Sliding Gaze : Head Movement and Touch Gesture Based Target Selection Technology

Bo Yang

44171637-0

Master of Engineering

Supervisor: Prof. Jiro Tanaka

Graduate School of Information, Production and Systems

Waseda University

July 2019

Abstract

Target selection technology plays an important role at all times when we try to interact with some elements in human computer interaction. For wearable devices, head movement and touch gesture are wildly used as the input source to select the target. Head movement is a deliberate and Concise method for target selection, and is wildly used in the wearable devices. However, it requires an unnatural and fatiguing head actions, and is not as precise as the touch input. For touch gesture, it is a mature technology in the human computer interaction for users to do the precise and high-quality input, but in the case of wearable device, touch gesture input need the support of an external input device.

This research explores the Slide Gaze: using both head movement and touch gesture techniques for wearable devices. We use a head mounted display for head movement input and the visual output, as well as a touch pad for the touch gesture input. User can choose to use two different kinds of input methods. The refine mode use touch gesture input as a precise refinement for the head movement input to select the target with high accuracy. Meanwhile, the overlay mode superposes the two kinds of input sources together to achieve a faster selection.

Since the sensitivity of head movement input is influenced by the distance of the object, we also need to adjust the sensitivity of touch gesture input to get a qualified input experience. We also provide several methods for users to use touch gesture to do further interaction with the selected objects.

Keywords: Target Select, Head Movement, Touch Gesture, Wearable Device

Acknowledgements

At the beginning, I am very glad to express my sincere gratitude and appreciation to my supervisor Prof.Jiro Tanaka, a very respectable and responsible mentor with rigorous academic attitudes and creative ideas. He has given me lots of help and encouragement, and is always very patient for guiding me to solve problems. Thanks a lot to him for the many valuable and earnest advices during every stage of preparation of this thesis.

Besides, it is grateful to all my dear labmates. They discussed a lot with me and always give me lots of inspiration. Thanks so much for their many helps in experiments so that our experiments could be performed successfully and the good results could be get. What is more, I want to express the depth of my gratitude to the people from Rakuten Project, Mr. Kelvin, Mr. Matsuda, Ms. Ochiai. I truly appreciate their help for my research.

Last but not least, I would like to thank my parents and all love ones. For my parents, they give me financial support and warm mental support throughout my master study life. I will keep going on and making progress to return all those people.

Contents

List of figures

Chapter 1

Introduction

1.1 Introduction

Recently available head-worn Augmented Reality (AR) devices will become useful for mobile workers in many practical applications. By using those applications, we can control the network of smart objects, shopping in the scene recognition-based AR Shops, sharing traveling experience with others who never need to go out. The Figure 1.1 shows a scene using AR application to share a joint shopping experience with another in-door user. In fact, almost all of those AR applications need users to select some targets. For users work with high mobility, it is very important to design the interaction technology which allows user to select and control virtual object with high accuracy in complex environments.

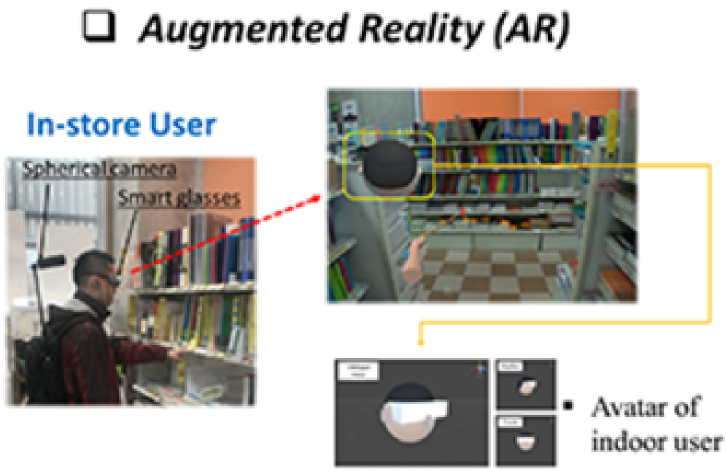
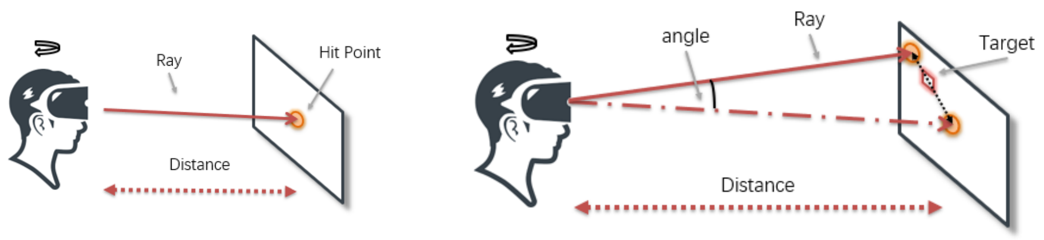


Fig. 1.1 AR Joint shopping system

For those wearable AR devices, currently head movement is a fairly precise way for target selection. As we all know, it is a concise and practice way for input, but it also has some problem such as requiring some unnatural, fatiguing actions. Head movement input method for the wearable devices usually shots a ray from the center of the main camera along the direction of head pointing, and select the object which is hit by the ray. The direction of the ray would be changed with the user's head movement, which is used to control the ray and do interact with the virtual object.

However, when we want to select an object in the distance, a little change of the ray angle will cause a large movement of the hit point. It means that if user want to select a target in distance, using head movement to do that may not be a good idea. As shown in the Figure 1.2, the head movement input is a kind of rotational input. When the size of the object is constant, the precision of head movement input is related with the distance between user and object. Meanwhile, touch gesture input does not have this limitation. So we try to find a way to combine the advantage of head movement input and touch gesture input, in order to create a well-performed target selection technology.

In this study, we tend to use the touch gesture to adjust the input of head movement in order to achieve a precise target selection method. We use the microsoft HoloLens as a wearable AR device, and the iPhone X as a touch screen.



(a) Head Movement Input (b) In Distance

Fig. 1.2 The difficulty for head movement input to select target in the distance.

We designed two kind of input methods. First one called refine mode, using the touch gesture to adjust the input after the end of the head movement. In this method, head movement and touch gesture both work independently. We can use head movement to move the ray to somewhere near the target, then switch to the precise touch gesture input and select the target. The Second one called overlay mode. As its name, in this method, the final input result is the overlying of head movement and touch gesture.

Besides, we also implement several functions for user to interact with the selected target by using touch gestures, such as zoom the object or drag the object.

1.2 Organization of the thesis

The rest of the thesis is organized as follows: Chapter 2 introduces the background about the thesis and also the related works in this field. Chapter 3 will tell the research goal and also the approach will be told briefly. Chapter 4 is the system design part, where the design concept and ideas will be introduced and the algorithm design will also be told. Chapter 5 will be the system implementation part where the detailed environment and implementation will be talked. Chapter 6 will introduce the related work. Chapter 7 will be about the experiments, we will talk about the performance of our approach and the comparison of different approached will also be done. The last part, Chapter 8, will be conclusion and future work part, where we will conclude the previous content and talk about the future possibilities.

Chapter 2

Background

2.1 Target Selection in Augmented Reality Applications

Augmented reality (AR) is an interactive experience of a real-world environment where the objects that reside in the real-world are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory and olfactory.

The target selection input methods for wearable augmented reality can mainly be divided in two kinds: the input using body gestures which is captured by the inner sensor of head mounted display, such as head movement [1], hand gesture on air, eye tracker[2] and so on. The other input method relies on the external device such as controller, touch screen and laser pen. Unlike the virtual reality, in the wearable augmented reality application, user need to walk and turn around in the real world to catch and interact with the virtual objects, with physical movements[2]. Since user need to interact with the real-world object as well as the augmented information, it becomes very important to design a target selection technology with precise input and support diverse interaction methods, without the unportable input device.

2.2 Gesture Based Target Selection Techniques

For the wearable augmented reality, considering the portability, in this research we concentrate mainly on the gesture input methods, without large devices which need users use two hand to do input and those input devices which is hard to do eye free input.

2.2.1 Hand Gesture

The in-air hand gesture is also regarded as a input method with lots of potential. At first, they use the inertial measurement unit of a hand-held device to get the track of the device and accept it as a kind of input. With the development of depth camera and pattern recognition, it is now becoming possible for those augmented reality devices to directly recognize the in-air hand gesture and capture the information as input[4].

The hand movement is both natural and precise if we have a powerful and portable sensor to capture all of the information. Limited by the technology, it is hard to set a reliable border for gestures recognition as a precise input. What's more, waving the hand in public may also involve the problem of leaking privacy, just like the voice control[3]. In this way, using a portable touch screen to catch the hand gesture is a well solution for wearable augmented reality to do target selection input.

2.2.2 Head Movement

The head movement pointing input is wildly used for the head mounted display for its concise and portability. The head movement is captured as a rotational movement input source like the laser based pointing technical. The head-based interaction methods are also attracting the attention in the field of augmented reality, In fact, head movement input is wildly accepted as a practical input method for head mounted display, such as Microsoft HoloLens[4], Oculus Rifts, and so on.

2.2.3 Eye Gaze

The eye tracking technology, which can recognize the object user is watching attentively, help eye gaze to be a kind of pointing method. Compare with other input methods, eye gaze can select an object with exactly small movement, without hand and voice[5]. However, as it is hard to distinguish the interaction function of eye gaze with the basic function of eye, to look and recognize something[6]. The well-known problem is also called the “Midas Touch Problem”. In order to solve that, we may need to use a independent input, such as a button press or hand gesture, as the secondary interaction functions[7].

2.3 Related Work

The related works will be introduced in this chapeter. There will be two kinds of related works, the gesture based target selection Techniques and the Combining Pointing Techniques.

