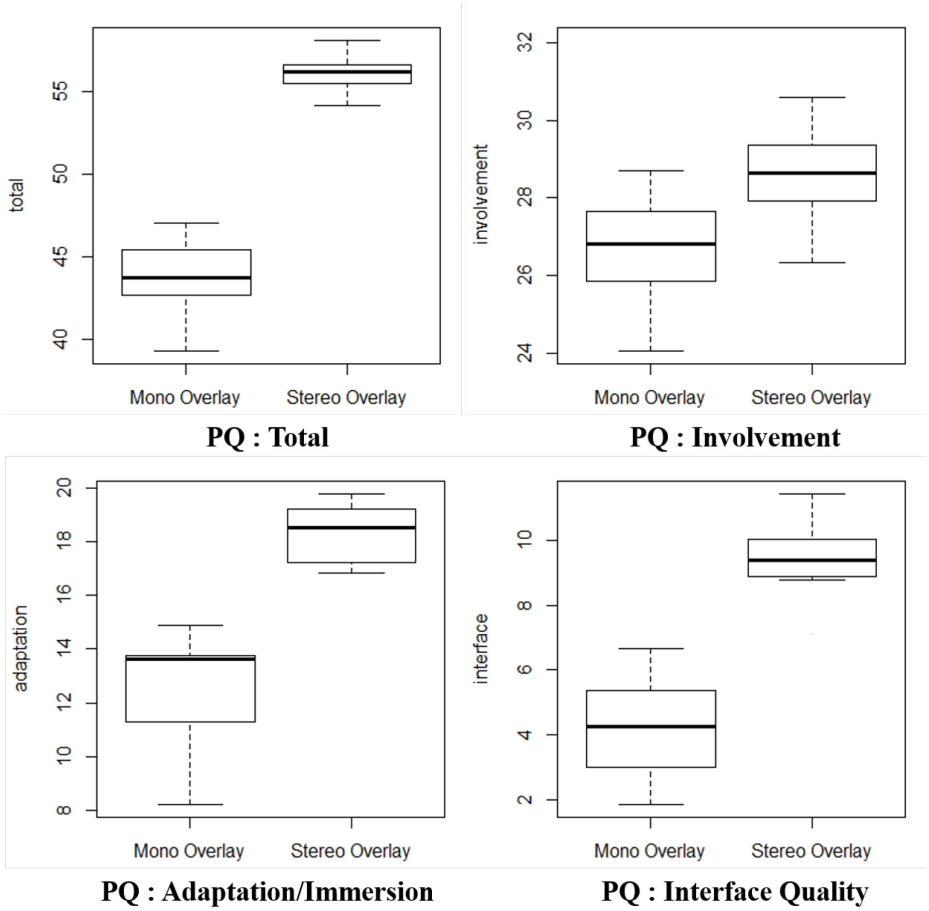

Figure 4.8: Total PQ scores and scores across three clusters

remaining clusters of adaptation / immersion ($p < 0.05$) and interface quality ($p < 0.01$), as mono overlay condition showed median of 13.64 (SD = 1.26) and 3.50 (SD = 1.53) and stereo overlay condition showed median of 18.53 (SD = 1.25) and 9.38 (SD = 4.81) respectively.

Results imply that both augmented virtuality platforms plays a decent role in attracting users' attention so far as the involvement is required, but the

latency and inaccurate display of real object issues in mono overlay condition negatively affect the interface quality perception and the subjective feelings towards the adaptation to the VE and immersion for the experience.

Initial visual representation without accepting users' input from their fingertips involve similar level of attention. But as the input session begins, the different factors that affects the overall virtual environment experience including latency, image blending with background segmentation, and the quality of the image add up to build a virtual reality platform with more usability, adaptiveness for input control and immersion towards the visual content.

4.6 Discussion

Our proposed augmented virtuality method with stereoscopy and depth-map based image representation resulted in several advantages over the system that utilized the pattern matching technique with monoscopic imagery.

First, the different approach we took for image representation resulted in a reduced "flickering" occasion. The flickering issue often happened in the previous method of the study. It came from the process of the system while tracking the detected object then once lose the tracking (pattern not detected) then resume to detect the object (pattern re-detected) again. Even though it was mentioned that such scenarios are prevented by implementing a simple filter to "keep the window open" for three seconds in case of losing tracking, it was little lacking for high level of fluency for representing the real object in

case of lost tracking. Also, as we changed the pattern matching technique to distance-based image segmentation using the real-time depth map, it became possible to represent the whole hand and arm of the user during the session. This enabled a more natural and more blended representation, as the boundaries of the objects in the scene became much smoother. With the fact that close-by objects appear in the HMD without any process requiring complex calculations, additional simple actions such as grabbing a cup or checking hand became possible. This leaves a potential possibility for the platform to be applied for utilizing any objects from the real-world, aside from the active object which was utilized in this study.

Second, the added stereoscopy eliminated the issue which was the mismatch in finger landing points. During the previous experiment we found out that the CER value hugely varied from person to person. There was a general level of confusion applied to all participants, but some showed remarkably better results compared to other who extremely suffered to correctly enter a key. Before conducting the study, we brought back one of the participants who showed the worst results and tested again with new method. Same as the results from the main experiment, the person was able to enter inputs more accurately.

Results from comparing the awareness of difference in touch interface gesture with the bare eye conditions did not show any statistically significant difference, meaning that participants did not noticeable confusion or inconvenience while operating touch screen input. Overall results of our experiment proved that with the new method, the platform is able to provide a

more natural, blended representation as well as a higher level of ability to manipulate the object with better performance and reduced confusion.

User survey using the Presence Questionnaire (PQ) was conducted in order to investigate the newly implemented proposed method in various aspects of user perception. As we have seen in the results, presenting an outside object inside the VR shows a similar level of user involvement towards the VR regardless of the types of visual representation, but the presence level including the virtual reality immersion begins to differ as the interaction with the represented object is involved. To maintain the level of immersion and presence in the virtual environment with outside context, we could find out that we have to consider various factors for proper establishment of the AV platform other than visual representation with original texture. Other factors include latency, background segmentation method, blending method (between the real and virtual). Distance-based segmentation / blending technique might fit better to the AV scenario rather than a traditional target-tracking based methods which are widely used in AR applications, as partial representation of body parts including users' arm and hand or even any other close-by objects is also very important in an AV platform. It provides more natural perception for the users with their surroundings including both virtual reality and reality, with less effort as it is nearly impossible for target-based applications to register and track all random objects around the users' physical location.

As current object representing method is based on distance matching technique, in some scenarios with visual noise (e.g., noise coming from the

variances of locations in lighting) sometimes the system partially fails contour detection of close-by objects. The robustness of our proposed method would improve by utilizing a scanned depth map, which can be obtained by pre-calibration process of deployment location. In this study, we did not use the assistance from the stored data map as we deployed our system in a controlled indoor area with constant lighting and fixed infrastructures. As vision-based methods are prone to create error upon the lighting factor, this could be utilized in the future work of the study for more expandability and a higher accuracy.

Overall, despite some limitations of the current platform, our proposed method has a great potential to be utilized as a framework tool for establishing a virtual reality system with physical object interaction support. Proper alignment of the outside object maintains accuracy while representing the physical object with the original texture, the low latency background segmenting method of real imagery minimizes confusion or presence interruptions. The data transmission package with the 3D engine enables seamless interaction with active objects from the real-world. The represented object in the virtual environment is not only limited to active objects, but also other offline objects are visible when placed within the distance of the detection range. This enables additional actions for users perceiving outside context during the virtual reality experiment, which also leaves an area for future research, expanding the method to importing offline objects into VR as input devices.

