A Study on Visual Interface on Palm and Selection in Augmented Space

Graduate School of Systems and Information Engineering

University of Tsukuba

March ２０１３

Seokhwan Kim

Abstract

This study focuses on the interaction of target acquisition in augmented space. In the augmented space, there will be more and more devices, and all the devices will be interactive. For having the interaction with a device, obviously the users need to access the devices first. The target acquisition means the processes of accessing to a device, and this study seeks the ways of enabling promising means of the easier accessing.

Generally this target acquisition requires some physical movement (i.e., holding a remote controller or approaching a physical device). To remove such physical movements, this study proposes an always available interface, which is called the palm display and given by the infrastructure in the space. Consequently, users do not need to move to access a device, because users can interact with a device using the interface on palm (i.e., the palm display).

There have been several studies to provide such always available interfaces on body. However, there has been no work to enable an always available interface on the palm using the infrastructure and also supporting direct manipulation metaphor, which is generally used in common mobile devices. The palm display enabled it and a prototype is developed. A controlled user experiment was conducted and it confirmed that the palm display can be usable for applications with simple layout.

When having such always available interface on the hand, another aspect we should investigate is the selection: connecting to a target device (i.e, establishing connections between the mobile interfaces and target devices). In the augmented space basically all objects are interactive, which means there are numerous selectable objects. Therefore, a decent selection technique is necessary.

There are many techniques for this selection purpose. Among existing selection techniques, pointing gestures and live video based techniques are representatives. However, both techniques can face problems. In pointing gesture, it can face occlusion problem when there are some physical barriers between users and target devices. In live video based technique, it can be problematic when there are too many selectable objects, which is the feature of the augmented space. In this situation, it requires precise pointing.

For addressing aforementioned problems, this study proposes to use a steerable camera on the ceiling. When such camera is installed at the middle of the ceiling, it can provide a view to users through mobile camera, and this view can be used for both pointing gesture or live video based techniques respectively for addressing the problems. For pointing gesture,

this view can be used as a complementary view when the occlusion occurs. For live video, this view can work as a naturally magnified view.

With the aforementioned considerations, two techniques were designed and they are called Point-Tap and Tap-Tap respectively. Also their benefits are discussed by comparing with existing techniques.

A controlled user experiment was conducted with two techniques and it found the significant effect of the users' familiarity with the space. The feature of proposed two techniques (i.e., Point-Tap and Tap-Tap) is that users should refer a location to make a selection. According to the theory of spatial cognition, there are two representations that humans memorize a location, which are egocentric and allocentric respectively. And, users can have the allocentric representation more after they become more familiar with the space, and the allocentric representation is deeply related to Tap-Tap.

Therefore, we could build a hypothesis: the users who are familiar with the space will use the Tap-Tap more efficiently. To confirm the hypothesis, a user experiment was designed and conducted. The result of the experiment found that the familiarity with the space has significant effect on Tap-Tap; the hypothesis was validated. With the result of the experiment and implications, this study proposes a guideline for selection techniques in the augmented space.

In summary, an always available interface (i.e., the palm display) in the space enabled users to access the interaction with any devices without physical burden. And, proposed two selection techniques could enable users to establish the connection between a mobile interface and a target device. The experiment with two techniques found a significant effect of the familiarity and suggested to consider it. Those findings and development of new interaction technology, technique, and their evaluation will contribute to improve the usability of target acquisition related interactions in the augmented space.

Acknowledgments

First of all, I would like to thank to my advisers Professor Jiro Tanaka and Professor Shin Takahashi. From the beginning of this doctorate course, they have given me numerous thoughtful comments and encouragements. Without their support I could not complete this thesis.

The committee faculties, Professor Akihisa Ohya, Professor Hideaki Kuzuoka, and Professor Hiroyuku Kudo, found various aspects of this study, which deserve to investigate more, and gave thoughtful comments. I deeply appreciate it.

All faculties, Professor Kazuo Misue, Professor Buntaro Shizuki, and Professor Simona Vasilache, and all students in IPLAB have always encouraged me to dig into this study more and share their valuable ideas. I thank to them all.

I also thank to my master thesis advisers, Professor Yongjoo Cho and Professor Kyoung Shin Park in Seoul. Whenever I visit Seoul, they shared delicious foods, which are very necessary source for hard working, and also encouraged me in research as well as in life.

The researchers in Microsoft Research Asia HCI team, in particular my mentor Doctor Xiang Cao, I would thank to them all. I could learn how to find research subject: having curiosity to all of what happens around us with highly motivated and positive manner. I will remember it through my life as long as I work in this research field.

My parents Bongryong Kim and Inrye Ryu in Seoul have always supported me for all aspects in my life. That support made me to take this path and complete this thesis. I love them sincerely and hope to have more time with them.

Finally, I would like to thank to my wife Nokyung Kim. Since my undergraduate hood, she has been always near to me. Every moment of being near to her is more than happy. She was, she is, and she will be my best friend, the most important person, and together.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The indoor space we are living now will become an interactive computer. Numerous studies share this expectation and have tried to prototype it [1][2][3]. In this environment, there will be lots of real and virtual devices and all of them will be interactive.

Working with such numerous devices obviously will bring many research concerns. Among numerous research subjects around it, this study focuses on in particular target acquisition related interaction. In this chapter, the detailed motivation of this study is described by clarifying the augmented space, the interaction, and the target acquisition.

1.1 Motivation

The augmented space in this study specifically means a space with many projectors and cameras, and Figure 1.1 illustrates an example. As shown in here there are many cameras and projectors, and all of objects are connected through the network; all objects can communicate with each other. This is the specific augmented space this study assumes.

With advancement of technology now we can recognize an object through image processing [4][5], the camera can work as a sensor to recognize objects. Therefore the projector is able to draw images onto the objects accordingly, and this is called the augmentation.

The augmentation enables the system to recognize an object and show some graphically generated visual contents onto the object; the augmentation can be considered a method of converting normal objects (i.e., non-interactive) to digital devices (i.e., interactive) by adding graphical content [6].

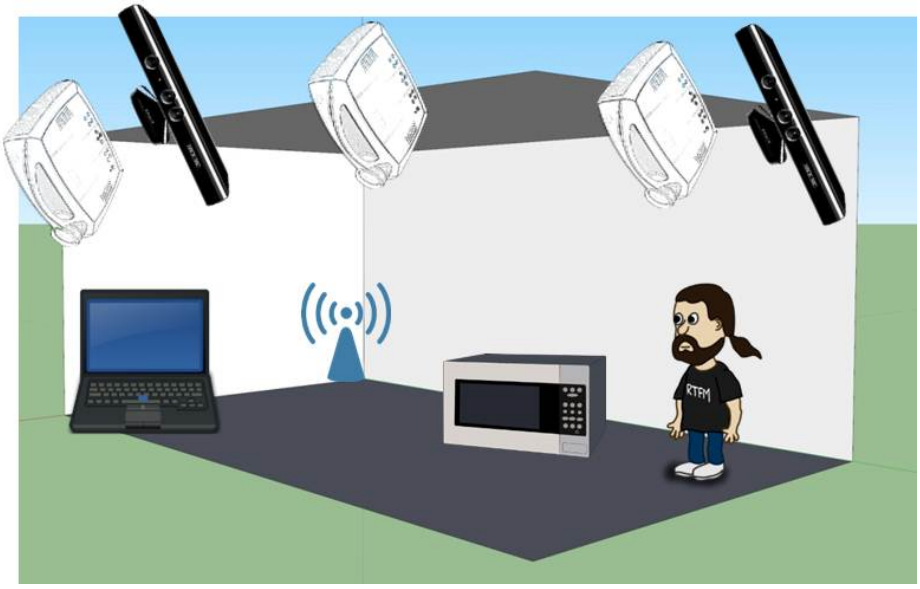

Figure 1.1: Illustration of augmented space.

In this environment (i.e., the augmented space), all real and virtual (i.e., augmented) devices are interactive; there are *much more* interactive devices than current environment. This study assumes the aforementioned augmented space and focuses on the interaction. In the next section the motivation of the interaction is clarified.

1.1.1 Interaction in Augmented Space

Before explaining the interaction in the augmented space, first *the necessity of the interaction* in future computing environment (i.e., the augmented space or ubiquitous environment) is described.

The augmentation can be considered a method of enabling ubiquitous environment, which is envisioned by Mark Weiser [7]. A basic assumption of his envision is all objects in the space become digital devices, which implies that they are under the control of computers. The ultimate goal of this ubiquitous environment is to have all of devices in the space work automatically with respect to the context so that the users do not need to interact with any devices potentially. This nice concept should be achieved and there are many studies aiming to enable it.

Indeed, if such context awareness mechanism becomes perfect in the augmented space, the interaction itself can be unnecessary. However, it is true that there is still long way to go

for achieving the perfect context awareness. And, even though the context awareness system is perfect, several researches pointed out that the users can become frustrated when they do not understand the internal mechanism of context recognition fully [8][9]. Those arguments imply that the interaction will be necessary even in the ubiquitous environment. Therefore *the researches around the interaction in the augmented space should be investigated.*

Indeed there are two types of interaction in the ubiquitous spaces (i.e., augmented space), and those are implicit and explicit interactions [10][11]. Explicit interaction includes most of common interaction in this era. For example, working with mice or keyboards for common laptops is obviously explicit interaction. In this case the system does not sense any implicit context; it just works as programmed.

Implicit interaction is relatively new. It tries to sense related context as possible and the given context is used as a consideration for future interaction. For example, if there is a less urgent notification messages (e.g., email / phone calling), it can adjust the notification timing by sensing the context (e.g., delayed notification when the user is at an important meeting) [12].

Therefore, such implicit interaction may remove the necessity of some interactions. For example, when there is an integrated controller for all devices on the user's hand, the controller may establish the connection with a device automatically by analyzing the user's heading direction or by referring to eye gaze (e.g., the user is heading to a television and his eye is also seeing it). Then, the manual selection (i.e., the establishment of the connection between the television and the integrated controller) is not necessary. However, again it is true that this implicit interaction requires relatively accurate context awareness. Again, this is still long way to go, and the applicable area (i.e., the area of gathering accurate context) is limited. Therefore, we expect that the explicit interaction will exist in the future ubiquitous spaces continuously.