2.3.1 Related Work about Target Selection Techniques

Since target selection is one of the basement interaction, We can find a number of researches, concentrate on using gestures as a input source to design some target selection techniques.

Stephen Brewster et al.[8] explored an interaction technique for mobile scene and eye-free input. Their research showed that the head movement and hand gesture with a support of guide sound can significantly improve the usability of a wearable device.

Rowel Atienza et al.[9] inspected the head-based interaction for virtual reality contents. The indirect control functions that do not allow users to use their hand and feet degrade the level of immersion. They got a result with reliability and improvement on terms of user control by using head movement.

Lutteroth et al.[10] explored a target selection technique using only eye gaze tracking. In order to improve the accuracy of gaze input and make it possible to resolve small objects, they prefer a method called Actigaze, which add distinguishable visual identifiers to the confirm button with fixed position, and let the user interface elements directed associated with the confirm button to minimizes the likelihood of inadvertent clicks.

Yuanchun et al.[11] compared several different text entry input methods for head mounted display. The input technique was based on head movement, and they provided several methods for confirm a selection, including tap, dwell and word-gesture. As a result, after training the users reach a speed of 24.73 WPM. Based on these results, they argue that head-based text entry is feasible and practical on HMDs, and deserves more attention.

2.3.2 Related Work about Combining Pointing Techniques

Next, We will introduce some work on compound input combining different input methods. Using a precise input method to complement a rough input method, is wildly used in various input scenarios. When we want to drag an object to a specific location, we can use mouse first for rough movement, and then use keyboard for precise adjustment.

The combination of different input methods is extensively used to improve the speed[12] or the precision[13] of rough input. Ken Pfeuffer et al. used the touch input with eye gaze to fetch various input scenarios, such as touch screen, touch pen[14] and tablet computers[15]. The gaze input, which is regarded as a direct input, can extend the reach of the indirect input, touch gesture. Mean while the touch gesture can used to do some interaction function like click, zoom, drag, as well as adjust the rough input by eye gaze to improve the accuracy. The refinement function they used, is called cursor shift which is very close to our idea, using a precise input to adjust the rough input for target selection.

Mikko Kytö et al.[16] furtherly explored the methods in augmented reality environment. They explored a refinement technique, using hand input or precise head input to adjust the input of eye gaze or head movement, for the wearable augmented reality application. In their research, in order to superimpose different types of input, they are all unified for rotational angle input.

Stellmach et al.[17] investigated the gaze-supported target selection methods. They propose gaze-supported interaction as a more natural and effective way combining a user's gaze with touch input from a handheld device. The techniques called magic touch and magic tap,just like the magic point resarch for gaze and point[18]. In this research, the technique uses touch gesture to support gaze input, demonstrated a high overall performance and usability for desktop environment.

Like the combination of direct input and indirect input, the combination of rotational input and translational input is also a meaningful research direction[19]. When we use the laser pan, we naturally combine the two way of input together as we can rotate our wrist or move our hand translationally in a nature way[20]. When we specify the head movement as the rotational movement input, since it is not so convenient for head to do a translational movement, we need to add another input function. Using a simillar method like cursor shift, Stellmach and Dachsel [21] also used a hand-held touch device to support the head movement and gaze input for a distance display[22].

In summary, there are several researches about gesture-based target selection techniques. The device, algorithm, user cases have many differences but also share some commons. Since all of the prior works have focused on selecting target on a certain surface, whereas few researches have explored target selection on different surfaces with different distance. What's more, most of the studies on the combination of different input methods did not delve into the relationship between distance and calibration accuracy.

Chapter 3

Research Goal and Approach

3.1 Goal

The fundamental goal of our research is to **combine touch gesture and head movement to find a target selection technology for AR application.**

Much previous research on improving gaze inputhas used hand input or head input for refinement focused on the desktop environment, or neglected the influence of distance in the AR environment, like the work of [16] . However, in the ray based selection system, sensibility changes with the distance between user and target. We want to achieve a high accuracy target selection in the complex AR environment, so our system should also satisfy these 3 requirements:

1. Could be used in Portable scene;
2. Could be used to select target with high accuracy;
3. Could be used for selecting an object precisely in various distances.

3.2 Use Case

Assume the following situations: 1 user wants to select a target from a list of menus, the target is 5 meters away from user.

The operation process, for the target, is: user move his head and focus the ray roughly on the menu. The cursor would be generated on the hit point. At this time, they may find it hard to select the target as the movement of ray is too rough for precise input requirement. Then, he can use touch gesture input to adjust the position of cursor which is much more precise than head movement input in this distance. With the refinement from touch gesture input, user can easily select the target in distance.

The use case above will be shown in Fig. 3.1. At first the user rotated his head to do a rough input. When he moved the cursor to a place near to the target, he can use the touch gesture to refine the input and select the target. In our system, the object will turn red when it is selected.

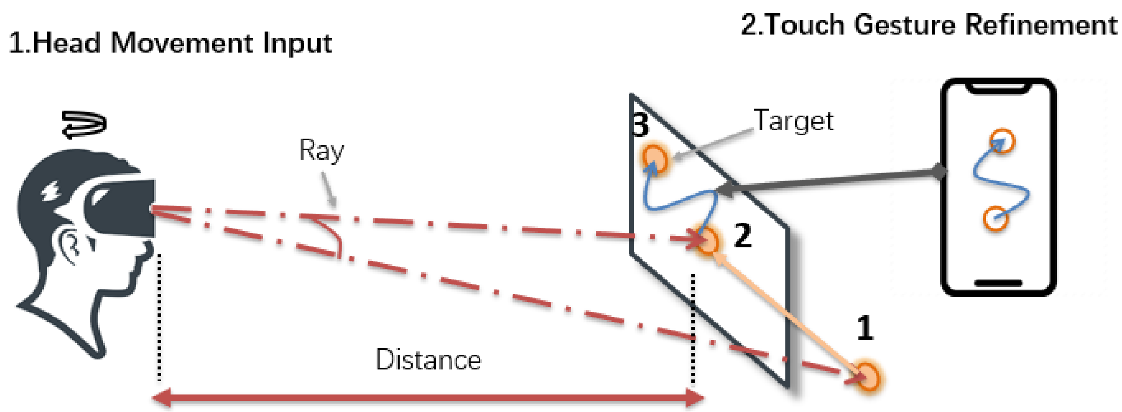


Fig. 3.1 Refinement by Touch gesture

3.3 Approach

To achieve the goal, we will introduce our approach of on the following aspects. Firstly, two key aspects will be introduced: the combination of head movement and touch gesture, the variable sensibility for adjustment based on distance.

Combine a rough input method with a precise input method to complement the low accuracy of the rough method is a common theme to improve the input experience. Head pointing is wildly used in the field of augmented reality and wearable devices, for its advantages of providing hand-less input and high portability[23] . Head pointing uses the ray-casting pointing technology and is well known as a kind of absolute position method, using rotational movement for pointing from a distance[24] . The performance and correctness of head pointing is considered not as good as hand input. Touch gesture input, which is regarded as a kind of precise way of input, is a kind of relative input, using translational movement, as its value depends on the touch trace. It also has its limitation as it is difficult for it to move a cursor for a long distance. Therefore, we tend to combine the two kind of input methods together to achieve an accurate selection technology. The combination of rotational input and translational input may be possible for the movements of higher degrees of freedom in the AR applications[20] .

The next aspect is the variable sensibility for adjustment based on distance. In order to adapt the requirement of selecting targets in different distances, we prefer to change the sensibility of touch gesture input to get a better result. As the head movement input is a kind of ray-casting pointing technology. It uses the angular motion to do input, which is completely different from the touch gesture based relative input. For the far distance target, we need a precise touch input to adjust the head movement input. On the other way, for the target in middle distance, the sensitivity of touch input need not to be as precise as the far one, because the head movement become more precise. In our research, we prefer to change the sensibility of touch gesture to match the head movement input in order to get an accuracy and convenience user experience.

To verify our approach achieves our goal and prove the effectiveness of our approach, we will perform several experiments. What's more, in our system user can also interact with the selected object. After selecting the target using head movement with touch gesture, user can use the touch pad to do some fundamental operations, such as click, drag and zoom the object[15] . Also we further discuss the result of these experiments and try to design some potential applications for our system.

3.4 Novelty

The novelty of our research mainly reflects in these aspects:

1. We propose combining touch gesture with head movement to design a general target selection technology for AR environment;
2. We propose using the variable adjustment sensibility to improve the performance;
3. Most prior works focused on desktop environment or a fixed distance, we propose to find a solution for selecting targets in different distances.