4.7 Summary

In this chapter, we explored through the process of implementing an AV platform with physical object interaction support with proper vision alignment and more blended visual representation of the physical object. Stereoscopy in both visions on the HMD device and the outside monitoring camera enabled more accurate object manipulation ability and less confusion for users in scenarios of handling dense UI interface with small margin. The representation technique for segmenting background was implemented by using a real-time active depth map, measuring distance between the lens and the target object. Close-by setup of the depth map enabled more blended visual representation of the physical objects and archived a potential to a feature to import any objects from the real-world, aside from an active mobile phone. Less delay in processing the image in the 3D engine involved a faster framerate and more crisp imagery shown through the HMD.

The effectiveness of the proposed method is verified experimentally as a better method compared to the method from the previous study which lacked in proper visual representation in augmented virtuality. Furthermore, the system maintains input performance for touch screen devices inside the virtual environment compared to the bare eye interaction condition without the HMD. As for the perception aspect of the platform, PQ test was conducted to measure users' subjective feelings of presence in virtual environment. Results show that the platform is able to maintain a higher level of presence and immersion compared to the system that uses other methods for interacting

with physical objects in HMD-based virtual reality. Overall, the proposed method showed good results on both physical representation and the user perception level, resulting in a better AV experience with more expandable possibilities.

Chapter 5. Conclusion and Future Work

5.1 Discussion

The purpose of this dissertation was to propose an efficient and accurate method for establishing a mixed reality platform which supports the encounter and manipulation of physical objects for users while staying inside the virtual reality. We proceeded our research using a physical, active object as a medium to link and integrate two different worlds, taking a novel approach targeting to reduce the gap between the reality and virtual reality. Throughout the results we gathered from our series of experiments, we could find out that our proposed method solves issues that current HMD-based virtual reality experience suffers from.

Current HMD-based virtual reality experience heavily relies on the virtual environment on the mixed reality platform, in a purpose for making the users to be immersed in a simulated experience without disturbances from outside world. Inevitably this created input devices specifically designed only to be utilized in the virtual environment with complexity, and also created the risk of virtual isolation by blocking out all real-world elements. By implementing our method for integrating the virtual environment and the reality by using object as medium, we successfully demonstrated that the concept and the method in an augmented virtuality form can be used as an efficient platform with expandability and reduced virtual isolation, while maintaining the level of immersion to the virtual environment. In terms of

factors of current HMD-based VR platform that we mentioned in Chapter 2, both quantitative and qualitative data from our experiments shows that the proposed method positively affects all four factors including user protection, cybersickness, isolation and indirect manipulation. As the main concept of the proposed method is to have a window for outside target representation, it provides visual cues for recognizing real-world context in a mediate manner and this helps prevent physical collision for users. Especially on the latter implementation mentioned in Chapter 4, the distance-based visual representation enhances physical protection side of the factor it is capable of showing any object near-by instead of showing through a rule-based algorithm shown in Chapter 3. The also reduces the level of the virtual isolation by enabling visual representation of outside contexts of the user. In real usage during experiments, users feedback included that our proposed method gives a sensation of connectivity between the reality and the virtual environment and provided more intuitive control when entering inputs. Additional opinion showed that the method might be useful even on conventional VR scenarios which is isolated virtual reality experience with dedicated controllers, as events that must involve real-world mobile phone (e.g., answering phone, texting messages) can be performed with our method without taking off the HMD device.

Compared to similar AV applications that enables physical object representation and interaction inside the VR based on template matching objects[36][2], distance-based method on displaying outside context of the proposed system enables an easier and more expandability in terms of object

Figure 5.1: Examples of background segmentation error on certain frames

range (e.g., expanding the range to offline objects). Other approaches that enabled interaction with the physical devices[133][4][126] had limitations as the object only could be utilized on specific scenarios and worked as proxies inputs, our method discards the necessity of proxy and enables direct input inside VR with the original texture of the object.

For a pleasant VR experience, the users need to perceive accurate visual movements with a low motion-to-photon (MTP) latency. Several studies suggest 20 to 25 ms[1][39] as the upper bound for MTP for smooth VR experience. As mentioned previously, the delay level of our final proposed method on visual representation is around 7 to 10 ms, which is under the upper bound for MTP latency. While compared to the previous implementation in Chapter 3, The user perception of delayed feeling has while interacting with outside object has significantly decreased in the response of the participants (Chapter 4).

Limitations of final structure of the method includes seldom errors in background segmenting caused by random visual noise at specific time frame

of the outside image. As mentioned in Chapter 4, real-time depth map detection often causes errors due to lack of pre-defined depth map of the location, and with visual noise created from lighting and shadows often failed to precisely segment the physical object resulting in a unsegmented chunks of pixels in the final representation as shown in Figure 5.1. Also, as our implementation targeted to detect and represent a mobile phone into the VR scene, current visual algorithm does not support representing “through” the objects, rather it shows the whole contour of the object regardless of the existence of see through holes or gaps in the target. Walton and Steed[120] represented a method to precisely segment objects in mixed reality scenarios by classifying pixels into four categories: *infront, behind, process* and *ignore.* The categories are associated with conditions that involves overlapped situation with both real and virtual elements, when physical objects have visual gaps that can be seen through (e.g., between the fingers). Through the classification of the pixels, this method enables accurate visual representation when real object with see-through portion is overlaid on to a virtual element. Similar approaches include[26], calibrating depth information into accurate reliable factor in visual representation. However, the latency in processing might be another factor that could negatively affect the final quality of the platform. Thus, adapting the classification method and balancing the latency resulted by the computation, might improve the future structure of the current implementation with a more precise natural real-world visual representation inside the HMD.

In terms of user perception of the virtual environment including

immersion and presence, we conducted experiments gathering subjective measurements through questionnaires. In Chapter 3, SSQ was utilized in order to measure subjective feelings on the disturbance of the virtual environment whereas PQ was used on Chapter 4 to figure out the immersion and presence when using the touch interface within the proposed AV platform. Although results indicated positive aspects in terms of user perception over the compared conventional virtual reality platform, it is known that swapping vision inside VR may increase the cognitive load for human perception[53] and often involves risk of breaking immersion[38]. In our case, switching between the real vision and the virtual elements required swapping vision. Further investigation on this undiscovered area with quantitative inspection methods using bio-signals would show a clearer passage for development and accurate user perception in augmented virtuality.

The method proposed in the dissertation can be applied to real-world scenarios where it is required to involve physical object manipulation during the VR experience. Many applications including different areas such as training[134], simulation[5] and medical[124] applications require virtual environment and inputs from the users. The input is gathered to properly give feedback back to the users, as the purpose of the applications is to deliver knowledge or train muscle memory. Even though real object manipulation is required in real-world scenarios of the contexts, proxy or gesture-based input is utilized in virtually reconstructed scenes as previously discovered areas of virtual reality platforms do not let physical objects to communicate with the virtual environment. With our method, those applications can be modified to

have a more intuitive input system, with direct manipulation not requiring the usage of proxy UIs or devices, letting inputs from outside can be directly reflected inside the virtual environment.

Overall, in this dissertation, we proposed a novel approach to display a real-world element inside the VR, with the purpose of providing an intuitive input control using external devices, and also reducing the physical gap between two different worlds in order to minimize the virtual isolation which current HMD-based virtual reality experience suffer from. Initial implementation focused on importing a mobile phone into the virtual environment as input controller for VR, and refined method targeted to precisely represent the target object inside virtual environment, in order to provide accurate control with minimized risk of breaking immersion and presence of VR. Through the series of user experiments, we demonstrated the potential of our method, establishing a platform to fill the gap in the literature, the undiscovered area in the mixed reality spectrum.