The target acquisition is the subject of this study and it is a necessary step for most of explicit interactions. In the next section, the motivation of the target acquisition is described.

1.1.2 Target Acquisition

This study focuses on the interaction and it studies in particular on *the target acquisition*. Figure 1.2 shows the general flow of interactions. Generally the users need to acquire

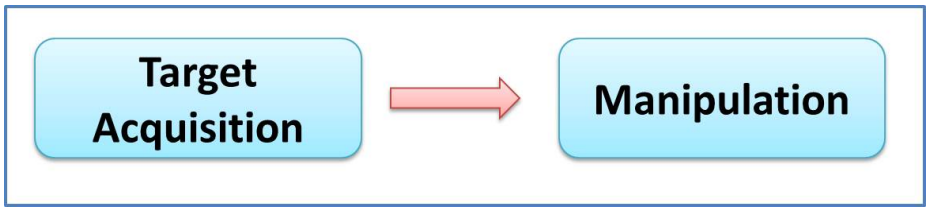

Figure 1.2: General flow of the interaction.

(access) a target device before manipulating it [13]. For example, if a user wants to change a television's channel (i.e., manipulation), he or she first needs to access its remote controller or channel control button (i.e., target acquisition).

This target acquisition is discussed in numerous studies in various domains [14][15][16] [17]. Among them, a definition of the terminology *selection* in the virtual reality study well-represents the target acquisition in the augmented space. Zhai et al. defined the selection as *the task of acquiring or identifying a particular object from the entire set of objects available* [18].

This target acquisition can be found in traditional interaction scenarios, and a remote controller can be an example. If a user wants to control a television using its remote controller, the user first needs to have the remote controller on the hand. Generally it requires some physical movement of finding or holding the controller. If the device does not support the remote controller (e.g., washing machines or microwaves), users need to access the machine by themselves; it also requires physical movement. Such physical movement of accessing to a device is an example of the target acquisition.

This physical movement may become a severe problem in the augmented space. In augmented space, basically all objects are interactive and there will be numerous devices; it requires more frequent access to the objects and it causes more physical movement. Therefore, a means of reducing such physical movement may improve the experience quality. Hence *a technology of enabling easily accessible means deserves to investigate.*

Figure 1.3 shows decomposition of the target acquisition related interactions. As shown in here there are two interactions with respect to the feature of augmented space: accessing to a device and connecting to a device.

First, *accessing to a device* is related to physical movement for having a target device itself. When reflecting the above example, having the remote controller for a television corresponds to it. Indeed, it is required for most of explicit interactions. *Connecting to*

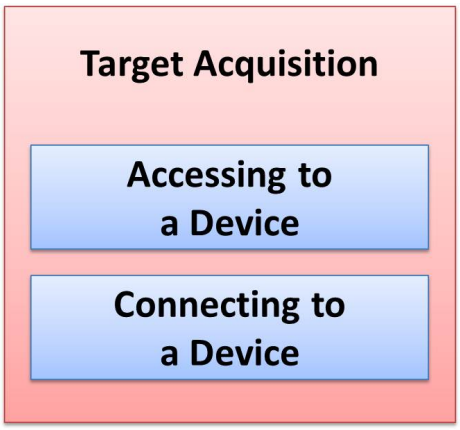

Figure 1.3: Decomposed interactions in target acquisition.

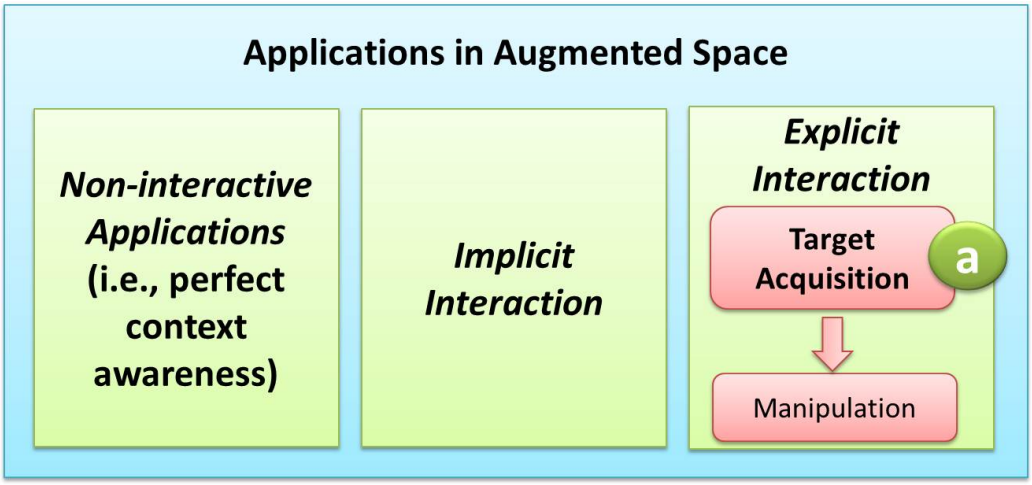

Figure 1.4: The position of *Target Acquisition* in applications in augmented space.

a device is deeply related to the feature of the augmented space; all objects in the space are connected and can communicate with each other. Therefore, if a user is holding a mobile device on the hand basically the mobile device can be used for manipulating all other device; indeed this is an integrated controller. However, before controlling a device with the integrated controller, the connection between the controller and the device should be established. This is the exact meaning of connecting to a device and it is deeply related to the above definition of selection in virtual reality by Zhai et al. [18].

1.1.3 Summary

Figure 1.4 shows the road map of the applications in the augmented space: non-interactive applications and applications with implicit and explicit interaction. Indeed non-interactive applications can be achieved through a perfect context awareness system. However, there is still a long way for having them. Therefore, the interactive applications will exist continuously. In interactive application, there will be implicit and explicit interactions. The implicit interaction can provide convenient features however it also requires the context, again which is still not easily achieved with current technology.

The explicit interaction must include two processes: target acquisition (i.e., Fig. 1.4(a)) and manipulation. This study focuses on the target acquisition. This target acquisition can be decomposed as shown in Figure 1.3. This study deals with those two subjects: accessing to a device and connecting to a device.

As a solution for those two subjects it suggests an always available interface, called *the palm display*, and two selection techniques, called *Point-Tap* and *Tap-Tap*. And, it evaluates two selection techniques. Chapter 2, 3, and 4 describe each subject in detail.

1.2 Thesis Statement

This study seeks to show that: **In augmented space, there will be many devices, and each of them will be interactive; users need to access such interactive devices more frequently. The target acquisition is a necessary step to have interaction with any devices; it must be improved through new technologies and techniques. As a technology, an always available interface that is able to connect to all devices in the space should be available and interaction techniques for establishing connections between the interface and devices also must be devised. Those interaction schemes will improve the user experience in the augmented space and they can be validated scientifically by comparing with existing work and through controlled user experiments.**

1.3 Organization of Thesis

This thesis is organized as follows. In chapter 2, it introduces the palm display system, which is an interaction technology for enabling an always available interface and related to

accessing to a device. Chapter 3 introduces two selection techniques, which are designed to overcome the limitation of common selection techniques and related to *connecting to a device*. Chapter 4 describes the result of experiment for two proposed techniques. Finally chapter 5 provides conclusions.

Chapter 2

Visual Interface on Palm

2.1 Introduction

As mentioned in the introduction, to manipulate a device users need to access the device first. Obviously physical movement to access a device may be required. The approach of reducing such physical movement in this study is to provide an always available interface. The always available interface means that users can access the interface without temporal and spatial constraint in the space thus it removes the chances of occurring physical movement. Indeed this is very obvious and easy solution to devise. However, its implementation is not simple. Therefore technologies of enabling such always available interfaces deserve to investigate and this chapter introduces a technology.

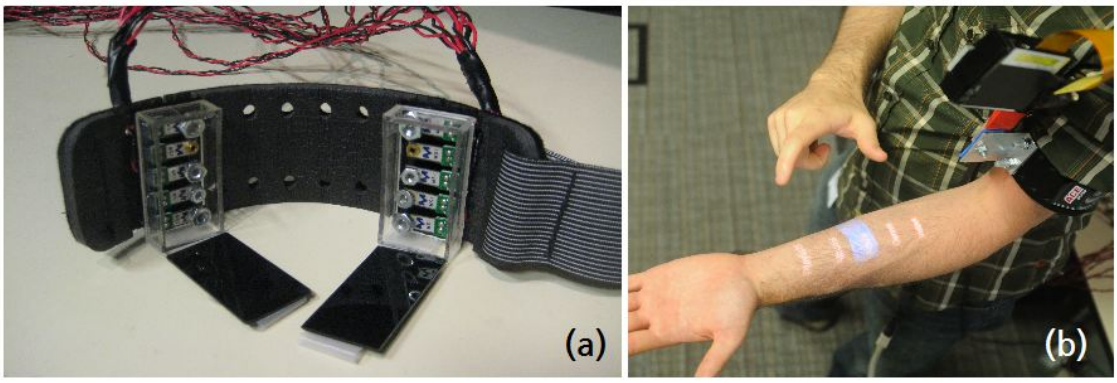

Figure 2.1: Skinput sensor prototype (a) and an example of real use (b).

Indeed, many of existing work have focused on this concern and most of wearable computers can be solutions for reducing such physical movement [19][20]. Figure 2.1 shows an example [21]. Harrison et al. prototyped an armband type system, which includes analog sound sensors and a micro projector (Fig. 2.1(a)). The band type system is attached around the elbow of a user (Fig. 2.1(b)). As shown in Figure 2.1, the system shows the images at around the forearm of the user, and it allows the touch interaction, which is commonly used metaphor in mobile devices.

This wearable computer can be always available to users without spatial and temporal limitations. However it imposes some burden, even very small, of wearing a device. A method of removing such physical burden at indoor is the projected display [4][5][22]. For example, Pinhanez installed a steerable projector on the ceiling of the space thus the projector could show images on common surfaces (e.g., walls) [23]. Therefore, it allows users to have an always available interface at indoor. When considering only indoor scenarios, projected display can be considered a promising technology to provide an always available interface without the physical burden.