Chapter 4

System Design

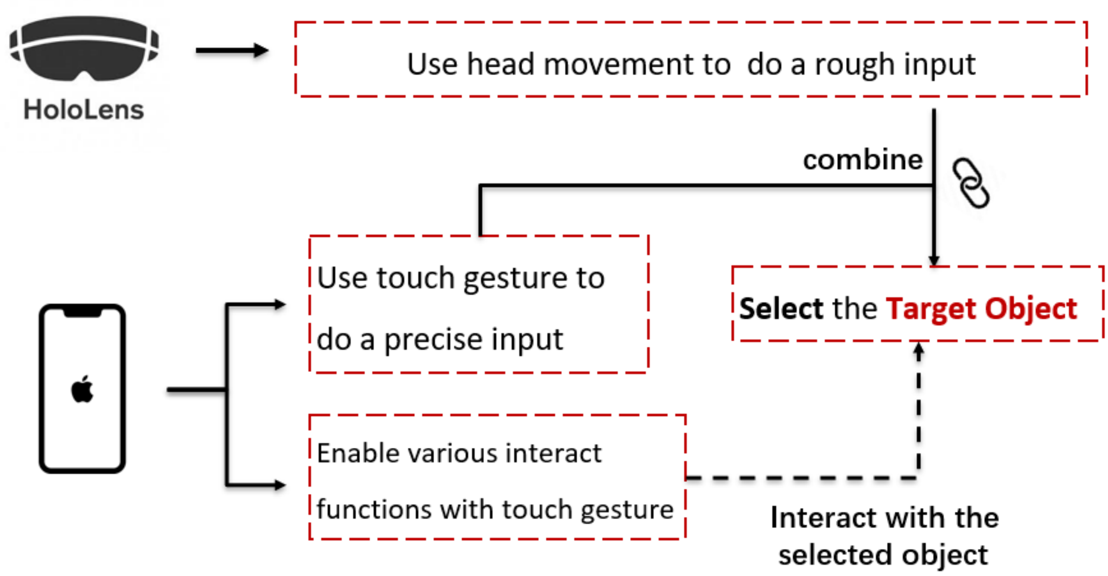


Fig. 4.1 Overview of our approach

In this chapter, we will introduce our system design and each point of our approach. The Fig. 4.1 shows the overall structure of the algorithm. We get the head movement input from HMD, and get the touch gesture input from smart phone touch screen, then we use the two kinds of input to design a target selection technique. We will divide the algorithm description into four parts:

- Part 1 introduces the head movement input from head mounted display and touch input from touch screen;
- Part 2 introduces how we combine the two input methods together.
- Part 3 introduces the strategy of variable sensiblity.
- Part 4 introduces the various interaction methods in our system.

4.1 Gesture Input Source

In our proposed system, the head movement act as an angular absolute-input to build a part of the selection technology. In this research, we use the HoloLens to catch the head input. HoloLens is a see-through type head mounted display, which features an inertial measure unit (including an accelerometer, gyroscope, and a magnetometer) and a 2.4-megapixel photographic video camera. The rotate of head can be catched by the inertial measure unit(IMU). With the rotational input, we can rotate the ray and select some object.

HoloLens also help us to capture the hand gesture such as air-pinch and air-drag. The energy-efficient depth camera with a 120°×120° angle of view can recognize the hand gesture from user and accept it as a source of input, which can also be used to adjust the primary input.

We use a Smart Phone, the iPhone X to catch the touch gesture input from users. A Capacity Touch Panel for the smart phone can capture the touch trace easily and precisely. What is more, we can also use the IMU of smart phone to get the device movement trace. It is also a choice of input source to refine the head movement input.

4.2 Combination Gesture Methods

Next, we discuss about the fundamental methods in our research to combine the two different kind of input. At first, there are two elements to construct the selection technology: the accidence input by head movement, and the refinement method to adjust the head input. The key point is how to combine the two different methods with different sensibility and input sources together.

4.2.1 Refine mode and Overlay mode

At first, we discuss the refine mode which is wildly used in the previous researchs to combine two kinds of different input source. In the refine mode, The main process of our method works like this:

1. Point the head to move the cursor to an approximate location.
2. Using a more precise to adjust the position of the cursor, such as touch gesture input, hand gesture input, precised head movement input, and so on.

In the refine mode, the head movement is used as the initial input. The head movement input is a template angular input for the head mounted display and augmented reality applications. Using head movement, user can point the cursor to a certain direction concisely, in a short time.

The secondary refinement input techniques make up the disadvantage of head movement input with a high precision. Though we prefer to use the touch gesture to adjust the head input, in the research we provide several different translational input methods as the precise refinement input. The list of techniques is showed in the Figure 4.2.

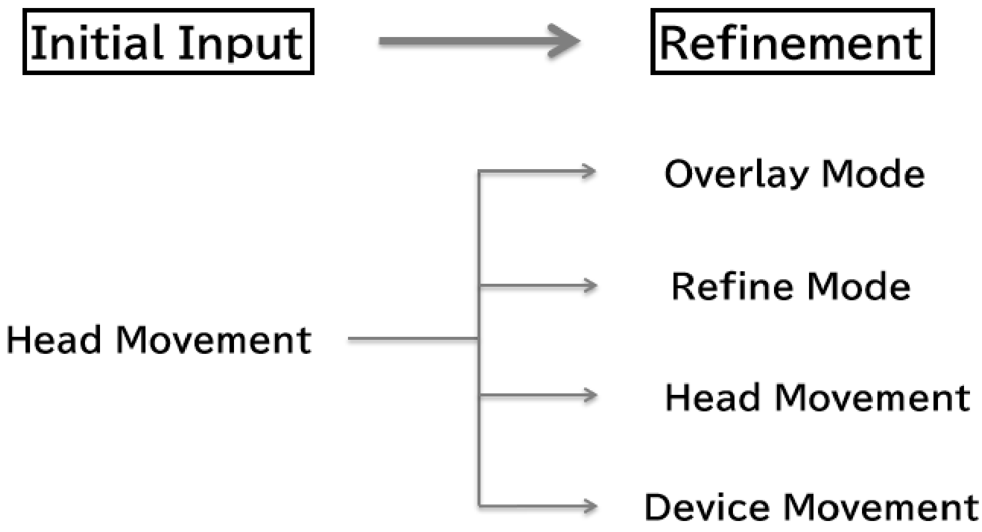
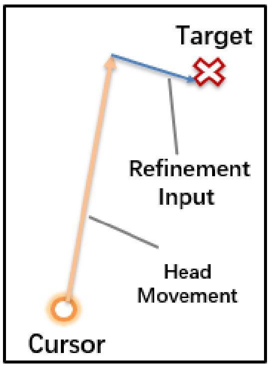


Fig. 4.2 Refinement Methods for refine mode

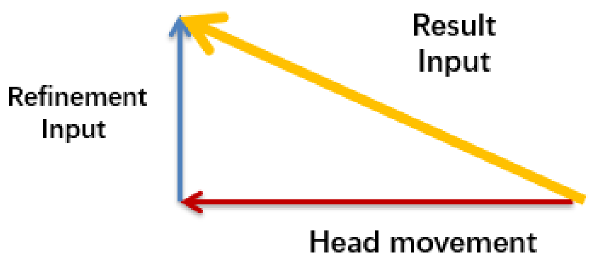
As we can see, we have four refinement methods in our system. Though we prefer the touch gesture method, we also implement other three methods in our system, for further evaluation. For each refinement methods, the Overlay Mode and Refine Mode use the smart phone as a touch pad to get the trace of touch as input. The Head Movement method, as the refinement method, is different from the initial head movement input. The control-display (CD) ratio is tuned to make it a more precise input, though it may cause some problem as the field of view of head mounted display is still changing in the same ratio. The Device Movement uses the IMU in smart phone to capture the translational movement of device itself and regard it as an input source.

In order to active the refinement input, we also need a trigger to stop the rough input and start the precise input. For the touch gesture based refinement method, the trigger function can be devided in two types: the manual type and the automatic type. In our system, the manual type is called refine mode, and the automatic type is called overlay mode.

The refine mode use the manual trigger to start the refinement. For the input methods, touch gesture, device movement as well as head movement, user only need to tap on the touch screen for one time to end the initial input and start the refinement input. For the input method hand gesture, user need to do hand pinch to trigger the function. In the refine mode, the refinement input and the initial input work independently. It means that the head movement input would be completely disabled when the refinement input has been triggered.



(a) Refine Mode



(b) Overlay mode

Fig. 4.3 Two different Trigger Methods

However, in the overlay mode, things are different. This time, user do not need to use trigger or action to start the refinement input. The refinement input is exactly overlying with the initial input all the time, just as shown in the Figure 4.3. With the overlay mode, user can get a seamless input experience, as well as avoiding the time lose in the trigger action. But since the head input always exists, the accuracy of the input will be affected.

In our research, we use several experiments to test the performance of the two different refinement modes, in different conditions. We also compare them with the native one, and we will explain the result in detail in the experiment part.

4.3 Variable Sensibility

The head movement input, for head mounted display, is a little different with the laser pan as the pointing method. In fact, when using the laser pan, user can move their hand translationally to move the ray translationally. However, for the head movement, it is very difficult to move the head translationally as user almost need to move their whole body to achieve that. So that, we tend to consider head movement input as a pure rotational input technique.

In order to explain our propose, we need to introduce some concepts first. The task for user is to move the cursor from start point to the target. If the action is only finished by the head movement, the head need to rotate the pointing ray to the target. The plane structure of our system is shown in Figure4.4. In this part, we set a refinement area, which means that the cursor would change to refinement method when it has been in this area. The refinement area shows a field for which, user can use head movement only input to select this area in a satisfactory speed and accuracy. Size of the target is fixed, and the gray line distance means the distance between target surface and user. The size of refinement area is recorded by angle, and will be increased with the distance between target surface and user.

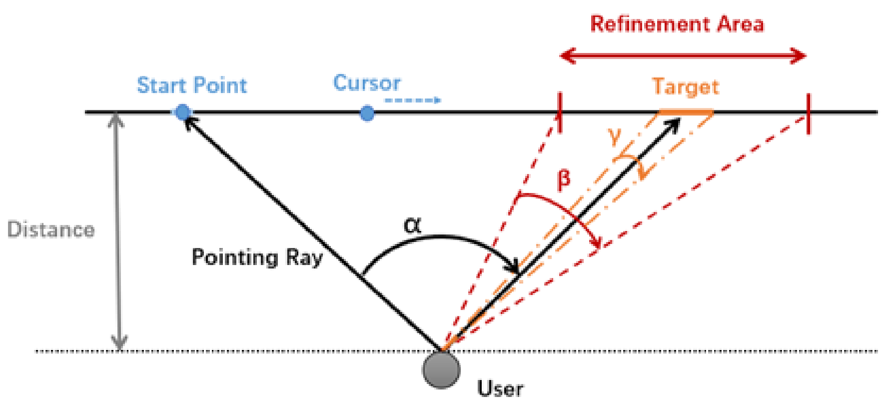


Fig. 4.4 The process of target selection in near distance for head movement

For the rotational movement, the key factor is the angle of head movement. If the refinement area's angle range for head are same with each other, the head movement user needs to finish is also the same. However, in this case, the real size of the refinement area, varies linearly with the distance between user and target.