5.2 Contributions

Virtual reality platforms still have limitations in terms of fully integrating with the real world. Many areas including sensory mismatch, isolation, stimulation and input have unresolved issues for a perfect human sensory deception and providing realistic simulated experiences. In this thesis, we focus on the area of blending both worlds by offering interactivity with the outside world while staying inside the virtual world and also to propose an efficient way to use the

physical object as intuitive, natural interaction device to control the virtual world.

We analyzed previously conducted researches in the area of virtual reality regarding different aspects of limitations (Chapter 2) and proposed a novel approach for overlaying real object into the virtual environment to integrate reality with virtual world with gathering observations through a user study (Chapter 3). The approach can be placed on the augmented virtuality technology under the taxonomy of mixed reality but could be considered as a new method as we enable additional interaction with real-world to the existing technology that only supports visual information. Also, focusing on the observation that there were series of factors to consider for the establishment of a proper AV platform with a precision and high level of usability, we presented a method which was not presented by traditional approaches (Chapter 5). We took a process of finding optimal representation method with a more precise, natural interaction support with minimized negative effects on the perception of virtual reality experience, by testing with different perspectives of the platform.

The main contributions of this thesis can be summarized as below:

• **Reducing the Physical Gap Between Reality and VR:** Conventional virtual reality experience blocks out the reality. Therefore, it faces the challenge in scenarios which involves physical objects encounter. The physical gap is crucial factor in virtual isolation[50]. Importing a real-world object into virtual environment with augmented virtuality can prevent visual

isolation of HMD-based virtual reality experience. Using object to link the reality and virtual environment results in a great reduction in physical restriction of the user and increases the awareness of simultaneous context within the real-world with minimized distraction from the immersive VR experience. Also, as shown in the results of performance experiments, a user-friendly device as an input device in the virtual environment improves usability and intuitiveness of the controllers. Previously introduced methods in the research field of augmented virtuality had limitations for natural integration of reality and virtual reality, as they lacked in delivering the object as an interactable device with original texture.

• **Visual Guideline for Augmented Virtuality Platform with Physical Object Manipulation:** As in the process of our research, we found out several factors on visual representation in an AV platform affects very different level of usability. Important factors including image alignment and latency of representation were observed in the study. Although two worlds are blended in this study and required more complex methods to implement compared to conventional platforms that only utilizes virtual environment, our method meets the upper bound of MTP latency level[1][39] for immersive VR experience. The minimized image distortion in AV platform is achievable through precisely matching the three factors on physical visual representation inter-screen distance (ISD), lens inter-ocular distance (IOD) and users' inter-pupil distance (IPD). The representation of detailed technical specifications for real object display inside the VR demonstrates guidelines for establishing

an augmented virtuality platform.

• **Finding Potential Aspects of the Platform to be Utilized in Different Applications of Virtual Reality:** In every application of virtual environment applications under current HMD-based virtual reality platforms including related research using AV term, encountering physical object is not possible. Thus, proxy interface or devices are utilized in every way, which also lacks representing the texture of the foreign object inside the VR. In this study, we managed to implement a method which could be applied to any random objects that exist in real-world without any tracking methods by utilizing distance-based detection in Chapter 4. We expect that this could be utilized in mixed reality scenarios to dispose of proxy apparatus for a higher usability.

As previously explored researches in the area of augmented virtuality was not different from the isolated virtual reality in terms of contacting physical objects outside, interactivity with objects was not considered in scenarios even implementations presented outside visual context inside the virtual environment. The isolated feeling can be partially solved by providing additional visual information, but the physical isolation in the virtual environment persists with restricted interaction. Our approach adds additional ability to the AV platform, to resolve isolation and with improved usability for input. The proposed implementation supporting a natural blending with reality and virtual reality also reduces the switching costs for outside context monitoring, as current HMD-based experiences require to take off their

headset to interact with outside, which completely destroys the immersion and presence of the experience.

Through experiments and discovering insights from our method, the results demonstrate that the proposed methods significantly improve and gives promising effect for better virtual reality input experience with less isolation. Hence, expected to be used as tool for the growing research field of mixed and augmented virtuality.

5.3 Future Work

5.3.1 Expanding Interactable Object Range

Although our presented method was capable of representing any objects from real-world into the virtual environment, we did not involve any interaction with them as the purpose of the study was to investigate an intuitive input device for augmented virtuality as well as finding an efficient and precise method for visual representation of the object.

Enabling the involvements of random physical items into virtual reality and using them as input or interaction device would require two main methods to resolve issues that might exist; object classification method and input signal gathering method. In order to distinguish the types or shapes of the objects would involve classifying different categories of the objects. Automatic object detection in traditional computer vision and image processing domain is a technology that detects instances of semantic objects of certain class[25]. In

order to classify the class among different objects, different approaches including machine learning and deep learning are two general ways taken for the implementation. Recent object detection methods support real-time object recognition and classification[94] and often the implementations are deployed in scenarios which requires automatic vision analysis such as surveillance. However, when combined with the HMD-based virtual reality experience, faster computation is required to meet the MTP level of 25 ms[1][39] for seamless integration with the virtual environment. Also, Pre-defined database of objects across different ranges should be obtained prior to the usage for a more efficient algorithm, and heavy computation would also be required.

In the aspect of input signal gathering, certain rule-based operation would be involved to catch the status change of the objects for input control inside the virtual environment. Variables such as the shape, fixture, and orientation of the object could be utilized as input triggers for offline objects. Often this method may be applied on controlled environments, such as task-specific training / simulation scenarios (e.g., lock picking in a real-world maze with synthetic background) where the real object interaction can benefit in terms of intuitiveness and performance compared to using proxy apparatus. The integration between the reality and the virtual environment could also become more solid as random inputs from the reality can be reflected in the virtual environment.

5.3.2 Improving Aspects of User Perception

Perception is an important factor in virtual environments, as the experience itself is created by combinations of synthetic visual / aural stimulus to deceive human sensory. As sensory receptors in different senses are very delicate portions of the human body, even a slight misleading factor in the virtual reality experience can lead to a drastic effect on the experience. Also, the immersion and presence levels from the user perception hugely vary from person to person as those terms could be defined as heavily subjective feelings. Numerous researches over two decades focused on the measuring and discovering the factors that could affect the VR experience, but it is yet to be clearly stated as it all comes from the subjective feelings, as mentioned above. For instance, up to this day, the clear culprit for the cybersickness is not clearly discovered, and only could provide general guidelines that could fit to majority of people, but there are always exceptions more than few.

Quantitative inspection for human perception is a widely studied area, where different types of bio signals are utilized to investigate the pattern of the signals with certain stimulus or scenarios. We utilized subjective measurement using questionnaires including widely known Simulator Sickness Questionnaire (SSQ) and Presence Questionnaire (PQ) in this dissertation, as we were concerned that involving quantitative measures into the study would drift the focus of this research. Throughout the user studies with abovementioned questionnaires we have obtained relatively decent results regarding the immersion and presence in our VR platform, but

blending two worlds inevitably involves focus swap and may result unknown negative effects on user experience. As there are debates on mixed reality immersion and cognitive load[38], an approach to discover the patterns for specifically targeting augmented virtuality with quantitative measurements would be able to contribute to the research area of mixed reality to establish different contents and platforms with a sophisticated sensory deception, resulting a seamless integration of reality and virtual reality with a high level of decent user perception.