Indeed projected display can be considered a technology of drawing images on a surface. For making the interface really always available, obviously an always available surface is necessary (i.e., a canvas for drawing images). We can use common desks or walls as the surfaces for this purpose. However, there is no guarantee that we can access such common surface always (e.g., at the middle of a large hall). Among many surfaces around us an easily accessible and really always available surface is the palm. Therefore if we can adapt projected display related technology to the palm, it can be an always available interface for indoor scenarios without any physical burden of holding devices. However, there has been no work of enabling such technology.

With those motivations, this study focuses to enable an always interface on palm, which should be necessarily interactive. The visual interface on the palm is designed with this concept. Through following related work section, the motivations and the benefits of approach of the palm display will be clarified.

2.2 Related Work

This related work section summarizes studies of projected displays, always available interfaces, and their related interactions.

2.2.1 Projected display

As described in the above, the technology of enabling projected display is a key of the palm display. Here first related studies of the projected display are reviewed.

Ashdown and Robinson designed a projected display system for desktop environment[24]. This system is composed of two projectors and a mirror. The first projector is installed on the ceiling, which is a general setup for showing images on the desk. The second projector is placed on the floor and the mirror on the ceiling reflects the light from the second projector (i.e., it is similar with periscopes). Consequently, the projector on the floor could cover whole range of desktop with relatively lower resolution and the projector on the ceiling could show image relatively brightly and with higher resolution. A large-sized digitizer that can cover whole desktop and an ultrasonic pen were used to support the interaction.

Molyneaux et al. designed a tangible interface using the projected display technology [4]. They enabled real-time projection onto common objects, which are mobile, and they called them the smart objects. The feature of this research is that they employed database which contains the information of external shape of the smart objects. The smart objects had common sensors inside so that they could detect simple events such as pick-up or put-down. In their research [5], they showed new type of application (i.e., sensor attached picture book). The applications in those studies showed the plausibility of the augmentation not only for surfaces but also for common objects with various shapes.

The work by Miyahara et al. is one of the very beginning researches which exploited mobile projected displays [25]. In this research they employed a common beam projector and a mobile device. The system shows the same image of being displayed on the mobile device. They attached infrared LED markers on the mobile device, and there was a computer that exists externally but communicating with the mobile device, and it could detect the LED markers; it could identify the mobile device. This computer also monitors the mobile device using a stereo camera thus it is able to detect the three dimensional posture of the mobile device. Currently, mobile projectors are commonly available, but they showed the plausible application scenarios using mobile projectors when the mobile projectors were still not easily available.

Seo et al. developed the useful interaction scheme in mobile projected display environments [26]. They focused on the mobility of small projectors and designed interaction schemes which can show large-sized image using limited-resolution display. Specifically, this

system showed a selected portion of the image at the mobile device, which is similar to spot light in darkness. They demonstrated two methods of navigating virtual space easily.

There are few studies that exploited the shadow for finger tracking with projected displays [27][28]. Echtler et al. developed a multi-touch enabled tabletop system, and it tracks the fingertip by the images which are taken from the back side of the display. They used acrylic glass, and the glass reflects light more brightly when user's finger pushes the glass. Wilson developed somewhat different system and it used only shadow information for tracking the finger and detecting the touch [28]. However, in their configuration, both screen and projector were statically fixed. Thus, they could track the shadow of finger in more stable way.

The projected display related studies can be categorized with respect to the configuration of projectors (i.e., statically fixed [29][25][26] and dynamically movable [24][30][5][23][22]). In the static configuration, user can be free from other attached device, because the projectors are installed in the space. On contrary, it is applicable for only indoor scenarios whereas the mobile projector (i.e., dynamic configuration) can be applicable at both indoor and outdoor scenarios.

2.2.2 Always Available Interfaces

The terminology *always available interface* was coined by Saponas et al., and indeed originally it was always available input [31]. They suggested a muscle-computer interface, which allows users to interact with devices using hand posture. And, the hand posture was recognized by tracking the movement of muscle, which generates different signals with respect to different postures.

When the size of tracking devices becomes small and the signal analysis technique is being more sophisticated, it potentially become always available with the very small physical burden. This is the envision of the always available interfaces using wearable computer types. Another type for the always available interface is to exploit the infrastructure in the space. Here, those two approaches are analyzed.

2.2.2.1 Wearable Computers

Interface obviously requires inputs and outputs. Most of wearable computers exploit micro projector as the output means whereas various technologies exist for the input side.

Mistry et al. demonstrated a series of plausible interaction scenarios with wearable computers [32]. The users in the scenarios wore a small web camera on the top of the head, a micro projector on the neck, and color markers at the fingertips. Because there were color markers on fingertips, the camera could track the fingertips easily and it could recognize hand gestures. Also, the camera could recognize some objects using image processing therefore the micro projector could show appropriate contents on the objects (i.e., the information of a book on its cover).

Muscle-computer interface indeed supports only input side [31]. It focused on a feature that human's muscle around the forearm generates different signals with respect to the hand posture. They attached sensors on elbow to track the signals; the sensors could detect different signals from different postures. The identification of different signals was accomplished through machine learning techniques [33].

Skinput took the similar approach but exploited different signals [21]. It attached analog sound sensors (i.e., microphones) on a user's elbow. When the user touched a point of forearm, naturally it generates some sounds and the sounds can be detected by the microphones. With respect to the touched position, it makes somewhat different sounds because the propagation paths from the point to the microphone are different. Machine learning techniques could distinguish such differences and they could demonstrate the system.

Omnitouch provided almost similar feature of the aforementioned approaches but it was implemented using shoulder-worn depth cameras [34]. As known as depth cameras can gather three dimensional data and robust on different lighting conditions. Therefore it could excellently track user's palm and fingers. A micro projector, which is also installed on the shoulder-worn package, could show some images on user's palm or walls. Then, the users could interact with the contents on a surface using hand gestures or touch metaphor.

The wearable computers indeed envision promising applications. Most of existing systems demonstrated the scenarios with common objects (augmented by micro projectors) and physical objects. Also the applications can be enabled for indoor and outdoor scenarios. When accuracy of data analysis techniques and the sensors become very small, it may work as an always available interface.

2.2.2.2 Infrastructure Dependent

When the physical size of sensors becomes and its capability is improved, the computers may be attached on the body with extremely small burden. However, there is still long way to go. An alternative is to exploit the infrastructure in the space. The projector is a good candidate to add some visuals on a surface. Indeed, commonly available projectors in these days are now reasonably bright and have enough resolution to provide the visuals even though they are installed on the ceiling. If an image is given on a surface through this way and users can interact with it, that is an always available interface without burden of holding any devices, and this is the always available interface through the infrastructure dependent way.

LightSpace by Wilson and Benko demonstrated an example of such implementation [35]. They constructed an experimental space and installed several depth cameras and projectors in the space. They synchronized the depth cameras and projectors using retro-reflection LEDs. Because depth cameras provide three dimensional data, they could manipulate a real space like a virtual space. Therefore they could recognize users and objects (e.g., common walls or tables) and could draw images on every surface including human body. In their demonstration, users could pick up content on the surface and drop the content onto other surfaces (i.e., physical metaphor).

Harrison et al. demonstrated the plausibility of this approach (i.e., using infrastructure) thorough various interaction techniques [36]. They constructed an experimental space, which is similar to LightSpace [35]. They envisioned scenarios in ubiquitous era (e.g., the space can identify and recognize all objects and users). Under such assumption, the projectors could show appropriate contents on the forearm automatically. For example, if a user finds a specific place the system shows an arrow of correct direction on the floor or on the forearm. In their setup, cameras were installed on the ceiling also; it could track the arm postures (e.g., folding arms or spreading arms). The users could interact with the space using such arm postures.

2.2.2.3 Advantages and Disadvantages of Both Approaches

Introduced two approaches (i.e., wearable computers and infrastructure dependent types) take different advantages and disadvantages respectively. Wearable computers can be applicable to both indoor and outdoor scenarios. Because users are always together with the

devices, they can use them without spatial and temporal limitations. However, users can have, even very small, some burden of holding or wearing the devices.

Infrastructure dependent type is opposite. Because it relies on the installed devices in the space, they are only applicable for the indoor scenarios. However, the users can be free from the burden of holding devices because the users do not need to wear devices.

As the title of this thesis indicates, this study focuses on the interaction in the augmented space, and it can be considered an indoor environment. Therefore, this study focuses on the infrastructure type. The palm display is developed for having the advantage of infrastructure dependent type and also minimizing the disadvantage in interaction as possible. The advantage and disadvantage in interaction will be explained in the next section.

2.2.3 Interactions in always available interfaces

Two technologies for enabling always available interfaces have been described. In here feature of interactions around two approaches will be explained.

The interactions in wearable computers are highly dependent on the characteristics of sensors. For example, muscle computers provide a set of hand postures. This is because the sensors can receive different signals according the postures. Skinput provides direct manipulation metaphor (i.e., touch), which is commonly used in mobile devices (e.g., Apple's iPhone[1]). This was possible because the sensors and analysis techniques for identifying signals could detect the points where users have touched.

The wearable computers can be optimized for especially some interested area (e.g., palm in Skinput) whereas the infrastructure dependent types can provide more various interactions engaged with the environment. For example, LightSpace demonstrated the interaction that allows users to drop a virtual object onto a physical surface. Another feature is it provides some interactions that require physically big gestures, which can be noticeable by the camera from 1-2 m distance. With respect to the limitation, LightSpace provided physical metaphor (i.e., dropping some contents onto a surface), which is big enough to be captured by depth cameras on the ceiling. The interactions in the work of Harrison et al. also demonstrated the interactions that require physical movement of arm or hand, which is also big enough to be sensed by the camera on the ceiling [36].

[1] http://www.apple.com/iphone

Table 2.1: The position of the palm display when comparing to related technologies.