The idea is that, as we assumed, the head movement is too rough for user to select the target in refinement area, we need to use a more precise secondary refinement to approve the input accuracy. User need to do a translational input to move the cursor from the edge of the refinement area to the target. The Figure 4.5 shows the same scene in far distance. The angle areas for target object and distance to move are changed. However, the angle for refinement area keeps constant.

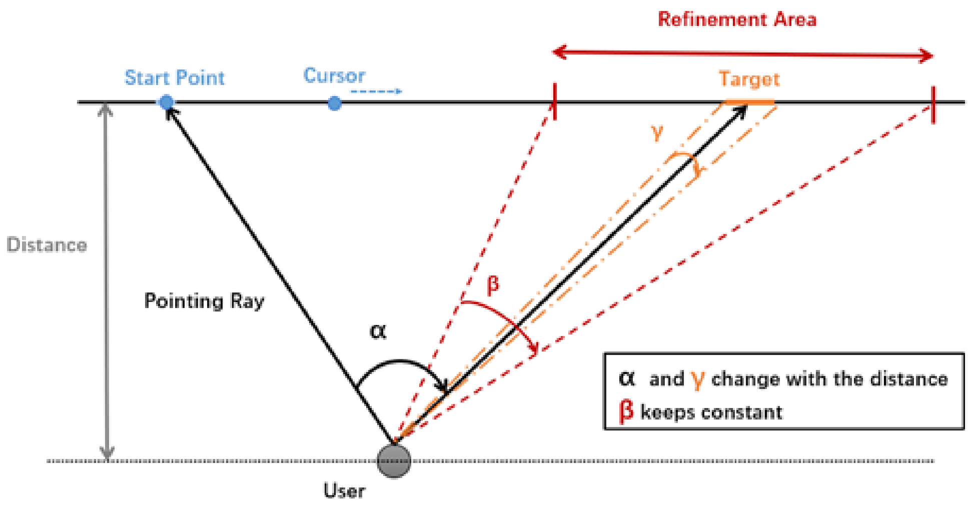


Fig. 4.5 The process of target selection in far distance for head movement

Unlike the rotational input, the translational speed of cursor in translational input is fixed and can only be changed manually. With the increasing of distance and width of the refinement area, in our function the sensibility of cursor in translational input is also changed to adapt it, to match the change of rotational input in our method. This function can provide users with a smooth input experience with out the feeling of interruption when they try to select another target in different distances. The translational speed of precise input and rough input need to be in the same or near ratio wherever the target is.

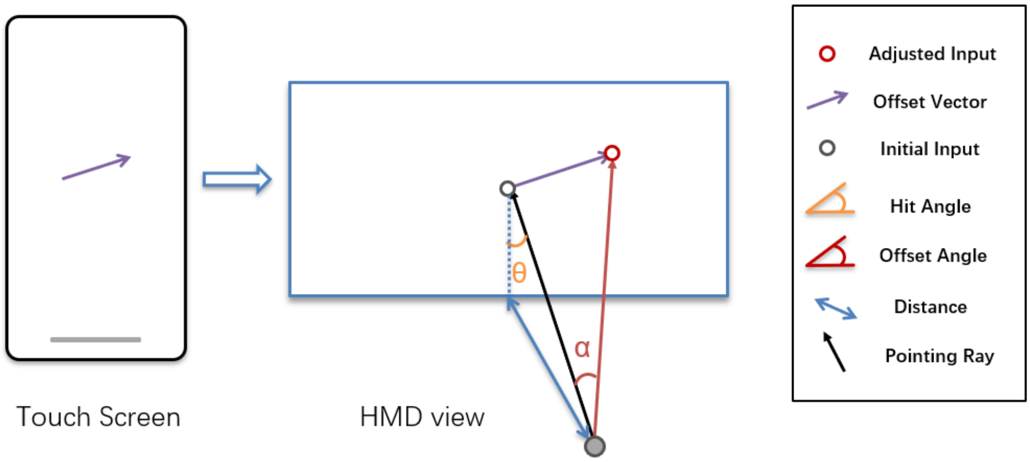


Fig. 4.6 Refinement Methods for refine mode

In our research, we use the moving distance per unit angle of cursor to represent the sensibility of input method. In order to get such performance, we need to calculate the actually offset angle to adjust the head pointing ray. The value we used is shown in the Figure 4.6. At first, we can use the hit angle θ and the distance between user and plate to calculate the length of the initial pointing ray L. Since the translational movement of cursor depends on the rotation angle ω and the length of pointing ray L, the sensibility of head movement input S_H can be shown as this equation.

$$f(\omega) = \omega * D / \cos(\theta) \tag{4.1}$$

$$S_H = f'(\omega) \tag{4.2}$$

For the refinement input, as the sensibility of refinement should vary linearly with the one of initial input, we can get the target sensibility and calculate the actual translational movement of cursor on the target surface. The offset vector $\vec{O}$ which is actually used to change the cursor's position can be calculate with the input vector $\vec{I}$ and a constant value r which can be used to control how precise the refinement input is.

$$\vec{O} = r * S_H * \vec{I} \tag{4.3}$$

In this way, we calculate the cursor speed of precise input in real time depends on the distance between target and user to improve our methods. After the variable sensibility function, the adjustment from translational input can work stable when the distance changes continuously.

4.4 Interaction with Touch gesture

The external input device, touch screen, can not only capture the touch trace to move the cursor, but also enrich the interaction functions between user and the selected target.[1] In fact, the touch screen can also capture the multi-pointing input, as well as the special touch gesture in certain field. Using the various input functions, we can provide user with some colorful interaction methods which can enrich the experience of the wearable augmented reality application.

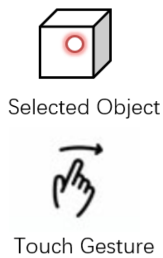


Fig. 4.7 Ineract mode for touch gestures

After selecting target through our methods, we also provide an interaction mode on the touch screen side application. In the interaction mode, user can use several different touch gestures to interact with the argument reality object for basic interaction, such as click, drag, zoom, delete, and so on.

Chapter 5

System Implementation

5.1 Hardware and Data preprocess

In this research, we use the HoloLens as the see-through type head mounted display to implement the AR applications. The iPhone X is used as a touch pad with IMU.

Fig. 5.1 The devices used in our research: HoloLens and iPhoneX

HoloLens can capture the distance using its depth camera, and it can also capture the head movement as well as several simple hand gestures by user. The touch screen supports a multi touch with force sensor, and can also get the movement of device itself by IMU.

The Figure 5.2 shows how we use the ray select an object. The camera symbol represents the user's visual range in the HMD. The green line is the ray we used to select object, and the cursor will be generated on the surface if the ray hit the surface. If users rotate their head, the green line will rotate with the head movement and point to the place they want. In fact, Ray-pointing techniques are often advocated as a way for people to interact with object several meters away.

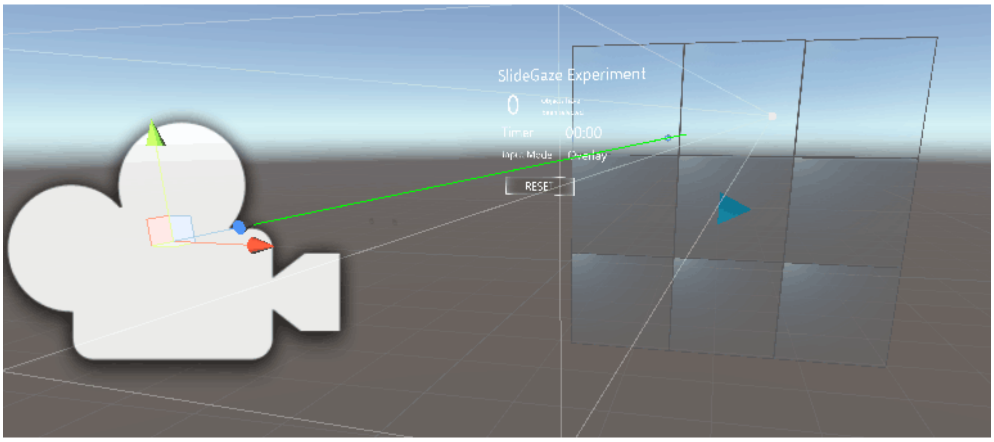


Fig. 5.2 Ray casting based Head movement input for Augmented Reality

After the rough input by head movement, we need to get the precise input from the smartphone touch screen. Since the input data from touch gesture is simple and short, we pay more attention to the efficiency of transmitting messages. In our research, we use the OSC(Open Sound Control) protocol to construct a data communication platform between different devices.

Open Sound Control (OSC) is a protocol for networking sound synthesizers, computers, and other multimedia devices for purposes such as musical performance or show control. OSC's advantages include interoperability, accuracy, flexibility and enhanced organization and documentation. OSC messages are transported across the internet and within local subnets using UDP/IP and Ethernet.