Bibliography

[1] Michael Abrash. 2014. What VR could, should, and almost certainly will be within two years. *Steam Dev Days, Seattle* 4, (2014).

[2] Ghassem Alaee, Amit P Deasi, Lourdes Peña-Castillo, Edward Brown, and Oscar Meruvia-Pastor. 2018. A User Study on Augmented Virtuality Using Depth Sensing Cameras for Near-Range Awareness in Immersive VR. *IEEE VR's 4th Workshop on Everyday Virtual Reality (WEVR 2018)* May (2018), 10.

[3] Pablo F Alcantarilla and T Solutions. 2011. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell* 34, 7 (2011), 1281–1298.

[4] Jatin Arora, Varnit Jain, Aryan Saini, Shwetank Shrey, Nirmita Mehra, and Aman Parnami. 2019. VirtualBricks: Exploring a scalable, modular toolkit for enabling physical manipulation in VR. *Conference on Human Factors in Computing Systems - Proceedings* Chi (2019), 1–12. DOI:https://doi.org/10.1145/3290605.3300286

[5] Turgay Aslandere, Daniel Dreyer, Frieder Pankratz, and René Schubotz. 2014. A Generic Virtual Reality Flight Simulator. In *Virtuelle und Erweiterte Realität, 11. Workshop der GI-Fachgruppe Tagungsband. GI-Workshop "Virtuelle und Erweiterte Realität" (GI VR/AR-2014), September 25-26, Bremen, Germany*.

[6] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. 2015. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *Proceedings of the IEEE International*

Conference on Computer Vision. DOI:https://doi.org/10.1109/ICCV.2015.226

[7] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. 2008. Speeded-up robust features (SURF). *Computer vision and image understanding* 110, 3 (2008), 346–359.

[8] Hrvoje Benko, Edward W. Ishak, and Steven Feiner. 2004. Collaborative mixed reality visualization of an archaeological excavation. *ISMAR 2004: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality* Ismar (2004), 132–140. DOI:https://doi.org/10.1109/ISMAR.2004.23

[9] Hrvoje Benko, Ricardo Jota, and Andrew D. Wilson. 2012. MirageTable: Freehand interaction on a projected augmented reality tabletop. In *Conference on Human Factors in Computing Systems - Proceedings*. DOI:https://doi.org/10.1145/2207676.2207704

[10] Hrvoje Benko, Andrew D. Wilson, and Federico Zannier. 2014. Dyadic projected spatial augmented reality. In *UIST 2014 - Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. DOI:https://doi.org/10.1145/2642918.2647402

[11] Shimmila Bhowmick and Keyur Sorathia. 2018. Report: State of the Art Seminar Explorations on Body-Gesture based Object Selection on HMD based VR Interfaces for Dense and Occluded Dense Virtual Environments. 166105005 (2018). Retrieved from http://embeddedinteractions.com/Files/SOAS_Report_Shimmila_Bhowmick.pdf

[12] Oliver Bimber and Ramesh Raskar. 2006. Modern approaches to augmented reality. In *SIGGRAPH 2006 - ACM SIGGRAPH 2006*

Courses. DOI:https://doi.org/10.1145/1185657.1185796

[13] Frank A Biocca and East Lansing. 1998. Body : Adaptation to Visual Displacement in See-Through , Head-Mounted Displays. *Presence* 7, 3 (1998), 262–277.

[14] Eberhard Blümel and Tina Haase. 2010. Virtual reality platforms for education and training in industry. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. DOI:https://doi.org/10.1007/978-3-642-12082-4_1

[15] John Bolton, M Lambert, D Lirette, and B Unsworth. 2014. PaperDude: a virtual reality cycling exergame. *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (2014), 475–478. Retrieved May 17, 2014 from http://dl.acm.org/citation.cfm?id=2574827

[16] Joshua Borah. 1995. *Investigation of Eye and Head Controlled Cursor Positioning Techniques.*

[17] Doug A. Bowman, Sabine Coquillart, Bernd Froehlich, Michitaka Hirose, Yoshifumi Kitamura, Kiyoshi Kiyokawa, and Wolfgang Stuerzlinger. 2008. 3D user interfaces: New directions and perspectives. *IEEE Computer Graphics and Applications* (2008). DOI:https://doi.org/10.1109/MCG.2008.109

[18] Pulkit Budhiraja, Rajinder Sodhi, Brett Jones, Kevin Karsch, Brian Bailey, and David Forsyth. 2015. Where's my drink? Enabling peripheral real world interactions while using HMDs. *arXiv preprint arXiv:1502.04744* (2015).

[19] Grigore C Burdea and Philippe Coiffet. 2003. *Virtual reality technology*. John Wiley & Sons.

[20] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. 2010. Brief: Binary robust independent elementary features. In *European conference on computer vision*, 778–792.

[21] James Calvin, Alan Dickens, Bob Gaines, Paul Metzger, Dale Miller, and Dan Owen. 1993. The SIMNET virtual world architecture. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, 450–455.

[22] Daniel W. Carruth, Christopher R. Hudson, Cindy L. Bethel, Matus Pleva, Stanislav Ondas, and Jozef Juhar. 2019. Using HMD for Immersive Training of Voice-Based Operation of Small Unmanned Ground Vehicles. DOI:https://doi.org/10.1007/978-3-030-21565-1_3

[23] Polona Caserman, Augusto Garcia-Agundez, Robert Konrad, Stefan Göbel, and Ralf Steinmetz. 2018. Real-time body tracking in virtual reality using a Vive tracker. *Virtual Reality*. DOI:https://doi.org/10.1007/s10055-018-0374-z

[24] T.P. Caudell and D.W. Mizell. 2003. Augmented reality: an application of heads-up display technology to manual manufacturing processes. DOI:https://doi.org/10.1109/hicss.1992.183317

[25] C H Chen. 2016. *Handbook of Pattern Recognition and Computer Vision*. DOI:https://doi.org/10.1142/9503

[26] Li Chen, Hui Lin, and Shutao Li. 2012. Depth image enhancement for Kinect using region growing and bilateral filter. In *Proceedings - International Conference on Pattern Recognition*.

[27] Inrak Choi, Heather Culbertson, Mark R. Miller, Alex Olwal, and Sean Follmer. 2017. Grabity: A wearable haptic interface for simulating weight and grasping in virtual reality. In *UIST 2017 - Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. DOI:https://doi.org/10.1145/3126594.3126599

[28] Inrak Choi, Elliot W. Hawkes, David L. Christensen, Christopher J. Ploch, and Sean Follmer. 2016. Wolverine: A wearable haptic interface for grasping in virtual reality. In *IEEE International Conference on Intelligent Robots and Systems*. DOI:https://doi.org/10.1109/IROS.2016.7759169

[29] Song Woo Choi, Min Woo Seo, Sang Lyn Lee, Jong Hwan Park, Eui Yeol Oh, Jong Sang Baek, and Suk Ju Kang. 2016. Head position model-based latency measurement system for virtual reality head mounted display. In *Digest of Technical Papers - SID International Symposium*. DOI:https://doi.org/10.1002/sdtp.10930

[30] Kyriaki Christaki, Konstantinos C. Apostolakis, Alexandros Doumanoglou, Nikolaos Zioulis, Dimitrios Zarpalas, and Petros Daras. 2019. Space wars: An augmentedvr game. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. DOI:https://doi.org/10.1007/978-3-030-05716-9_47

[31] Joon Hao Chuah and Benjamin Lok. 2012. Experiences in Using a Smartphone as a Virtual Reality Interaction Device. *Workshop on Off-The-Shelf Virtual Reality, ...* (2012).