	Wearable Computer	Infrastructure Dependent
Direct Manipulation (precise interaction)	Skinput [21] OmniTouch [34]	*Palm Display*
Gesture or Posture	Muscle-computer interface [31] Harrison et al. [36]	LightSpace[35]

When reflecting the interactions on both approaches, they have advantages and disadvantages together. Wearable computers can provide small and precise interactions. Skinput allowed small fingertip gestures on the small palm. Those small and fine interaction is hardly implemented in infrastructure dependent way. This is mainly because the sensors are installed in the space and there can be some distance between the region of interest and the sensors. Obviously the longer, indeed significantly longer than wearable computers, distance is hard to provide fine sensing capability. However, the sensors in the space also can be useful to track objects in the space simultaneously. Thus, the infrastructure dependent type can support the interaction with the environment more easily.

2.3 Research Goal

In related work section, the technology and interactions of related studies are summarized. Table 2.1 summarizes the interaction around in this domain. As shown in the Table, there has been no work that supports direct manipulation metaphor, even fine and small interactions, using infrastructure dependent approach. As mentioned in the above, this is mainly because the sensors are installed in the space, normally on the ceiling, it is hard to support fine interactions. The palm display tries to support it.

The goal of this study is to seek a basic hardware setting and to develop a technology for providing a direct manipulation metaphor on the palm. With this goal, the palm display is designed and implemented. From the next section the detail of the palm display is described.

2.4 The Palm Display

The purpose of the palm display is to develop a technology of enabling direct manipulation (i.e., the touch metaphor in common mobile devices) on the palm through the

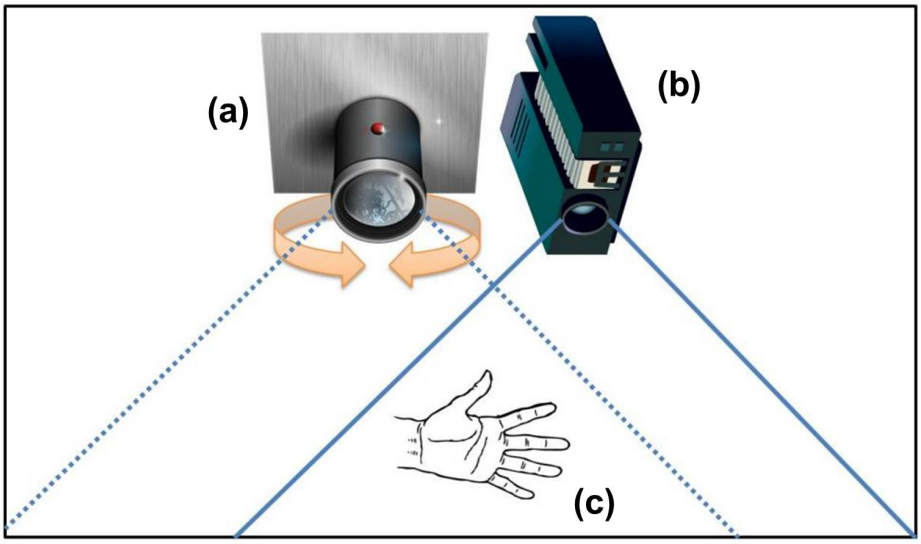

Figure 2.2: Illustration of the palm display environment.

infrastructure dependent approach. To enable direct manipulation, first the system needs to draw some images on the palm and should track the fingertip gesture on the palm. If such experience (e.g., the palm becomes a mobile screen and the finger works as a stylus pen) is enabled, then users can have an always available interface in indoor space with less steep learning curve because most of users are familiar with touch interaction on mobile device.

The palm display is designed and implemented with respect to the aforementioned purpose. This section will describe the detail of the implementation. First the environment setting, which is related to the installation of hardware, is explained.

2.4.1 Environment Setting

Figure 2.2 illustrates the environment to enable the palm display. To draw some images on palm, obviously a projector should be installed. For this purpose, a projector is installed on the ceiling (Fig. 2.2(a)). For tracking fingertip gestures on palm, image processing technique is implemented and a camera near to the projector is installed for this purpose (Fig. 2.2(b)). When a palm is between the coverage of two devices (i.e., the projector and the camera), the projector can show some images on the palm and fingertip gestures on the palm is tracked by the camera (Fig. 2.2(c)). Figure 2.3 shows it.

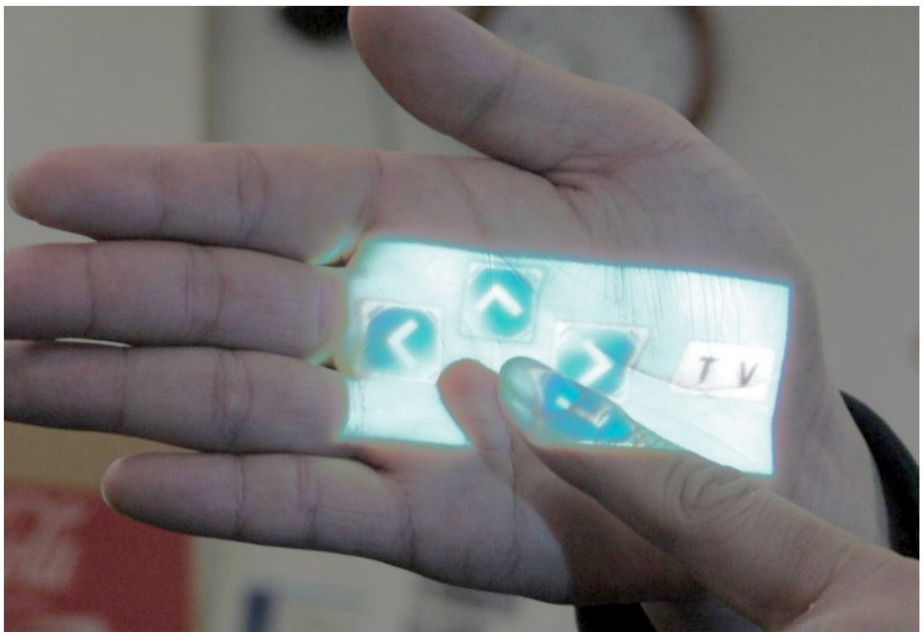

Figure 2.3: The palm display.

2.4.2 Hardware

After setting up the environment, we had an informal test to confirm the plausibility of the setting. A common projector, which provides XGA (1024 X 768) resolution, was enough to show images when the distance between the palm and the projector was about 1-2 m (Fig. 2.4(a)). However, a common web camera was not enough to provide sufficient resolution. When image is shown on the palm, the common web camera could gain the image of projected area with less than 100 pixel resolution (i.e., image projected region in Fig. 2.3). This was too small resolution to give acceptable result from the image processing.

To address this issue, a zoom and heading direction adjustable camera was exploited. Figure 2.4(b) shows it (AXIS 214 PTZ Network Camera[2]). This camera allows controlling its heading direction and zoom level through specified network access. Through the informal test, this camera could gain the region of interest with about 300 X 400 pixel resolution when its zoom level was maximized (i.e., the smallest field of view).

[2]http://www.axis.com/products/cam_214

Figure 2.4: The camera and the projector for the implementation.

2.4.3 Implementation

Now the system can show some images on the palm and the camera can gain the images with enough resolution. To enable interactions with fingertip gestures, a fine image processing to track the fingertip is necessary. Here the implementation of the image processing is described.

Figure 2.5 shows the overall architecture of the implementation. The architecture is using reactor pattern and chain of responsibility pattern for loosely coupled architecture [37][38]. The right five rectangles designate five key modules of image processing. Here first the event dispatcher is described, which is the core of architecture and having communication with all modules.

2.4.3.1 Event Dispatcher

As shown in Figure 2.5, event dispatcher has the key role in the architecture. All of modules have communication channel with the event dispatcher. In reactor pattern [37], all communication between modules is coped through the event. All modules have their own events and can accept handlers for corresponding events, and event dispatcher has a role of receiving the notification of event from modules and calling corresponding event handlers.

For example, if color filter needs to fetch the result of background subtractor in Figure 2.5, the color filter registers its handler (i.e., for background subtraction completion event) to event dispatcher. Then, whenever back ground subtractor completes its own job, the event dispatcher invokes the corresponding handler (i.e., registered by the color filter) auto-

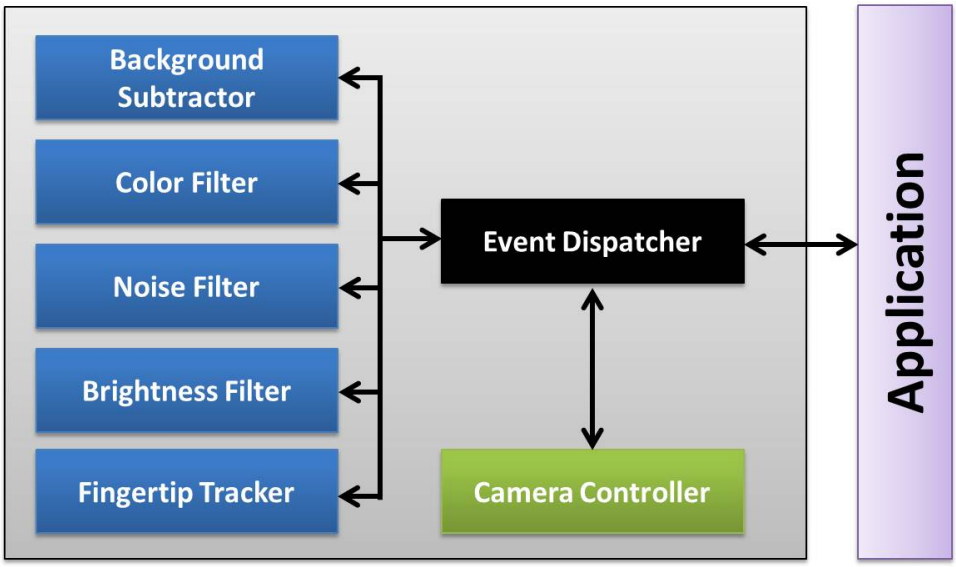

Figure 2.5: The architecture of palm display.

matically. Therefore color filter only has the information of the event in background filter but does not need to communicate with it directly, which means it minimize the coupling between modules.

The main advantage of this approach is the loosely coupled architecture; the modules do not need to be coupled even though they indeed communicate with each other directly. In this architecture, even application works as an independent module (see Fig. 2.5); the application can exist independently from the implementation of tracking algorithm. From next section, each image processing step will be described in detail.