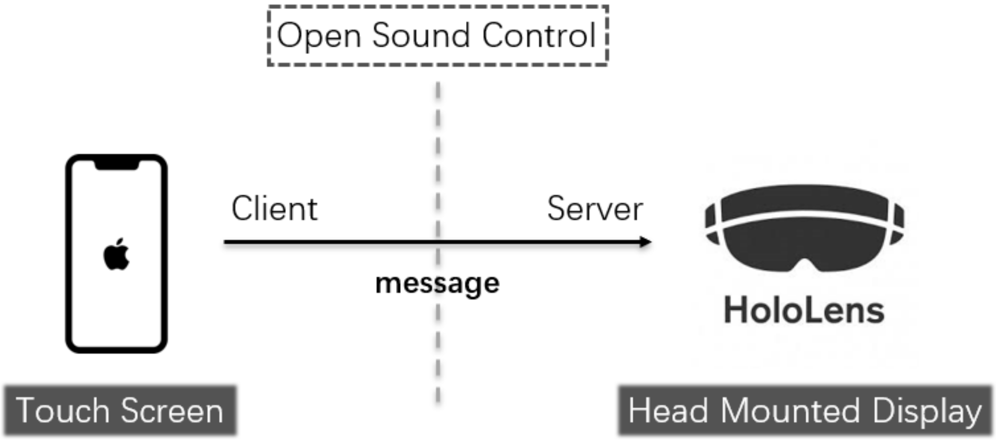


Fig. 5.3 Message send and receive structure

For every frame, when the touch screen catches the valid touch input, it will send a message with several data of different types to the head mounted display. The message contents the following parameters: time stamp, the current input mode, the touch point data from the touch screen, and the type of gesture. Figure 5.3 shows us the main framework of the system. If there are more than one finger touch the screen, the function will send all of the data in order.

```
//
void OnDataReceived(Message message)
{
    //clear the temp data used last time
    transformReceived.Clear();

    // address
    msg = message.address + ": ";

    // timestamp
    msg += "(" + message.timestamp.ToLocalTime() + ") ";

    List<float> cursorPoint = new List<float>();

    // values
    foreach (var value in message.values)
    {
        msg += value.GetString() + " ";
        if (value is int) inputMode = (int)value;
        Debug.Log(inputMode);
        if (value is float) cursorPoint.Add((float)value);
    }

    //write the received data
    transformReceived.Add(cursorPoint[0]);
    transformReceived.Add(cursorPoint[1]);
```

Fig. 5.4 On received function to process the data strings

Meanwhile on the head mounted display side, there is also a function “onDataReceive()” which is called when the process received a message. The contents of the function is shown in the Figure 5.4. In this function, first we filter the message and get the target value, based on the current input mode. Also, we need to calculate the track of touch gesture from the input data of touch point to get the offset we want for refinement. The value of the touch point track will be stored in a vector variable, which can be used to adjust the initial input.

5.1.1 Implementation of Combined Input

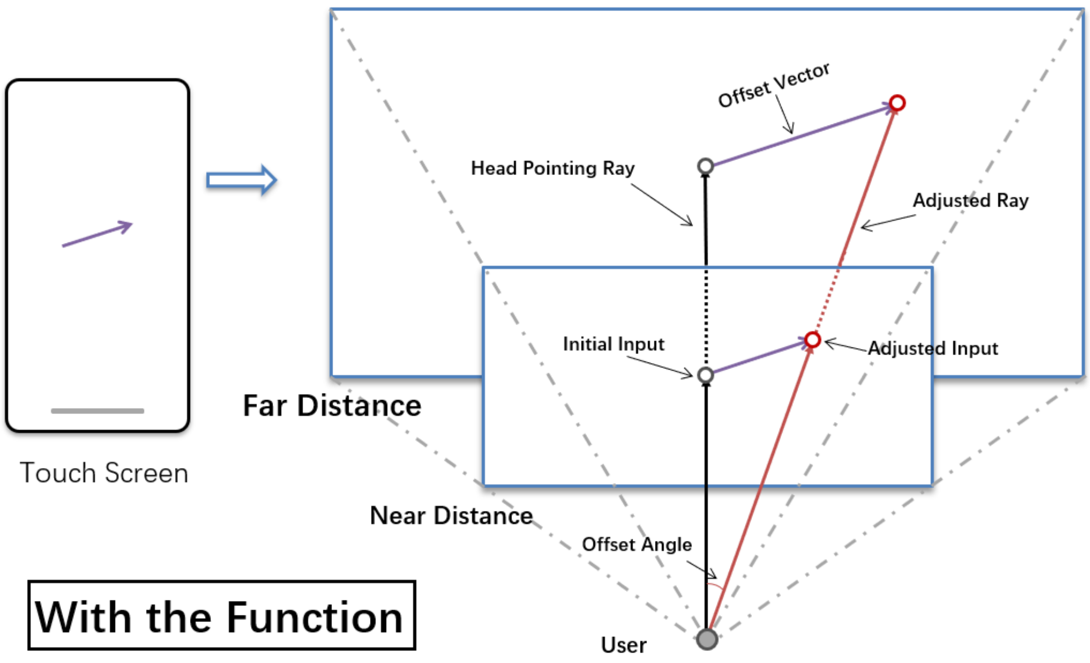


Fig. 5.5 Getting the offset vector from touch gesture input to adjust the head movement input depends on the distance

For the head mounted display side, we need to use the vector variable we get to adjust the head movement input. The head movement input can be captured as a angle to control the pointing ray, which is shot from the main camera of augment reality contents. As a result, usually the cursor is on the center of the field of view. The cursor will be generated at the place where is hit by the ray, usually on some AR objects. When there is nothing to hit, the cursor will be generated at a default distance. In this research, the cursor is generated 2 meters away from user when the ray hits nothing, and the farthest distance from which the cursor can be generated, is 10 meters.

After receiving the data from smart phone, we need to use the value to adjust the pointing ray. In the system design part, we have introduced about how to use the distance and a fixed translational movement to calculate the offset angle.The Figure 5.5 shows the situation to get an offset vector from a touch gesture on touch screen, and use it to adjust the initial input by head movement. We can get the length of the ray from the structure RayCasting from Unity. We can also get the angle between hit surface and the pointing ray. Using the length and the angle, we can get the distance between user and the target surface. Then we can get the offset angle and use it to adjust the pointing ray, calculating from the offset vector and the distance. The Figure 5.6 shows the result if we do not use the various sensibility function. The offset vector for adjustment changes by a certain ratio, which dose not take into account the impact of distance changes on input accuracy.

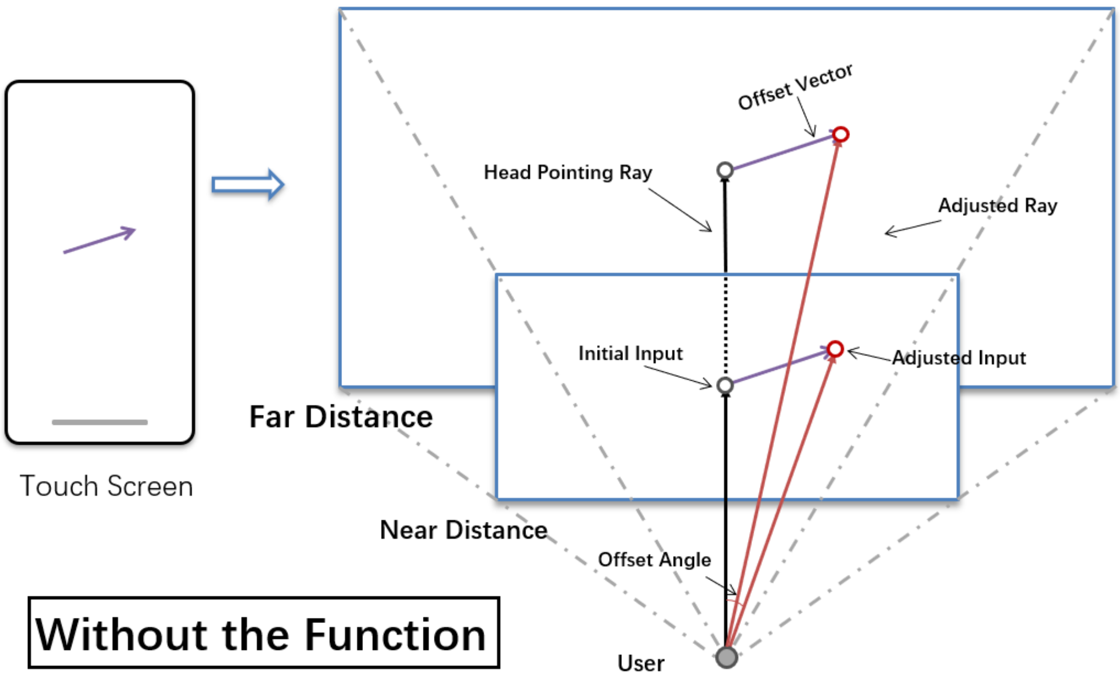


Fig. 5.6 Getting the offset vector from touch gesture input to adjust the head movement input without the various sensibility function

```
/// <summary>
/// Updates the current gaze information, so that the gaze origin and normal are accurate.
/// </summary>
private void UpdateGazeInfo()
{
    if (GazeTransform == null)
    {
        Rays[0] = default(RayStep);
    }
    else
    {
        Vector3 newGazeOrigin = GazeTransform.position;
        Vector3 newGazeNormal = GazeTransform.forward;

        // Update gaze info from stabilizer
        if (Stabilizer != null)
        {
            Stabilizer.UpdateStability(newGazeOrigin, GazeTransform.rotation);
            newGazeOrigin = Stabilizer.StablePosition;
            newGazeNormal = Stabilizer.StableRay.direction;
        }
        Vector3 change = Vector3.zero;
        if (transformReceived.Count > 0)
        {

            if(inputMode == 2)
            {
                change = Vector3.zero;
                refineModeOnFlag = 0;
                refineInstance = Vector3.zero;
            }
            //calculate the translational movement value

            else change.Set(variableSensibility(transformReceived[0]), variableSensibility(transformReceived[1]), 0);

        }
        if(inputMode != 3) {
        Rays[0].UpdateRayStep(newGazeOrigin, change + newGazeOrigin + (newGazeNormal * FocusManager.Instance.GetPointingEx
        }
        if(inputMode == 3&&refineModeOnFlag ==0)
        {
            refineModeOnFlag = 1;
            refineInstance = newGazeNormal;
            Rays[0].UpdateRayStep(newGazeOrigin, change + newGazeOrigin + (refineInstance * FocusManager.Instance.GetPoint
        }
```

Fig. 5.7 Adjust the head pointing ray using the received data

The core function to adjust the position of the pointing ray is shown in the Figure 5.7. Meanwhile, the data of the operation time, the movement trace and the input from device and touch screen for each task will be collected and stored in the data structure called TaskData, which can be used for further evaluation.