[32] Shang Wen Chuang, Chun Hsiang Chuang, Yi Hsin Yu, Jung Tai King, and Chin Teng Lin. 2016. EEG Alpha and Gamma Modulators Mediate Motion Sickness-Related Spectral Responses. *International*

Journal of Neural Systems (2016). DOI:https://doi.org/10.1142/S0129065716500076

[33] Patrick Costello. 1997. *Health and Safety Issues associated with Virtual Reality - A Review of Current Literature*. DOI:https://doi.org/http://citeseerx.ist.psu.edu/viewdoc/download?doi = 10.1.1.6.3025&rep=rep1&type=pdf

[34] Carolina Cruz-Neira, Daniel J. Sandin, Thomas A. DeFanti, Robert V. Kenyon, and John C. Hart. 1992. The CAVE: Audio Visual Experience Automatic Virtual Environment. *Communications of the ACM* (1992). DOI:https://doi.org/10.1145/129888.129892

[35] G F P Deecker and J P Penny. 1977. Standard input forms for interactive computer graphics. *ACM SIGGRAPH Computer Graphics* 11, 1 (1977), 32–40.

[36] Amit P. Desai, Lourdes Pena-Castillo, and Oscar Meruvia-Pastor. 2018. A Window to Your Smartphone: Exploring Interaction and Communication in Immersive VR with Augmented Virtuality. *Proceedings - 2017 14th Conference on Computer and Robot Vision, CRV 2017* 2018-Janua, (2018), 217–224. DOI:https://doi.org/10.1109/CRV.2017.16

[37] Kevin Desai, Suraj Raghuraman, Rong Jin, and Balakrishnan Prabhakaran. 2017. QoE Studies on Interactive 3D Tele-Immersion. In *Proceedings - 2017 IEEE International Symposium on Multimedia, ISM 2017*. DOI:https://doi.org/10.1109/ISM.2017.27

[38] Tafadzwa Joseph Dube and Ahmed Sabbir Arif. 2019. Text Entry in Virtual Reality: A Comprehensive Review of the Literature. DOI:https://doi.org/10.1007/978-3-030-22643-5_33

[39] Mohammed S. Elbamby, Cristina Perfecto, Mehdi Bennis, and Klaus Doppler. 2018. Toward Low-Latency and Ultra-Reliable Virtual Reality. *IEEE Network* 32, 2 (2018), 78–84. DOI:https://doi.org/10.1109/MNET.2018.1700268

[40] Carmine Elvezio, Frank Ling, Jen Shuo Liu, and Steven Feiner. 2018. Collaborative virtual reality for low-latency interaction. In *UIST 2018 Adjunct - Adjunct Publication of the 31st Annual ACM Symposium on User Interface Software and Technology*. DOI:https://doi.org/10.1145/3266037.3271643

[41] Mark Fiala. 2005. ARTag, a fiducial marker system using digital techniques. In *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*. DOI:https://doi.org/10.1109/CVPR.2005.74

[42] S. S. Fisher, M. McGreevy, J. Humphries, and W. Robinett. 2004. Virtual environment display system. (2004), 77–87. DOI:https://doi.org/10.1145/319120.319127

[43] Paul M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* (1954). DOI:https://doi.org/10.1037/h0055392

[44] Jesse Fox, Sun Joo Grace Ahn, Joris H. Janssen, Leo Yeykelis, Kathryn Y. Segovia, and Jeremy N. Bailenson. 2015. Avatars versus agents: A meta-analysis quantifying the effect of agency on social influence. *Human-Computer Interaction* (2015). DOI:https://doi.org/10.1080/07370024.2014.921494

[45] Friðriksson, Kristjánsson, Sigurðsson, Thue, and Vilhjálmsson. 2016.

Become your Avatar: Fast Skeletal Reconstruction from Sparse Data for Fully-tracked VR. In *Icat-Egve*. DOI:https://doi.org/10.1109/ICVR

[46] Sebastian Friston and Anthony Steed. 2014. Measuring latency in virtual environments. *IEEE Transactions on Visualization and Computer Graphics* (2014). DOI:https://doi.org/10.1109/TVCG.2014.30

[47] Gameplus. 2017. PaperStick VR Controller. Retrieved from https://sites.google.com/site/gameplusvr/

[48] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and R. Medina-Carnicer. 2016. Generation of fiducial marker dictionaries using Mixed Integer Linear Programming. *Pattern Recognition* (2016). DOI:https://doi.org/10.1016/j.patcog.2015.09.023

[49] Google. 2015. Google Cardboard. Retrieved from https://vr.google.com/cardboard/

[50] Jan Gugenheimer, Mark McGill, Frank Steinicke, Christian Mai, Julie Williamson, and Ken Perlin. 2019. Challenges using head-mounted displays in shared and social spaces. In *Conference on Human Factors in Computing Systems - Proceedings*. DOI:https://doi.org/10.1145/3290607.3299028

[51] Jan Gugenheimer, Evgeny Stemasov, Julian Frommel, and Enrico Rukzio. 2017. ShareVR: Enabling co-located experiences for virtual reality between HMD and Non-HMD users. In *Conference on Human Factors in Computing Systems - Proceedings*. DOI:https://doi.org/10.1145/3025453.3025683

[52] Zhenyi He, Fengyuan Zhu, Aaron Gaudette, and Ken Perlin. 2017.

Robotic Haptic Proxies for Collaborative Virtual Reality. (2017). Retrieved from http://arxiv.org/abs/1701.08879

[53] Edward J. Hillman, Jacob J. Bloomberg, P. Vernon McDonald, and Helen S. Cohen. 1999. Dynamic visual acuity while walking in normals and labyrinthine- deficient patients. *Journal of Vestibular Research: Equilibrium and Orientation* (1999).

[54] Mochamad Hilman, Dwi Kurnia Basuki, and Sritrusta Sukaridhoto. 2019. Virtual hand: VR hand controller using IMU and flex sensor. *International Electronics Symposium on Knowledge Creation and Intelligent Computing, IES-KCIC 2018 - Proceedings* (2019), 310–314. DOI:https://doi.org/10.1109/KCIC.2018.8628594

[55] Ming Hou. 2001. User experience with alignment of real and virtual objects in a stereoscopic augmented reality interface. In *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*, 6.

[56] P A Howarth. 1994. Virtual Reality: an occupational health hazard of the future. In *RCN Occupational Health Nurses Forum. Working for Health*.

[57] P A Howarth and P J Costello. 1996. Visual effects of immersion in virtual environments: Interim results from the UK Health and Safety Executive Study. In *SID INTERNATIONAL SYMPOSIUM DIGEST OF TECHNICAL PAPERS*, 885–888.