2.4.3.2 Overview

Indeed the modules start to work in straightforward order. First, the camera controller fetches images from camera and sends it to background subtractor. Then, the image is processed from the background subtractor to fingertip tracker (i.e., stacked order in Fig. 2.5) in turn. From next section the detail of each image module is described.

2.4.3.3 Camera Controller

Before explaining the image processing, first the role of camera controller is described. The camera controller has mainly two roles, and those are capturing images and physically

controlling the camera. Commonly cameras support 25 - 30 frames per second, which means it takes about 30 ms to gather one frame. Indeed, this is long time to be ignored in computers.

To minimize such waiting time, the camera controller runs as an independent thread and continuously gathers new frames. When a new frame is available, the camera controller notifies it to the event dispatcher as described in the above. As a result, image processing related modules do not need to waste the time to wait for new frames. In empirical test, the image fetching took only 1-2 ms, which is for copying a memory block and I/O interruption.

Another role of the camera controller is to control cameras physically. Currently most cameras support the manipulation of its digital (e.g., white balance) and physical properties (e.g., direction or zoom level). There is still no standard APIs of accessing such functionaries. Applications on Microsoft Windows platform can use some related interfaces in DirectX library [39] but many of third party vendors provide their own APIs independently. Therefore, currently an independent layer to wrap those functions is required to minimize the effect for the case that a new camera is added and it provides own APIs. In our architecture, the camera controller works as a layer.

2.4.3.4 Background Subtractor

When a new frame is given from the camera controller, the first task is to find the location of the palm. For this purpose, first background subtraction technique is used. Background subtraction is simply compares the difference between two consecutive frames. For example, if identically same images are given, it eliminates all pixel values, and it can be defined as following Equation 2.1.

$$f(g(d)) = \begin{cases} 0, & \text{if } g(d) \geq threshold \\ 1, & \text{if } g(d) < threshold \end{cases} \tag{2.1}$$

In here an important factor is the threshold. Through informal test, the threshold was set 10 empirically. The function g(d) in in Equation 2.1 is defined as follows.

$$\begin{aligned} &g(d) = |pixel(n) - pixel(n-1)| \\ &\text{where } pixel(n) \text{ is } n\text{th pixel value of given image} \end{aligned} \tag{2.2}$$

Therefore, this background subtractor provides a binary image that eliminated the background.

2.4.3.5 Color Filter

Using Equation 2.1, the system is able to eliminate background. Thus when the palm is shown on the camera (see Fig 2.6(a)), the background subtractor is able to show only palm and forearm. However, the system wants to track only the palm and thus it needs to eliminate the forearm. For this purpose the system uses the technique of color filtering. The color filtering also exploits the same Equation 2.1 but g(d) in here is defined as follows.

$$\begin{aligned} &g(d) = |pixel(n) - P| \\ &\text{where } pixel(n) \text{ is } n\text{th pixel value of given image} \end{aligned} \tag{2.3}$$

The value P is user's skin color, and it is manually extracted by analyzing sample images containing user's skin.

Figure 2.6 shows the result after background subtraction and color filtering. As shown in the Figure, it extracted the palm clearly (Fig. 2.6(b)). Indeed the purpose of this process is not the fine recognition of hand. It needs to recognize the location of hand roughly. Therefore the system finds the biggest contour [40] using the image (Fig. 2.6(b)). The location is transferred to the camera controller and the camera changes its heading direction to the palm.

2.4.3.6 Focusing to the Palm

Before explaining the next image processing (i.e., noise filter in Fig. 2.5), first it explains how the system accordingly draws images and makes the camera focus on the palm. The first task is to calibrate the camera and the projector. When starting the system, the camera is set at maximized field of view and set to a certain direction. Therefore the camera and the projector can be calibrated using common techniques [41]. Consequently, when a palm is shown within the camera's coverage the projector is able to draw images accordingly.

The next step is to change camera's heading direction. We used the Axis 214 camera [3] for the prototype and it includes an embedded server that can receive messages through HTTP (Hyper Text Transfer Protocol). It provides a set of APIs (i.e., the specified format

[3] http://www.axis.com/products/cam_214

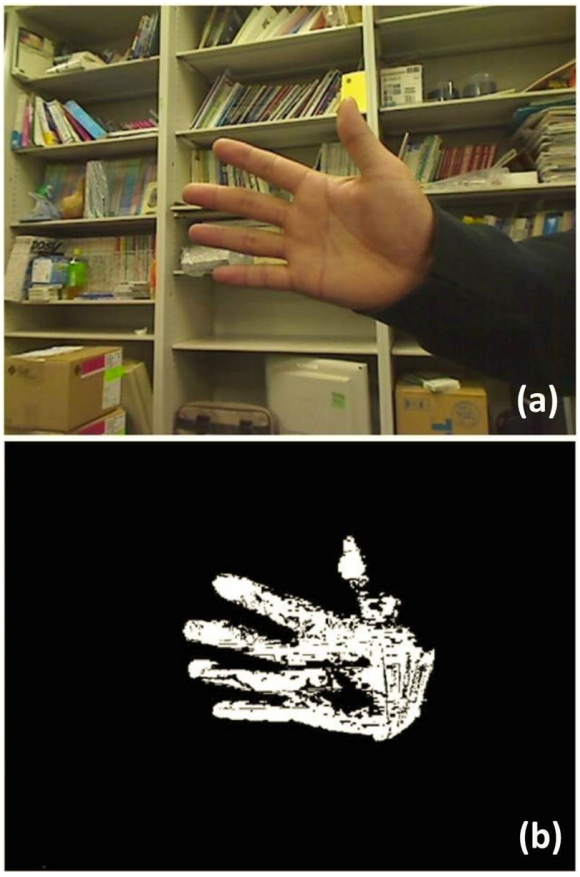

Figure 2.6: The result after background subtraction and color filtering.

of the message), and one API allows changing its heading direction. Therefore, when the palm is shown at the camera, it sends the tracked location and the camera changes its direction.

2.4.3.7 Brightness Filter

Now the system is able to have an image focusing to the palm with reasonable resolution (i.e., the camera's zoom is adjusted) through the above process. The next step is to secure the region of interest (i.e., the image projected area). For this purpose brightness filter is used. Here the detail is explained.

First the exact meaning of fingertip tracking needs to be clarified. The exact meaning of fingertip tracking in here is to find relative location of the fingertip within the area of being projected (i.e., the image projected region: Fig. 2.3(a)). The region can be considered a

Figure 2.7: Image projected region (a) on palm.

virtual screen and the relative location at the region should be tracked. Therefore, it first should track the region of interest (i.e., the image projected region).

For this purpose it exploits brightness filter. When an image is shown on the palm, the region is obviously brighter than other region (Fig. 2.7). By focusing on this feature, the brightness filter literally examines the brightness of the image. Among exiting many color spaces, HSL (Hue, Saturation, Lightness) space is a good choice because its lightness property describes the brightness well [42]. Therefore it first converts RGB colors to HSL colors and after this conversion, it also exploits Equation 2.4.

$$f(l) = \begin{cases} 0, & \text{if } l \geq threshold \\ 1, & \text{if } l < threshold \end{cases} \tag{2.4}$$

where l is lightness property of HSL image

This simple equation generates a binary image with a threshold. Thus obviously the key is to find an appropriate threshold. The detail of calculating threshold is explained in the next section.

2.4.3.8 Calculating Appropriate Threshold

When the application image is constant, it is possible to have a static threshold through empirical test. For example, we can sample several pixels at bright and dark region respectively and can find the appropriate threshold.

However, the application content is obviously variable. Therefore it is difficult to estimate (i.e., pre-determine) a threshold. Through empirical trials, we could have an initial threshold 90. In the beginning, the system uses this threshold. Using this threshold, the system counts the number of bright region (i.e., 0 values in Eq. 2.4) and adjusts the threshold. The following Equation 2.5 describes this process.

$$t = \frac{\sum_{i=1}^{n} f(i)}{n} \tag{2.5}$$

where n is the number of pixel in the region of interest and *f(i)* is defined as follows.

$$f(i) = \begin{cases} 0, & \text{if } i < threshold \\ 1, & \text{if } i \geq threshold \end{cases} \tag{2.6}$$

With respect to the result of the Equation 2.5, the system increments or decrements the threshold (i.e., adding 1 or subtracting 1). After the iteration of this process, the system confirms the threshold when t from the Equation 2.5 is more than 0.8.

After the processing by the brightness filter, the system can gain the image at Figure 2.8(a). As shown in here it clearly detected the image.

2.4.3.9 Noise Filter

Obviously the ultimate goal of this image processing is to find the location of fingertip in Figure 2.8(a). However, as shown in the image, there are lots of unnecessary portions (e.g., wrinkles on the finger). To reduce them it has noise reduction process. For a pixel, it decides whether the corresponding pixel is a noise or not through the following Equation.

$$f(g(d)) = \begin{cases} 0, & \text{if } g(d) \geq 0.3 \\ 1, & \text{if } g(d) < 0.3 \end{cases} \tag{2.7}$$

where *g(d)* is defined as follows.