5.2 Graphical User Interface

Figure 5.8 shows the user interface of the touch screen. The circle on the center works as the feedback for touch gesture since it will move with the touch finger and show the track of touch input. The current input module is shown on the bottom of the user interface.

Fig. 5.8 Smart Phone side User Interface

As we have talked in the system design part, we have four kinds of input mode. Overlay mode, refine mode, device mode and the interact mode. The first three input mode is used as a refinement method to adjust the head movement input, and the interact mode focus on interacting the selected object using touch gesture input. User can slide on the bottom side of the screen to switch the mode between the different input methods.

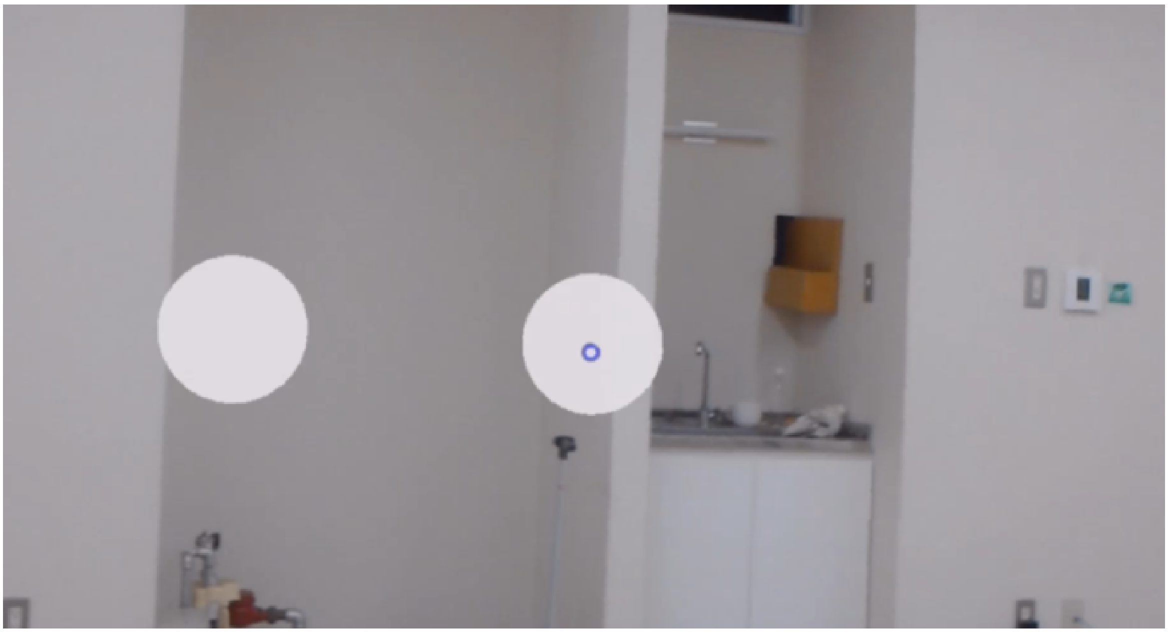


Fig. 5.9 Cursor Appearance in HMD view

On the other hand, for the head mounted display side, we just concentrate on the appearance of cursor which is used to select the object. The Figure 5.9 shows the appearance of cursor on a simple white sphere. When the pointing ray hits a surface, the cursor will become a hollow ring which is attached to the hit point. However, when the point ray hits nothing, the cursor looks like a blurred circle on the certain distance.

What is more, the color of cursor will change depends on the current input mode. For example, the cursor will turn yellow on the overlay mode, turn white on the refine mode. The change of color can help user cognite the current input mode intuitively, and help to achieve an eye-free input as user does not need to look at the touch screen.

5.3 Experimental Tasks

We designed several experimental tasks for Participants to test the performance of our system. There are four main factors used to design the experiments:

1. Selection techniques;
2. Distance between target and participant;

3. Number of targets that need to be selected;

4. Whether to use the variable sensibility function.

There are five selection techniques in our research, to compare with each other. Head movement only, head movement with touch gesture, head movement with head movement and head movement with device movement.

The distance between target and participant mainly influence the selectable angle size of the target. We need to check the near distance and far distance. For the near distance, the virtual object shows on the surface two meters from user, while for the far distance, the object is five meters from user.

We also considered the situation when user need to select several targets continuously. For the one-target task, user only need to select one target to finish the experiment, for the three-targets task user need to select three targets. The targets are generated automatically, within a specific area. The distance between different targets is fixed, but the direction and order are disrupted.

We designed two kind of experimental task for evaluation works. First is the Target Selection task, used to test the speed of the input methods. participant need to select one or more target object on the surface, and system will record the time of task. The second one is the Menu Selection task, used to test the accuracy of the input methods. participant need to focus the cursor to a certain button from a list. The target button will be presented above the menu, if participant select a wrong button, the result will be error.

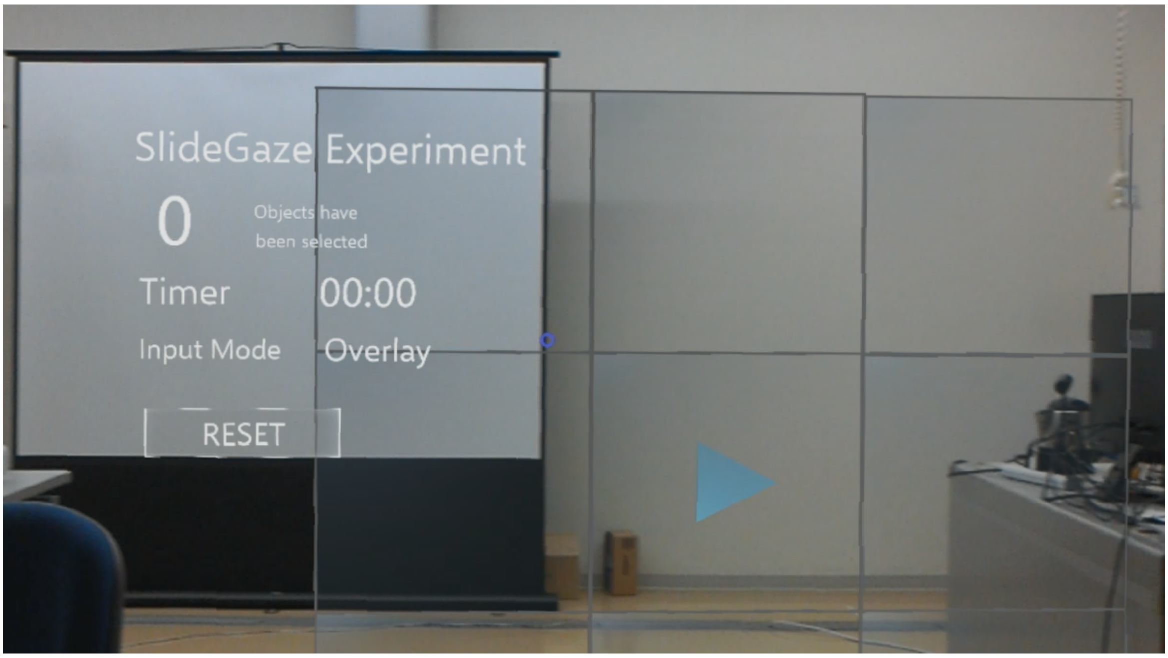


Fig. 5.10 Target Selection Experimental task view

The figure 5.10 shows the view of Target Selection task for user. The informational text is showed on the top left of the surface. We create a surface which is kept in certain distance from participant, which is used to place the targets. Target surface is divided into nine area, used to control the distance between two different targets. The start point is on the center of the target surface.

To start a task, participant need to move the cursor to the start trigger, the blue triangle, and stay for a short time to select it and start task. When the task begins, start trigger will turn black and the targets will be generated in the certain blocks of the surface, by a predetermined amount. The target is selected and turn to red when the cursor hit the target for over 0.5 second. After all target being selected, the task will be finished and the time cost will be recorded.