[58] Christopher R. Hudson, Cindy L. Bethel, Daniel W. Carruth, Matus Pleva, Stanislav Ondas, and Jozef Juhar. 2018. Implementation of a speech enabled virtual reality training tool for human-robot interaction. In *DISA 2018 - IEEE World Symposium on Digital Intelligence for*

Systems and Machines, Proceedings.
DOI:https://doi.org/10.1109/DISA.2018.8490615

[59] Hikaru Ibayashi, Yuta Sugiura, Daisuke Sakamoto, Natsuki Miyata, Mitsunori Tada, Takashi Okuma, Takeshi Kurata, Masaaki Mochimaru, and Takeo Igarashi. 2015. Dollhouse VR: A multi-view, multi-user collaborative design workspace with VR technology. *SIGGRAPH Asia 2015 Posters, SA 2015* (2015), 2–3.
DOI:https://doi.org/10.1145/2820926.2820948

[60] Brooke J. 1996. A quick and dirty usability scale. *Usability evaluation in industry* (1996). Retrieved from https://cui.unige.ch/isi/icle-wiki/_media/ipm:test-suschapt.pdf

[61] Florian Jentsch, Michael Curtis, Randy Pausch, Thomas Crea, and Matthew Conway. 2019. A Literature Survey for Virtual Environments: Military Flight Simulator Visual Systems and Simulator Sickness. In *Simulation in Aviation Training*.
DOI:https://doi.org/10.4324/9781315243092-10

[62] Jason Jerald. 2015. *The VR Book*.
DOI:https://doi.org/10.1145/2792790

[63] Fan Jiang, Xubo Yang, and Lele Feng. 2016. Real-time full-body motion reconstruction and recognition for off-the-shelf VR devices. In *Proceedings - VRCAI 2016: 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry*.
DOI:https://doi.org/10.1145/3013971.3013987

[64] Matthew Johnson, Irene Humer, Brian Zimmerman, Joshua Shallow, Liudmila Tahai, and Krzysztof Pietroszek. 2016. Low-cost latency compensation in motion tracking for smartphone-based head mounted

display. In *Proceedings of the Workshop on Advanced Visual Interfaces AVI*. DOI:https://doi.org/10.1145/2909132.2926076

[65] Ebrahim Karami, Siva Prasad, and Mohamed Shehata. 2017. Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726* (2017).

[66] Shunichi Kasahara, Keina Konno, Richi Owaki, Tsubasa Nishi, Akiko Takeshita, Takayuki Ito, Shoko Kasuga, and Junichi Ushiba. 2017. Malleable embodiment: Changing sense of embodiment by spatial-temporal deformation of virtual human body. In *Conference on Human Factors in Computing Systems - Proceedings*. DOI:https://doi.org/10.1145/3025453.3025962

[67] H. Kato and M. Billinghurst. 1999. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proceedings - 2nd IEEE and ACM International Workshop on Augmented Reality, IWAR 1999*. DOI:https://doi.org/10.1109/IWAR.1999.803809

[68] Robert S. Kennedy, Norman E. Lane, Kevin S. Berbaum, and Michael G. Lilienthal. 1993. Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness. *The International Journal of Aviation Psychology 3*, 203–220. DOI:https://doi.org/10.1207/s15327108ijap0303_3

[69] Robert Samuel Kennedy and Lawernce H Frank. 1985. *A review of motion sickness with special reference to simulator sickness*.

[70] Behrang Keshavarz and Heiko Hecht. 2011. Validating an efficient method to quantify motion sickness. *Human Factors* (2011).

DOI:https://doi.org/10.1177/0018720811403736

[71] Sebastian Knoedel and Martin Hachet. 2011. Multi-touch RST in 2D and 3D spaces: Studying the impact of directness on user performance. *3DUI 2011 - IEEE Symposium on 3D User Interfaces 2011, Proceedings* (2011), 75–78. DOI:https://doi.org/10.1109/3DUI.2011.5759220

[72] Hideki Koike, Yoshinori Kobayashi, and Yoichi Sato. 2001. Integrating Paper and Digital Information on EnhancedDesk: A Method for Realtime Finger Tracking on an Augmented Desk System. *ACM Transactions on Computer-Human Interaction* (2001). DOI:https://doi.org/10.1145/504704.504706

[73] Eugenia M Kolasinski. 1995. Simulator Sickness in Virtual Environments. *United States Army Research Institute fo the Behavioral and Social Sciences*. DOI:https://doi.org/10.1121/1.404501

[74] Belinda Lange, Skip Rizzo, Chien-yen Chang, Evan A. Suma, and Mark Bolas. 2011. Markerless Full Body Tracking: Depth-Sensing Technology within Virtual Environments. In *Interservice/Industry Training, Simulation, and Education Conference*.

[75] Marc Erich Latoschik, Jean Luc Lugrin, Michael Habel, Daniel Roth, Christian Seufert, and Silke Grafe. 2016. Breaking bad behavior: Immersive training of class room management. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*. DOI:https://doi.org/10.1145/2993369.2996308

[76] J LaViola. 1999. A survey of hand posture and gesture recognition techniques and technology. *Brown University, Providence, RI* (1999).

[77] Myungho Lee, Nahal Norouzi, Gerd Bruder, Pamela J. Wisniewski, and Gregory F. Welch. 2018. The physical-virtual table: Exploring the effects of a virtual human's physical influence on social interaction. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST* (2018). DOI:https://doi.org/10.1145/3281505.3281533

[78] Xiaoxu Liu, Xiaoyi Feng, Shijie Pan, Jinye Peng, and Xuan Zhao. 2018. Skeleton tracking based on kinect camera and the application in virtual reality system. In *ACM International Conference Proceeding Series*. DOI:https://doi.org/10.1145/3198910.3198915

[79] Pedro Lopes, Sijing You, Lung Pan Cheng, Sebastian Marwecki, and Patrick Baudisch. 2017. Providing haptics to walls & heavy objects in virtual reality by means of electrical muscle stimulation. In *Conference on Human Factors in Computing Systems - Proceedings*. DOI:https://doi.org/10.1145/3025453.3025600

[80] Pedro Lopes, Sijing You, Alexandra Ion, and Patrick Baudisch. 2018. Adding force feedback to mixed reality experiences and games using electrical muscle stimulation. In *Conference on Human Factors in Computing Systems - Proceedings*. DOI:https://doi.org/10.1145/3173574.3174020

[81] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.

[82] Christoph Maggioni and Otto Hahn Ring. 1993. Gestural Input Device for Virtual Reality. *Image Processing* (1993), 118–124.

[83] Keigo Matsumoto, Yuki Ban, Takuji Narumi, Yohei Yanase, Tomohiro

Tanikawa, and Michitaka Hirose. 2016. Unlimited corridor. DOI:https://doi.org/10.1145/2929464.2929482

[84] Mark Mcgill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. 2015. A Dose of Reality: Overcoming Usability Challenges in VR Head-Mounted Displays. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15* (2015), 2143–2152. DOI:https://doi.org/10.1145/2702123.2702382

[85] Felipe A Medeiros, John K Zao, Yute Wang, Masaki Nakanishi, Yuan-Pin Lin, Alberto Diniz-Filho, and Tzyy-Ping Jung. 2016. The nGoggle: a portable brain-based method for assessment of visual function deficits in glaucoma. *Investigative Ophthalmology & Visual Science* 57, 12 (2016), 3940.

[86] Paul Metzger. 1993. Adding reality to the virtual. In *1993 IEEE Annual Virtual Reality International Symposium*. DOI:https://doi.org/10.1109/vrais.1993.380805

[87] Paul Milgram, Herman Colquhoun, and others. 1999. A taxonomy of real and virtual world display integration. *Mixed reality: Merging real and virtual worlds* 1, (1999), 1–26.

[88] Paul Milgram and Fumio Kishino. 1994. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems* 77, 12 (1994), 1321–1329.