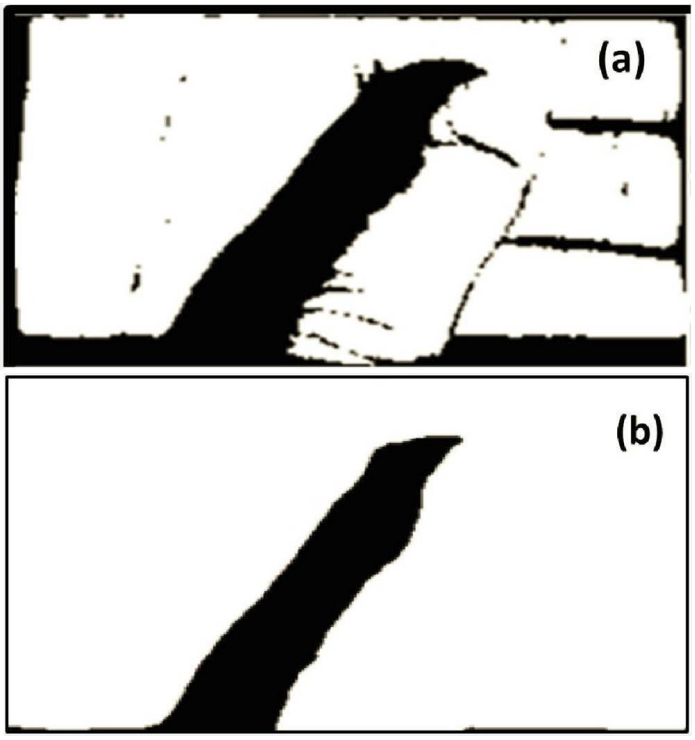

Figure 2.8: Raw image (a) and the image after noise reduction (b).

$$g(d) = \frac{\sum_{i=-n}^{n} \sum_{j=-n}^{n} p(i,j)}{n^2}$$
where $p(i,j)$ is the value of pixel of the binarized image
(i.e., Fig. 2.8a) at column i and row j. (2.8)

In the actual implementation, the n was set as 3. This value and a constant in the Equation 2.7 (i.e., 0.3) was given empirically. We processed every pixel using the Equation 2.7 and the result of this noise reduction is shown at Figure 2.8(b).

2.4.3.10 Fingertip Tracking Using Shadow

When an object is on the image projected region, it makes a shadow. As shown in Figure 2.8(b), has the noise reduced image is already available and it shows the shadow clearly. The palm display tracks the location of this shadow. The use of shadow is successfully demonstrated by several previous studies [28][29].

```
find crossing point of edge and shadow

if bottom
  scan from bottom-left point, each line from left to right
  if shadow is found
     store candidate point
  else
     stored candidate point is fingertip
if top
  scan from top-left point, each line from left to right
  ... same to bottom case ...
if left
  scan from left-top point, each line from top to bottom
  ... same to bottom case ...
if right
  scan from right-top point, each line from top to bottom
  ... same to bottom case ...
```

Figure 2.9: Pseudo code of fingertip tracking algorithm.

Figure 2.9 shows the pseudo code of the implementation and Figure 2.10 illustrates the algorithm. First, the system examines each edge side; it finds black pixels by following dash lines at Figure 2.10. Then, it can find the point of circle shown at Figure 2.10(a). If the point is at bottom as shown Figure 2.10(a), obviously it means that the fingertip is at opposite direction (i.e., the upward direction from the crossed point) because the finger is always heading to the center from an edge side. Then, the system starts to find edge of shadow from left to right (from Fig. 2.10(b) to (c)). At some point, it cannot find the edge of shadow (Fig. 2.10(d)), which means the previously found edge is the fingertip. The detailed algorithm of this process is described through the pseudo code in Figure 2.9.

2.4.4 Interaction

Through the aforementioned implementation, the system now can track the location of the fingertip. Using only this tracking capability, the system could provide an event generation method, called ***pause***.

This pause event occurs when the user stops the movement of fingertip at certain location for two seconds and specifically it can be denoted as following Equation.

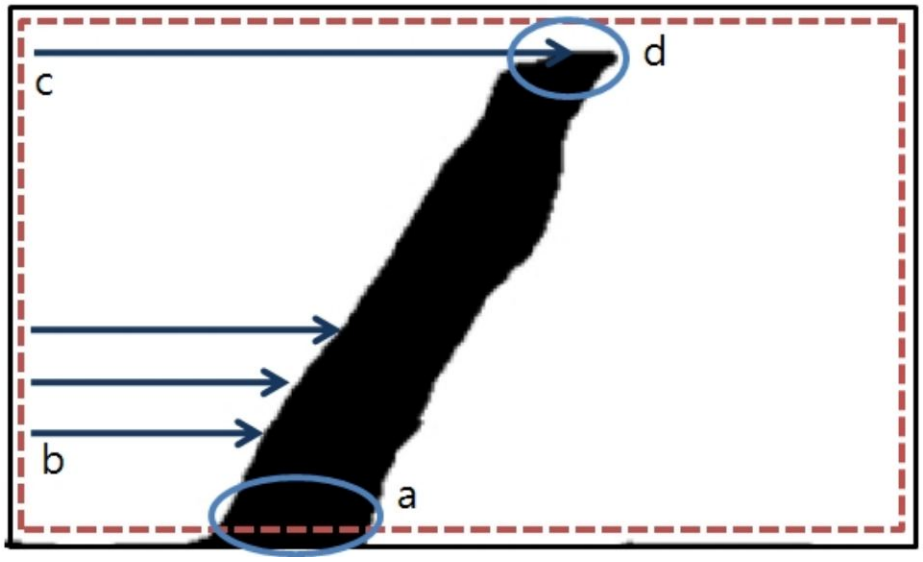

Figure 2.10: Illustration of fingertip tracking algorithm.

$$\sum_{k=0}^{n} f(p) = 0 \tag{2.9}$$

And, f(p) in the Equation 2.9 is defined as follows.

$$f(p) = \begin{cases} 0, & \text{if } |p_i - p_{i+k}| < threshold \\ 1, & \text{if } |p_i - p_{i+k}| \geq threshold \end{cases} \tag{2.10}$$

Where pi is the position of the first point in given sequence with window size n, and the size of n is determined with respect to the number of gathered points within two seconds.

2.5 Applications

2.5.1 Television Controller

Figure 2.11 shows the application of controlling a television. As shown in Figure 2.11(a), there are several buttons and each button is mapped to basic functions of common televisions (e.g., channel or volume control). When users stop the movement for two seconds at each button, corresponding command is executed.

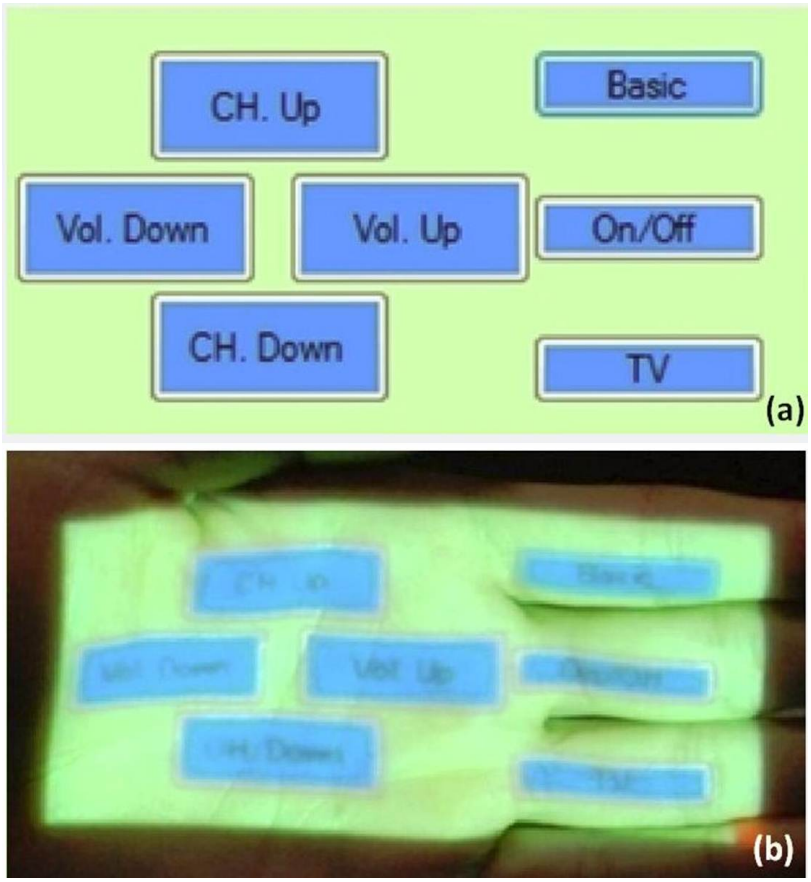

Figure 2.11: The application of controlling a television. Captured images of the application (a) and on the palm (b).

Currently most of televisions provide their own remote controllers. Also most of appliances (e.g., video players or lights) provide the remote controllers. The problem of such situation is there can be too many remote controllers. Indeed many people, in particular when visiting a relative or friend's house, suffer the difficulty of finding correct remote controller for a certain appliance. Indeed the outer shape of remote controllers for common television or video player is almost same.

Also, most people have the experience of ransacking a small remote controller in their home. Those problems can be addressed through this remote controller application. This controller can be given without any constraints in the home (e.g., living room). Also, when the environment have more context (e.g., profiles of users or temporal context), the controller can be more usefully customized; it can avoid limit the use by children at late night.

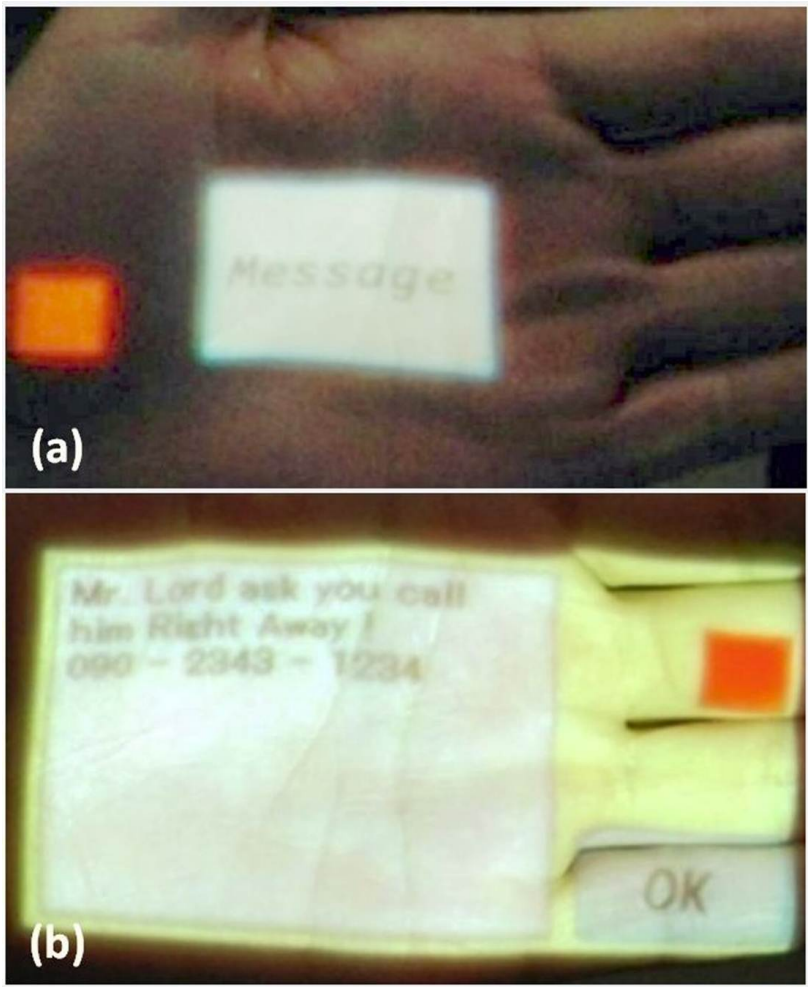

Figure 2.12: The example of message viewer. A new message is notified through the palm display (a) and the user can check the message (b).