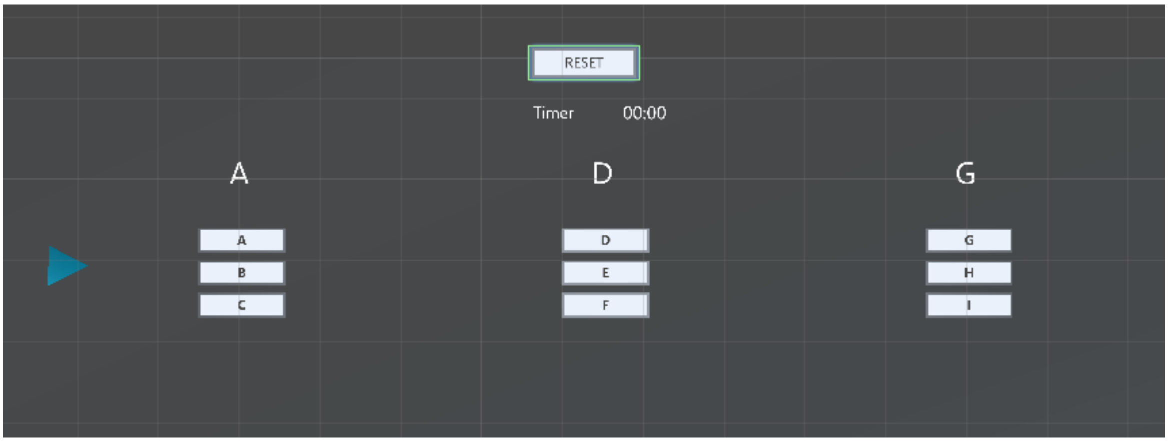


Fig. 5.11 A sample scene for menu selection in Experiment

The Figure 5.11 shows the main structure of Menu Selection task. The targets button which is supposed to be selected, are presented on the top side of each menu. Same to the Target Selection task, participant need to select the start trigger to start the experiment. Then participant need to select the correct button in order, from three different menus. The time cost and the result, success or failure, will be shown on the top text.

In order to do further analysis, we collect the cursor's position, when it starts and ends the refinement method, and the detailed time cost for each step. After the experimental task, we ask the participants to fill out a questionnaire, asking about the experience and the feeling of each input methods.

Chapter 6

Experiment and Result

In order to prove the performance of target selection, the experiments mainly aim to collecting the result of speed and accuracy of our input refinement system as well as the origin one without adjustment. The To realize that experiment thinking, we will perform an experiment and the result will also be shown.

The experiment environment is mainly introduced in the Chapter 5. We create a modifiable experimental scene in wearable augmented reality application, and work for the target selection in different situations. In order to minimize the learning effects, the experiments used the Latin square design for each participant.The Figure 6.1 shows one scene when participant is about to start the target select experiment with overlay mode.

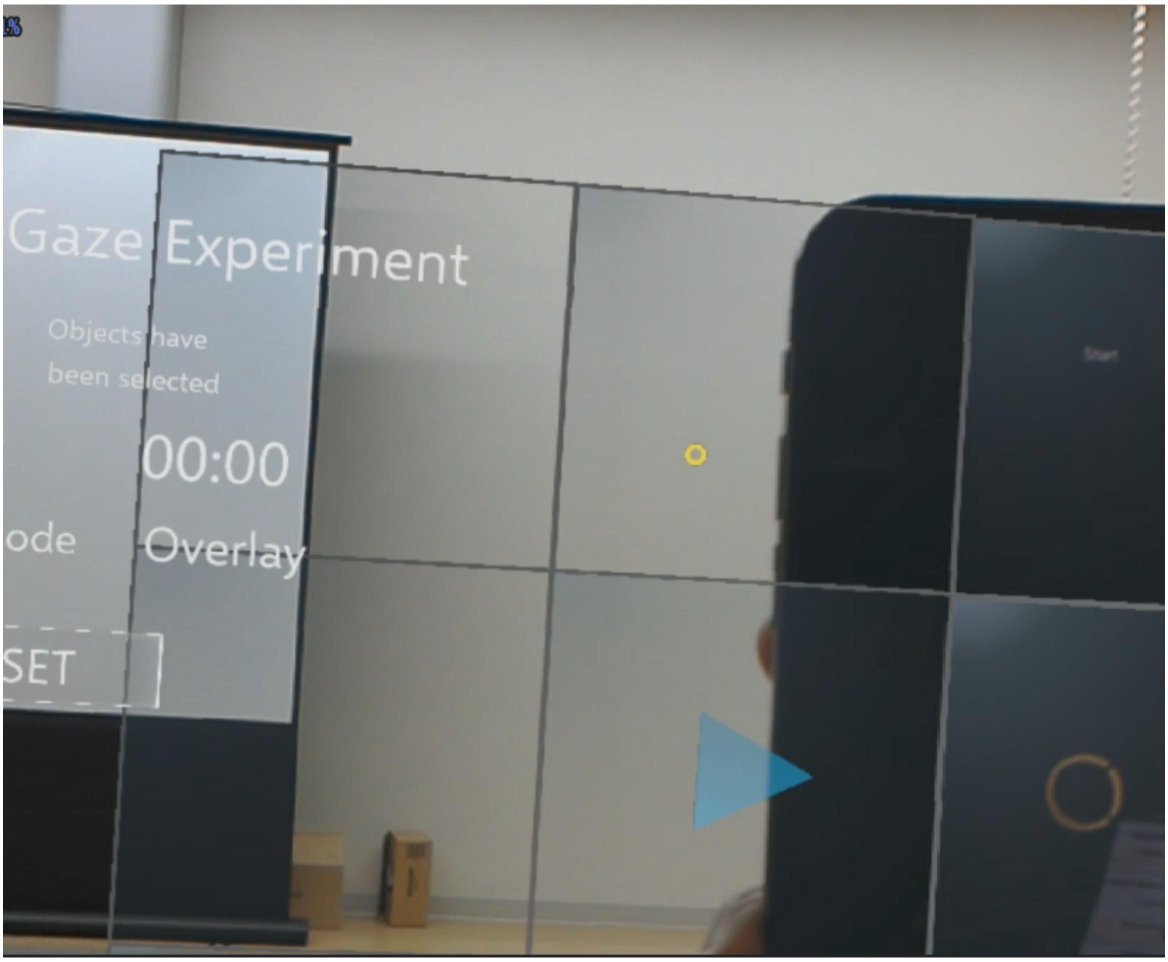


Fig. 6.1 A sample scene for target selection in Experiment

6.1 Experiment

In order to test the performance of our system, we need to get the result data of target selection tasks for all of the different target selection techniques. We make a experimental application to evaluate the performance of our approach. The detailed function of the application is introduced in the Chapter 5. In one of the experimental task, we collect the mean time of each trial and each phase. The mean time for target selection need to deduct the interaction time cost.

For participants, we asked 11 users of 20 to 26 years old who had no experience of this kind of system but they were taught about how to use our system.

1. In order to start a test task, the participant need to align the cursor to the start target which is on the center of the surface. The task will start after a random time between 1.5 second to 3 second. The target object, one or more virtual solid circles with blue color will be generated on the target surface when the task is activated. For menu based task, only the target button of menu should be actived, falsely actived buttons are considered as errors;

2. They were asked to move the cursor and select the target object as fast as possible, and as correct as possible. The time and the moving trace of each step would be collected by the application, which would be used for analysis;

3. Each participant need to finish the task of five different input methods, for two kinds of distances, for one or more targets in one task, as well as for whether to use the variable sensibility function. The total number of task for each user will be 5*2*2*2 = 40 times.

Note that in every stage, their original data will be saved for further analysis. From the experiment, we will analyze the result by:

1. Mean time and error rate for each task to select one target object, as the basic performance of target selection technique;

2. Comparison of five different input methods, the head only input, the head-head input, the head-gesture input with refine mode, the head-gesture input with overlay mode, the head-device input;

3. Comparison of two different discances of target objects: the far distance for 5 meters and the near distance for 2 meters;

4. Comparison of the performance for whether to use or not use the variable sensibility function;

5. Comparison of our approach and previous works.

6.2 Result

In order to caculate the relationship between distance and sensibility, we used the famous Fitts's law to prove our methods. Fitts's law is a predictive model of human movement primarily used in human–computer interaction, and is well-used to model the speed=accuracy trade-off for the human movement. This equation illustrates Fitts' law according to the Shannon formulation (MacKenzie, 1989).

$$MT = a + b\log_2(1 + D/W) \tag{6.1}$$

As we can see in this equation, the model shows that the mean time(MT), to finish a movement, varies linearly with the index of difficulty(ID). D is the distance from the starting point to the center of the target and W is the width of the target. a and b are two constants depends on the input device. Mostly a means the time cost for start and end input. At the same time, b means the speed of device, which is called the sensibility of device in our research. This type of Fitts's law is used for the translational movement in simple one-degree applications, such as move the hand from right to left.

6.2.1 Comparison by Mean Times

The primary results for mean times are summarized in Figure 7.2 and 7.3. As an objective measure, we collect the mean times for the participants to finish each target.

For the near distance, the width of target is 0.1 meter, and the angle for target is 2.87 degree. The distance between target and the start point is randomly selected from a range of 11.4 degree to 34.2 degree. In this situation showed in Figure 7.2, we can see that the head-gesture input method with overlay mode perform better than other one, however, just slightly better than other methods. Compared with the four different kinds of input method, in near distance the performances of various methods are very close. The mean time of overlay mode, is only 5.53 percent better than the mean time of head movement only input. The head movement only input method is regarded to be sufficient for this task, so that an unnecessary refinement operation may make the mean time longer.

Fig. 6.2 Mean times with different input methods for near distance

For the far distance, the width of the target is the same but the distance between participant to target surface changes from 2 meters to 5 meters. So that the angle of target width also changes, from 2.87 degree to 1.43 degree. The result for far distance shows in the Figure 7.3, we can see that with the decrease of target angle, the difficult for participants to select the target increased significantly, just as the result of Fit's Law. The most significant increasement is head movement only input, which is increased by almost 23.2 percent. Meanwhile, all of the refinement techniques show a better performance than head movement in the far distance situation. Similar to the near task, the head + gesture method with overlay mode works better than others, 13 percent better than head movement only method. In our consideration, the head + device refinement method seems difficult to use, due to the mode switch between finger touch to arm movement.