[89] Eric R. Muth, Robert M. Stern, Julian F. Thayer, and Kenneth L. Koch. 1996. Assessment of the multiple dimensions of nausea: The nausea profile (NP). *Journal of Psychosomatic Research* (1996). DOI:https://doi.org/10.1016/0022-3999(95)00638-9

[90] David Nahon, Geoffrey Subileau, and Benjamin Capel. 2015. “Never Blind VR” enhancing the virtual reality headset experience with augmented virtuality. In *2015 IEEE Virtual Reality Conference, VR 2015 - Proceedings*. DOI:https://doi.org/10.1109/VR.2015.7223438

[91] Miguel Pedraza-Hueso, Sergio Martín-Calzón, Francisco Javier Díaz-Pernas, and Mario Martínez-Zarzuela. 2015. Rehabilitation Using Kinect-based Games and Virtual Reality. In *Procedia Computer Science*. DOI:https://doi.org/10.1016/j.procs.2015.12.233

[92] Thammathip Piumsomboon, Gun Lee, Robert W. Lindeman, and Mark Billinghurst. 2017. Exploring natural eye-gaze-based interaction for immersive virtual reality. *2017 IEEE Symposium on 3D User Interfaces, 3DUI 2017 - Proceedings* (2017), 36–39. DOI:https://doi.org/10.1109/3DUI.2017.7893315

[93] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. The go-go interaction technique: non-linear mapping for direct manipulation in VR. In *ACM Symposium on User Interface Software and Technology*, 79–80.

[94] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. DOI:https://doi.org/10.1109/CVPR.2016.91

[95] E C Regan and K R Price. 1993. Some side-effects of immersion virtual reality: An investigation into the relationship between inter-pupillary distance and ocular related problems. *Army Personnel Research Establishment Report 93R023* (1993).

[96] Zhou Ren, Jingjing Meng, and Junsong Yuan. 2011. Depth camera based hand gesture recognition and its applications in Human-Computer-Interaction. *ICICS 2011 - 8th International Conference on Information, Communications and Signal Processing* (2011), 1–5. DOI:https://doi.org/10.1109/ICICS.2011.6173545

[97] Louis B. Rosenberg. 1993. Virtual fixtures: perceptual tools for telerobotic manipulation. In *1993 IEEE Annual Virtual Reality International Symposium*. DOI:https://doi.org/10.1109/vrais.1993.380795

[98] Edward Rosten and Tom Drummond. 2006. Machine learning for high-speed corner detection. In *European conference on computer vision*, 430–443.

[99] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2.

[100] Mohamad Salimian, Stephen Brooks, and Derek Reilly. 2018. IMRce: A unity toolkit for virtual co-presence. *SUI 2018 - Proceedings of the Symposium on Spatial User Interaction* (2018), 48–59. DOI:https://doi.org/10.1145/3267782.3267794

[101] Samsung. 2015. Samsung Gear VR. Retrieved October 1, 2019 from https://www.samsung.com/global/galaxy/gear-vr/

[102] Carlos Santos, Tiago Araújo, Jefferson Morais, and Bianchi Meiguins. 2017. Hybrid Approach Using Sensors, GPS and Vision Based Tracking to Improve the Registration in Mobile Augmented Reality Applications. *International Journal of Multimedia and Ubiquitous Engineering* (2017). DOI:https://doi.org/10.14257/ijmue.2017.12.4.10

[103] Srivishnu Satyavolu, Gerd Bruder, Pete Willemsen, and Frank Steinicke. 2012. Analysis of IR-based virtual reality tracking using multiple Kinects. In *Proceedings - IEEE Virtual Reality*, 149–150. DOI:https://doi.org/10.1109/VR.2012.6180925

[104] Dominik Schmidt, Robert Kovacs, Vikram Mehta, Udayan Umapathi, Sven Köhler, Lung Pan Cheng, and Patrick Baudisch. 2015. Level-Ups: Motorized stilts that simulate stair steps in virtual reality. In *Conference on Human Factors in Computing Systems - Proceedings*. DOI:https://doi.org/10.1145/2702613.2725431

[105] Jongkyu Shin, Gwangseok An, Joon Sang Park, Seung Jun Baek, and Kyogu Lee. 2016. Application of precise indoor position tracking to immersive virtual reality with translational movement support. *Multimedia Tools and Applications* 75, 20 (2016), 12331–12350. DOI:https://doi.org/10.1007/s11042-016-3520-1

[106] Linda E. Sibert and Robert J.K. Jacob. 2000. Evaluation of eye gaze interaction. In *Conference on Human Factors in Computing Systems - Proceedings*. DOI:https://doi.org/10.1145/332040.332445

[107] Mel Slater. 1999. Measuring Presence: A Response to the Witmer and Singer Presence Questionnaire. *Presence: Teleoperators and Virtual Environments* (1999). DOI:https://doi.org/10.1162/105474699566477

[108] Kay M Stanney. 2002. *Handbook of virtual environments: design, implementation, and applications*. Retrieved from http://books.google.com/books?id=6GnFavzHZZcC&pgis=1

[109] Anthony Steed. 2008. A simple method for estimating the latency of interactive, real-time graphics simulations. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*.

DOI:https://doi.org/10.1145/1450579.1450606

[110] M LAVALLE Steven. 2016. Virtual Reality.

[111] Ivan E Sutherland. 1965. The ultimate display. *Multimedia: From Wagner to virtual reality* (1965), 506–508.

[112] Franco Tecchia, Giovanni Avveduto, Raffaello Brondi, Marcello Carrozzino, Massimo Bergamasco, and Leila Alem. 2014. I'm in VR! (2014), 73–76. DOI:https://doi.org/10.1145/2671015.2671123

[113] Stefaan Ternier, Roland Klemke, Marco Kalz, Patricia van Ulzen, and Marcus Specht. 2012. AR Learn: Augmented reality meets augmented virtuality. *Journal of Universal Computer Science* (2012).

[114] Bruce Thomas, Ben Close, John Donoghue, John Squires, Phillip De Bondi, Michael Morris, and Wayne Piekarski. 2000. ARQuake: an outdoor/indoor augmented reality first person application. *International Symposium on Wearable Computers, Digest of Papers* (2000). DOI:https://doi.org/10.1109/iswc.2000.888480

[115] James S. Thomas, Christopher R. France, Samuel T. Leitkam, Megan E. Applegate, Peter E. Pidcoe, and Stevan Walkowski. 2016. Effects of real-world versus virtual environments on joint excursions in full-body reaching tasks. *IEEE Journal of Translational Engineering in Health and Medicine* (2016). DOI:https://doi.org/10.1109/JTEHM.2016.2623787

[116] Sebastian Thrun. 2008. Simultaneous localization and mapping. *Springer Tracts in Advanced Robotics*. DOI:https://doi.org/10.1007/978-3-540-75388-9_3

[117] Keith Vertanen and Per Ola Kristensson. 2011. A versatile dataset for

text entry evaluations based on genuine mobile emails. (2011), 295. DOI:https://doi.org/10.1145/2037373.2037418

[118] Craig Villamor, Dan Willis, and Luke Wroblewski. 2010. Touch gesture reference guide. *Touch Gesture Reference Guide* (2010).

[119] Juan Pablo Wachs, Mathias Kölsch, Helman Stern, and Yael Edan. 2011. Vision-based hand-gesture applications. *Communications of the ACM* (2011). DOI:https://doi.org/10.1145/1897816.1897838

[120] David R. Walton and Anthony Steed. 2017. Accurate real-time occlusion for mixed reality. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST* Part F1319, (2017). DOI:https://doi.org/10.1145/3139131.3139153

[121] Wearvision.de. 2016. Microsoft HoloLens. *"Gewinn" Nr. 07-08/2016 vom 29.06.2016 Seite 103 Ressort: IT & Innovationen* (2016).