2.5.2 Message Viewer

The controller represents the use of an interaction: when the user starts the interaction. Whereas another useful scenario of this palm display is the message viewer: the system starts the interaction. Indeed most people often forget to bring their mobile device temporarily. For example most people just put down their mobile phones on the desk when starting work or going to meeting room. And, occasionally they miss something important phone calls.

The message viewer can be used in those situations. When the system needs to notify some messages (e.g., email or short message service in common mobile phones), it shows a

Figure 2.13: Three types of applications for the experiment.

notification on the palm automatically. Figure 2.12(a) shows it. When the user makes an event on the *Message* button, then it shows its message on the palm (Fig. 2.12(b)).

This message notification service can be more useful when working with a context awareness system together. For example, frequent notification while having meetings can be rather cumbersome. Thus, some private schedule related context should be considered in this case; when such context awareness system is given, this palm display can be useful as a means of smart notification service.

2.6 Evaluation

A user experiment was conducted to test the capability of the current approach (i.e., recognition of pause method using the image processing techniques). Here the detail of experiment and its procedure are described in detail.

2.6.1 Application for the Experiment

For the experiment, one application was developed and there were three types with respect to different difficulties. Figure 2.13 shows them. As shown in the Figure, the applications have different numbers of targets (i.e., two, four, and twelve targets). Obviously more numbers of targets would be more difficult.

Figure 2.14 illustrates the use of the application with four targets example. After the application is drawn on the palm and a finger is on the application, the red colored rectangle is drawn on the fingertip for the visual feedback (Fig. 2.14(a)). When the application is ready, it changes the color of a target to yellow color (Fig. 2.14(b)). Then, the user moves her or his fingertip to the target and makes an event. Then, the application changes the color of the target to light blue (Fig. 2.14(c)).

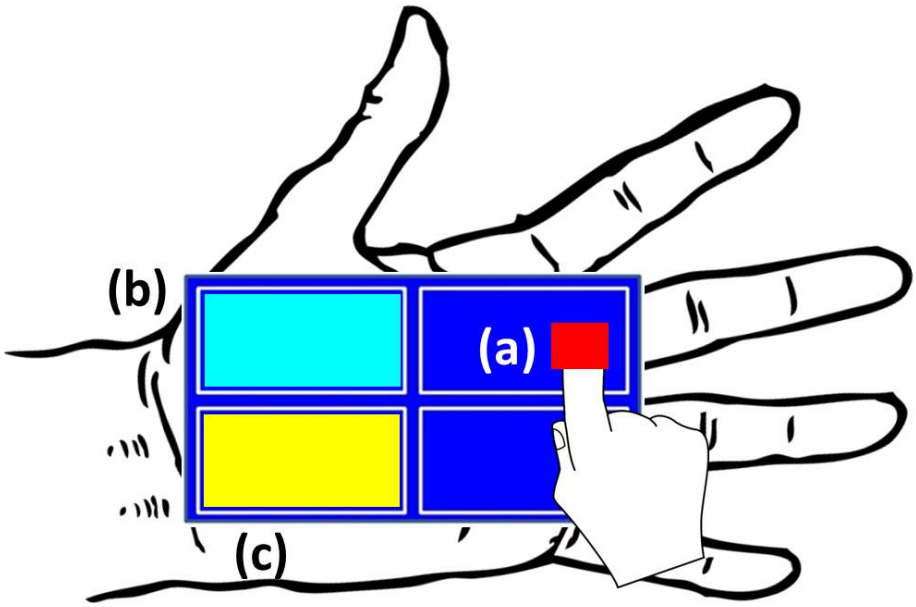

Figure 2.14: Illustration of the use of application for the experiment.

2.6.2 Users and Procedure

In total, eleven users were recruited and all of them were male and majored computer science. The age was from 24 to 31.

Before having the experiment, we explained the usage of the system for about 10 min. And, all users had one practice session. The order of the experiment was from easy to difficult levels (i.e., from two targets to twelve targets). The participants were asked to generate an event at designated target. There were 10 times of selection for each application types (i.e., two - twelve targets). The order of the target location for all users was same.

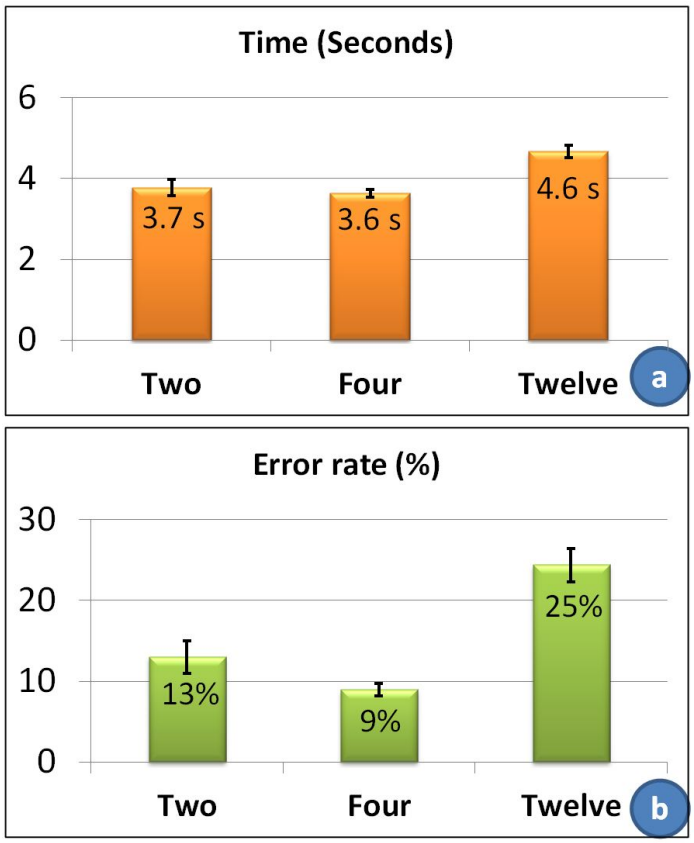

Figure 2.15: The result of the experiment.

2.6.3 Result

Figure 2.15 shows the result. The graph of *Time* (Fig. 2.15(a)) indicates the mean time of selection for a target. And, the graph of *Error rate* (Fig. 2.15(b)) is the mean error rate of failure cases (i.e., generating the event at incorrect target).

As described in the graphs, two or four targets took relatively short time (i.e., about 3.6 - 7 s) and showed about 10% of error rate. Therefore, it can be regarded as relatively good. When considering two seconds for stopping the movement (i.e., for generating the event), users took about 1.6 - 1.7 s. for generating one event.

The most difficult one took little more time and marked relatively high error rate (i.e., 25%). When considering the result of the experiment, this palm display and the *pause* method can be useful for the applications with simple layout.

When analyzing the result using one way ANOVA test between different difficulty levels, it showed significant effect between four and twelve targets. For performance (i.e., Fig.

2.15(a)), they showed $F(1, 20) = 9.29$, $p < 0.01$. Between two and twelve targets, it showed $F(1, 20) = 3.13$, $p < 0.1$. Obviously there was no significant effect between two and four targets ($F(1, 20) = 0.10$, $p = 0.74$).

It was similar when having the statistics test with error rates (i.e., Fig. 2.15(b)). There was significant effect between four and twelve targets ($F(1, 20) = 12.35$, $P < 0.01$). Between two and twelve, it $F(1, 20) = 3.97$, $p < 0.1$. And, there was no significant effect between two and four targets ($F(1, 20) = 0.82$, $p = 0.34$).

There were mainly two causes of errors: user's mistake and limitation of current implementation. First even though the experiment itself was straightforward, some of users made mistake (i.e., generating events at incorrect targets). And, currently the system tracks the location of fingertips using shadow and there is obviously spatial offset between the locations of fingers and shadow (see Fig. 2.17). With respect to such different angles between the palm and projector, the offset can become severe at optimistic situations. In this case the system could not track the location well.

2.7 Limitations and Future Work

Here limitations of current implementation of the palm display and their future solutions are discussed.

The method of *pause* can be regarded as useful for applications with simple layout. Indeed it is true that the *pause* method is not that natural. Therefore a method of enabling touch should be investigated and the shadow also can be recognizing the touch.

Wilson had demonstrated the recognition of touch using the shadow [28]. In his demonstration, the shadow of fingertip becomes very sharp when the fingertip is touched on a surface. He focused this feature and recognized the touch by examining the geometrical shape of the shadow. However, this is not applicable to the scenario of this study; the palm is movable. Instead of such geometrical approach, tracking the amount of shadow may work and Figure 2.16 shows it. In this Figure, not-touched (Fig. 2.16(a)) and touched (Fig. 2.16(b)) show significant difference in terms of the amount of shadow. Note that both images are after noise reduction. Therefore, a sophisticated method to analyze such characteristics between two different states may enable the recognition of touch.