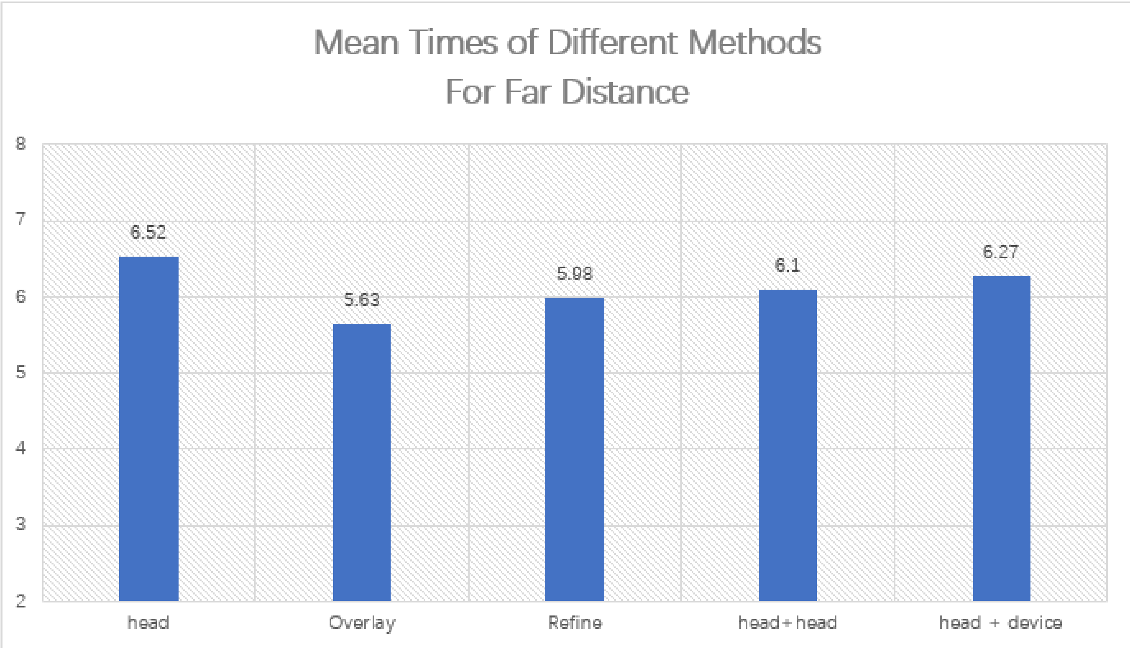


Fig. 6.3 Mean times with different input methods for far distance

6.2.2 Comparison by Accuarcy

The Figure 7.4 shows the error rate of each different input methods in the menu selection task. In this task, all of the buttons including target one and others, can be selected after a certain dwell time. Each task requires users to select three buttons from three menu which contains a list of buttons.

Fig. 6.4 Error rate with different input methods

The result shows that, for the head movement based input, all of the requirement techniques improved the accuracy of head only input method. Head + gesture input method with the refine mode, particularly performs best in accuracy. The technique improves significantly, nearly three times compared with the basic head only input, from 0.14 to 0.047. The performance of overlay mode in accuracy is obviously inferior to several other methods. Comparing with other refinement methods, in overlay mode, the rough head movement input is still enabled in refinement phrase, which can be a reason for the result. What is more, the head + head method and the head + device method also get a better accuracy result than the head + gesture method with overlay mode.

Overall, the head movement only input method works well in the near distance situation, which explains the reason why it is wildly accepted as a practical input method for head mounted display. For the refinement methods, in general, all of them significantly improved the accuracy of head only method. The head + gesture method with overlay mode has the best selection speed among those methods, as well as the worse accuracy. Meanwhile, the head + gesture method with refine mode is the most precise way of input, which improves the performance of head only input for three times.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In general, this research explores the Slide Gaze: using both head movement and touch gesture techniques for wearable devices.

Target selection technology plays an important role at all times when we try to interact with some elements in human computer interaction. For wearable devices, head movement and touch gesture are wildly used as the input source to select the target. Head movement is a deliberate and Concise method for target selection, and is wildly used in the wearable devices.

We use a head mounted display for head movement input and the visual output, as well as a touch pad for the touch gesture input. User can choose to use two different kinds of input methods. The refine mode use touch gesture input as a precise refinement for the head movement input to select the target with high accuracy. Meanwhile, the overlay mode superposes the two kinds of input sources together to achieve a faster selection. Since the sensitivity of head movement input is influenced by the distance of the object, we also need to adjust the sensitivity of touch gesture input to get a qualified input experience. We also provide several methods for users to use touch gesture to do further interaction with the selected objects.

A user study of our system shows the performance and the speed-accuracy trade-offs of our methods. Comparing with the several methods by previous works, the overlay mode input method has the best speed for target selection, meanwhile the refine mode is preciser than all other methods.

7.2 Future Work

For the future work, we would like to combine more kinds of input methods to find the possibility of target selection for wearable device. For example, the eye tracing and the in-air hand gesture, are very interesting and potential input method for wearable devices. What is more, we also want to find out some techniques for high-speed selection of a large number of targets, for the AR applications which may need more complex input. For example, using the movement of index finger as a laser pointer, as well as tring to combine it with the head movement and eye gaze.

References

[1] Jeremy Hales, Diako Mardanbeigi, and David Rozado. Interacting with objects in the environment by gaze and hand gestures. In *17th European Conference on Eye Movements*, volume 6, 2013.

[2] Hyung Min Park, Seok Han Lee, and Jong Soo Choi. Wearable augmented reality system using gaze interaction. In *Proceedings of the 7th IEEE/ACM international Symposium on Mixed and Augmented Reality*, pages 175–176. IEEE Computer Society, 2008.

[3] Ishan Chatterjee, Robert Xiao, and Chris Harrison. Gaze+ gesture: Expressive, precise and targeted free-space interactions. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 131–138. ACM, 2015.

[4] Microsoft. The leader in mixed reality technology hololens. *Internet WWW-page, URL: http://www.microsoft.com/en-us/hololens(10.07. 2017)*, 2017.

[5] Aulikki Hyrskykari, Howell Istance, and Stephen Vickers. Gaze gestures or dwell-based interaction? In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 229–232. ACM, 2012.

[6] Linda E Sibert and Robert JK Jacob. Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 281–288. ACM, 2000.

[7] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, and Hans Gellersen. Gaze-touch: combining gaze with multi-touch for interaction on the same surface. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 509–518. ACM, 2014.

[8] Stephen Brewster, Joanna Lumsden, Marek Bell, Malcolm Hall, and Stuart Tasker. Multimodal'eyes-free'interaction techniques for wearable devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 473–480. ACM, 2003.

[9] Rowel Atienza, Ryan Blonna, Maria Isabel Saludares, Joel Casimiro, and Vivencio Fuentes. Interaction techniques using head gaze for virtual reality. In *IEEE Region 10 Symposium*, pages 110–114. IEEE, 2016.

[10] Christof Lutteroth, Moiz Penkar, and Gerald Weber. Gaze vs. mouse: A fast and accurate gaze-only click alternative. In *Proceedings of the 28th annual ACM symposium on user interface software & technology*, pages 385–394. ACM, 2015.

[11] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. Tap, dwell or gesture?: Exploring head-based text entry techniques for hmds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4479–4488. ACM, 2017.

[12] P ar Anders Albinsson and Shumin Zhai. High precision touch screen interaction. In *In Proceedings of the 2003 CHI Conference*, page 105–112. ACM, 2003.

[13] Clifton Forlines, Daniel Vogel, and Ravin Balakrishnan. Fluid switching between absolute and relative pointing with a direct input device. In *Proceedings on User Interface Software and Technology*, page 211–220. ACM, 2006.

[14] Ken Pfeuffer and Hans Gellersen. Gaze-shifting: Direct-indirect input with pen and touch modulated by gaze. In *Proceedings of the 28th Annual Symposium on User Interface Software and Technology*, page 373–383. ACM, 2015.

[15] Ken Pfeuffer and Hans Gellersen. Gaze and touch interaction on tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, page 301–311. ACM, 2016.

[16] Kytö Mikko, Ens Barrett, Piumsomboon Thammathip, Gun A. Lee, and Billinghurst Mark. Pinpointing: Precise head- and eye-based targetselection for augmented reality. In *In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 81. ACM, 2018.

[17] Sophie Stellmach and Raimund Dachselt. Look & touch: gaze-supported target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2981–2990. ACM, 2012.

[18] Shumin Zhai, Carlos Morimoto, and Steven Ihde. Manual and gaze input cascaded (magic) pointing. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 246–253. ACM, 1999.

[19] David C. McCallum and Pourang Irani. Arc-pad:absolute+relative cursor positioning for large displays with a mobile touchscreen. In *Proceedings of the 28th Annual Symposium on User Interface Software and Technology*, page 153–156. ACM, 2009.

[20] Martin F. Stoelen and David L. Akin. Assessment of fitts' law for quantifying combined rotational and translational movements. *Human Factors: The Journal of the Human Factors and Ergonomics*, 15(52):63–77, 2010.

[21] Sophie Stellmach and Raimund Dachselt. Still looking: Investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. In *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 285–294. ACM, 2013.

[22] Ken Pfeuffer, Jason Alexander, and Hans Gellersen. Partially-indirect bimanual input with gaze, pen, and touch for pan, zoom, and ink interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2845–2856. ACM, 2016.

[23] Nancel Mathieu, Chapuis Olivier, Pietriga Emmanuel, Yang Xing-Dong, IraniPourang, and Beaudouin-Lafon Michel. High-precision pointing on large wall displays using small handheld devices. In *Proceedings of the SIGCHI Conference on Human Factors and Computing Systems*, page 10. ACM, 2013.

[24] Jota, Nacenta, Jorge, Carpendale, and Greenberg. A comparison of ray pointing techniques for very large displays. In *Proceedings of graphics interface*, pages 269–276. CIPS, 2010.