[122] P. Weber, E. Rueckert, R. Calandra, J. Peters, and P. Beckerle. 2016. A low-cost sensor glove with vibrotactile feedback and multiple finger joint and hand motion sensing for human-robot interaction. In *25th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2016*. DOI:https://doi.org/10.1109/ROMAN.2016.7745096

[123] Dong Wei, Steven Zhiying Zhou, and Du Xie. 2010. MTMR: A conceptual interior design framework integrating mixed reality with the multi-touch tabletop interface. In *9th IEEE International Symposium on Mixed and Augmented Reality 2010: Science and Technology, ISMAR 2010 - Proceedings*. DOI:https://doi.org/10.1109/ISMAR.2010.5643606

[124] Holger Weiss, Tobias Ortmaier, Heiko Maass, Gerd Hirzinger, and Uwe Kuehnapfel. 2003. A virtual-reality-based haptic surgical training system. In *Computer Aided Surgery*. DOI:https://doi.org/10.3109/10929080309146063

[125] Graham Wilson, Mark McGill, Matthew Jamieson, Julie R. Williamson, and Stephen A. Brewster. 2018. Object manipulation in Virtual Reality under increasing levels of translational gain. In *Conference on Human Factors in Computing Systems - Proceedings*. DOI:https://doi.org/10.1145/3173574.3173673

[126] John R. Wilson. 1996. Effects of participating in virtual environments: A review of current knowledge. *Safety Science* (1996). DOI:https://doi.org/10.1016/0925-7535(96)00026-4

[127] Bob G. Witmer, Christian J. Jerome, and Michael J. Singer. 2005. The factor structure of the Presence Questionnaire. *Presence: Teleoperators and Virtual Environments*. DOI:https://doi.org/10.1162/105474605323384654

[128] Bob G. Witmer and Michael J. Singer. 1998. Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoperators and Virtual Environments* (1998). DOI:https://doi.org/10.1162/105474698565686

[129] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D. Wilson, and Hrvoje Benko. 2018. MRTouch: Adding touch input to head-mounted mixed reality. *IEEE Transactions on Visualization and Computer Graphics* (2018). DOI:https://doi.org/10.1109/TVCG.2018.2794222

[130] Jiachen Yang, Yafang Wang, Zhihan Lv, Na Jiang, and Anthony Steed. 2018. Interaction with Three-Dimensional Gesture and Character Input

in Virtual Reality: Recognizing Gestures in Different Directions and Improving User Input. *IEEE Consumer Electronics Magazine* (2018). DOI:https://doi.org/10.1109/MCE.2017.2776500

[131] Keng Ta Yang, Chiu Hsuan Wang, and Liwei Chan. 2018. ShareSpace: Facilitating shared use of the physical space by both VR head-mounted display and external users. In *UIST 2018 - Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. DOI:https://doi.org/10.1145/3242587.3242630

[132] Ryota Yoshimoto and Mariko Sasakura. 2017. Using real objects for interaction in virtual reality. In *2017 21st International Conference Information Visualisation (IV)*, 440–443.

[133] André Zenner and Antonio Krüger. 2017. Shifty: A Weight-Shifting Dynamic Passive Haptic Proxy. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 23, 4 (2017), 1285–1294. Retrieved from https://doi.org/10.1109/TVCG.2017.2656978%5Cnhttp://www.dfki.de/web/forschung/publikationen/renameFileForDownload?filename=Shifty-TVCG2656978-Author-Version.pdf&file_id=uploads_3047

[134] Hui Zhang. 2017. Head-mounted display-based intuitive virtual reality training system for the mining industry. *International Journal of Mining Science and Technology* (2017). DOI:https://doi.org/10.1016/j.ijmst.2017.05.005

[135] Yiwei Zhao, Lawrence H. Kim, Ye Wang, Mathieu Le Goc, and Sean Follmer. 2017. Robotic assembly of haptic proxy objects for tangible interaction and virtual reality. *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces, ISS 2017* (2017), 82–91. DOI:https://doi.org/10.1145/3132272.3134143

[136] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. 2008. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In *Proceedings of the 7th IEEE/ACM international symposium on mixed and augmented reality*, 193–202.

[137] Oculus Rift. Retrieved July 1, 2019 from http://www.oculusvr.com

[138] HTC Vive. Retrieved October 1, 2019 from http://www.vive.com

[139] Omni by Virtuix. Retrieved October 1, 2019 from https://www.virtuix.com/

[140] Leap Motion. Retrieved November 1, 2019 from https://www.ultraleap.com/

[141] Go Touch VR. Retrieved July 1, 2019 from https://www.gotouchvr.com

[142] Plexus Immersive. Retrieved October 1, 2019 from http://plexus.im/

초 록

실감형 가상현실 경험은 HMD(Head Mounted Display)기기의 발전과 보급에 따라 시각 및 청각적 가상체험이 필요한 상황에서 폭넓게 활용되고 있다.

본 연구에서는 최근의 일반적인 HMD기반의 가상현실 상에서 문제점으로 지적되고 있는 비 직관적인 입력 방식, 그리고 가상환경내에서의 물리적 고립(Isolation) 문제를 보완하고자 현실에 존재하는 물리적 스마트폰을 가상환경으로 도입하여 이를 가상현실 내 입력장치로 활용하도록 한다. 복잡한 컨트롤러보다 보다 사용자에게 익숙한 기기의 도입을 통해 기존의 플랫폼 상 상호작용시 필요로 하였던 프록시(Proxy) 인터페이스의 활용을 필요치 않도록 하며 기존의 방식에 비해 더욱 직관적이고 사용성 높은 플랫폼을 구현하도록 한다. 기존의 HMD 장비에 외부 모니터링이 가능한 카메라를 통하여 외부 환경의 스마트폰을 가상현실 내부로 가져오도록 하며, 이는 가상현실의 배경에 현실의 오브젝트를 오버레이 시키는 증강가상현실(Augmented Virtuality)의 개념을 활용하는 것이다. 외부에서 들어오는 이미지에서 목표 오브젝트를 제외한 백그라운드를 제거하여 실제 오브젝트의 질감을 가져올 수 있도록 분리를 진행하고, 이를 사용자가 입력장치로 활용할 수 있는 새로운 형태의 플랫폼을 제안한다.

본 연구에서의 구현 과정에 있어 다양한 종류의 성능 실험과

사용자 실험을 동반하게 되며, 일반적인 컨트롤러 환경에서의 결과와 비교하도록 한다. 사용자 실험을 통해 제안하는 방식의 한계점과 강점을 검증하며, 증강가상현실 상 필요한 정확한 시각적 표현 방식의 제안과 사용자의 가상현실 경험에 대한 인지에 영향을 줄 수 있는 부분에 대한 가이드라인을 제시하도록 한다. 제시하는 방식을 토대로 한 증강가상현실 플랫폼은 사용자에게 기존의 가상현실 환경 보다 더욱 높은 입력 성능을 보여주었으며, 더 좋은 인지 결과를 보여주었고, 현실과 가상환경 사이에 존재하고 있는 물리적 격차를 줄였다는 점을 파악할 수 있다. 이를 토대로 본 연구의 제안 방식은, 점점 커지고 있는 증강가상현실 및 혼합현실(Mixed Reality)의 연구 분야에서 오브젝트를 매개체로 하여 현실과 가상환경을 이어주는 하나의 도구로 활용될 수 있을 것으로 기대한다.

주요어: 가상현실, 입력 인터페이스, 인간-컴퓨터 상호작용, 혼합현실, 증강가상현실
학 번: 2014-30810