The current implementation of fingertip tracking showed acceptable performance. However, indeed it is true that about 10% of error rate and the time of 3-4 s for selecting a target

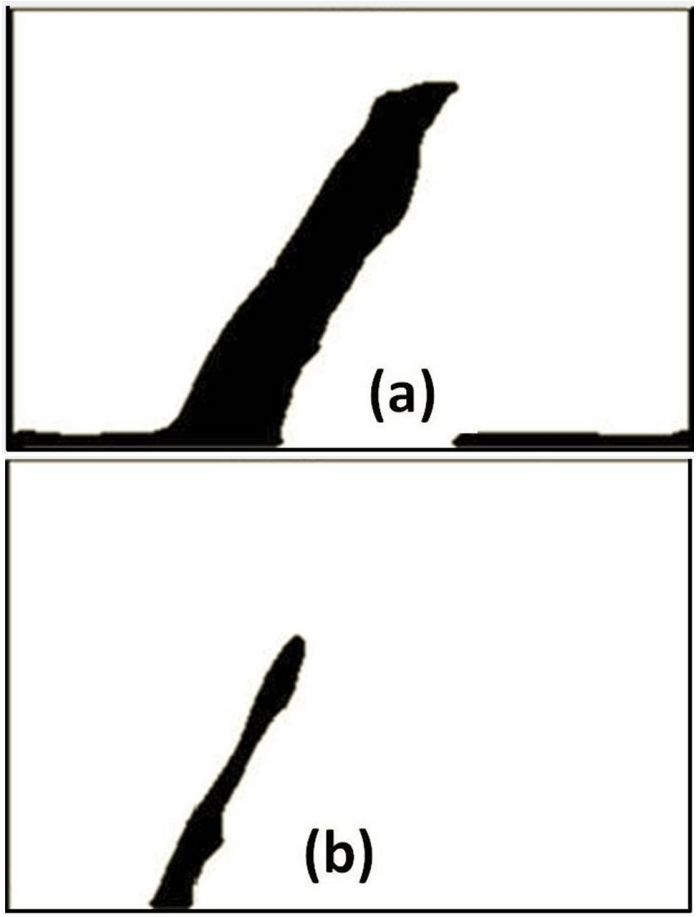

Figure 2.16: The different shadows between not touched (a) and touched (b) states.

are poorer than common touch sensitive devices. Minimizing error rate and providing faster performance should be investigated.

Currently all of image processing implementation is based on RGB image processing. Indeed, such RGB image processing can be error prone in different lighting conditions. For example, even skin color of a person can be changed with respect to different lighting conditions. Current implementation can be useful for having experiment in optimized environment, but the robustness to different lighting conditions should be addressed to have more practicality.

Another problem of current implementation is it can be error prone at different angles between the palm and the projector. Figure 2.17 illustrates it. When a projector is heading to the direction of Figure 2.17(a), the ideal location of the palm is at around Figure 2.17(a).

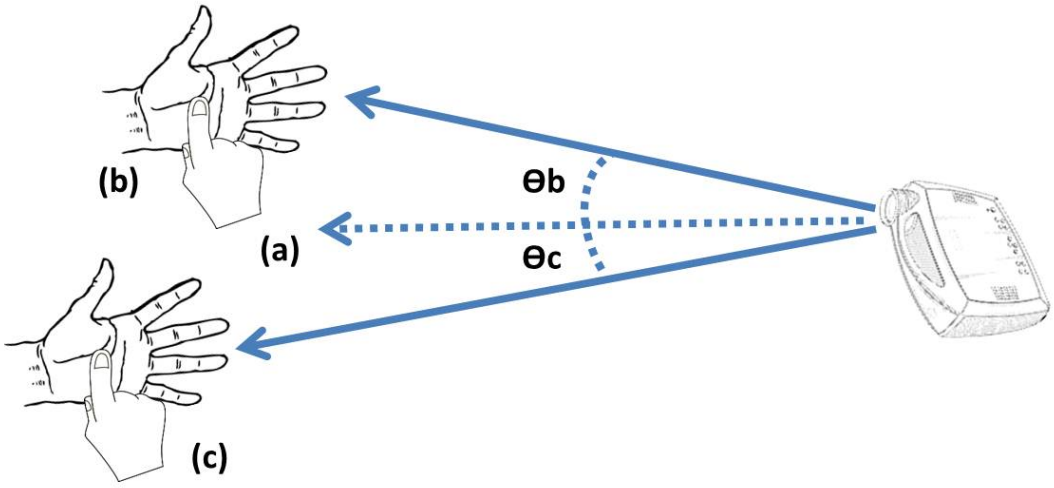

Figure 2.17: Illustration of problematic scenario.

However, if the palm is at Figure 2.17(b) or (c), the angle between the projector and the palm become θb or θc. Obviously such different angles can make different amount of shadow therefore it can be error prone when the shadow become extremely small so that the system is hard to detect it.

Currently many researchers are actively using depth cameras. Depth cameras can provide three dimensional data and robust to different light conditions. Thus using depth cameras may solve the current limitations. Using depth cameras and devising different solutions deserves to investigate.

2.8 Conclusion

There are mainly three contributions in this study. First is the novelty; it presented an always available interface on palm using infrastructure and also supporting direct manipulation. There has been no work to propose this combination. Second, it presented a concrete implementation of this approach. The image processing techniques introduced in this study can be useful for further studies. The third, it presented the result of user experiment using current implementation. The experiment clarified the limitation of current status and presented the future direction.

Table 2.1 shows the position of the palm display when comparing it with related techniques. Wearable computers that support direct manipulation (i.e., Skinput[21] or

OmniTouch[34]) can be highly accurate but also private. Indeed, it is relatively more difficult to work with the environment than infrastructure dependent type. Gesture or posture oriented interfaces on both wearable or infrastructure dependent types are hard to support fine interaction and it can require a steep learning curve. However, the palm display can have the benefit of direct manipulation, which requires less learning curve, and the advantage of infrastructure dependent type together.

Indeed the palm display can be replaced with common mobile device. However, indeed it can be considered an always available device in the space. Therefore users do not need to ransack a small device, which occurs commonly. And, it also can have more benefit in some scenarios (e.g., when user's hand is wet).

Currently, many related systems are being proposed, i.e., providing always available interfaces using various approaches. Most of those works provide their featured interaction schemes also. The introduced palm display can be considered as an effort to move a common mobile device onto the palm but without any attachment. This approach can be useful in terms of its learning curve; most of users are familiar to the usage. Therefore, it can casually work with lots of interactions in the augmented space.

Chapter 3

Selection in Augmented Space

3.1 Introduction

Now the users can have an always available interface that is given by the environment. The palm display can be used as an integrated controller for all devices in the space, and in this case the users can access to any devices without any burden of holding or ransacking the controllers. It implies that the palm display provides the easier means of accessing to devices in terms of the physical movement.

When such an integrated controller is given, a next subject we must consider is the appropriate selection techniques. As mentioned in earlier, a purpose of this study is to present an easily means of the target acquisition related interactions in the augmented space. Therefore even though users can have a platform of an always available interface, techniques of establishing the connection between the interface and a target device should be devised.

With this concern the terminology selection in here exactly means that users designate a target device so that the system can establish the connection between an interface and a target device. This is a necessary technique to complete the easier means of accessing to a device in the augmented space because basically all of objects in the augmented space are interactive.

Indeed, there are already many selection techniques around this concern. For example, in a common desktop computing environment, it generally shows a list of connected computers and users can select a computer (see Fig. 3.1). This WIMP (Windows, Icon, Menu, and Pointing) based interface can be used within a mobile interface. Then users can

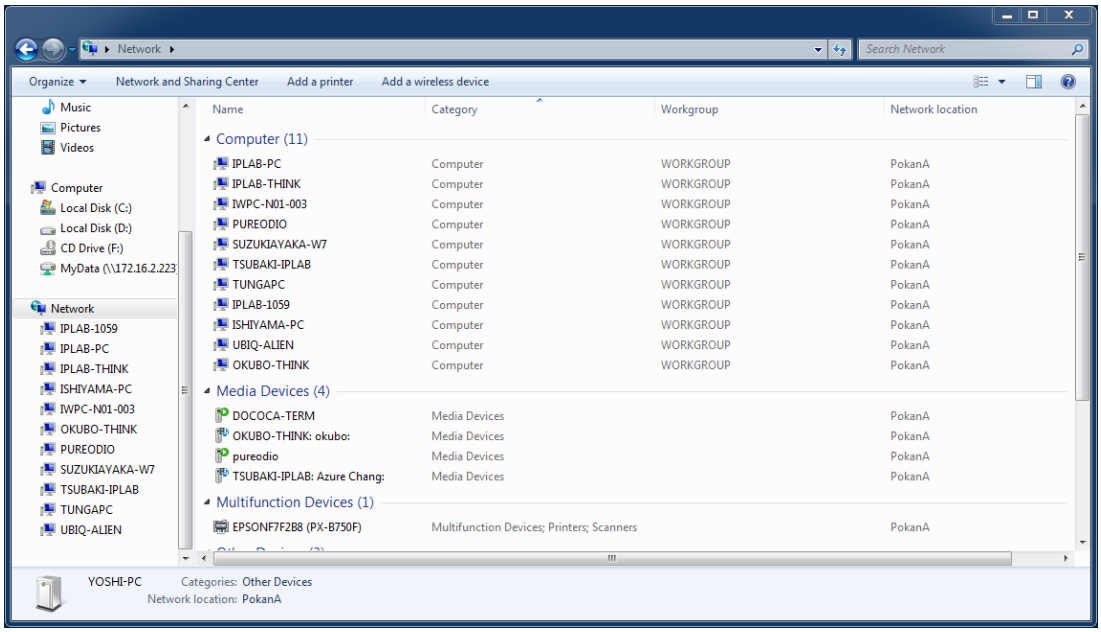

Figure 3.1: Example of enumeration of devices in common GUIs (Microsoft Windows 7 Interface).

select one object through this interface. However this technique is not well suited in the augmented space.

A feature of augmented space is there are numerous devices and all of them are interactive. In this scenario, the traditional method (i.e., GUI or CUI) must enumerate a long list of connected device. It means users should memorize the identifier of devices; obviously it is difficult. For example, if all lights or displays in a laboratory or office are interactive and users need to select one of them, in this case the traditional interface must enumerate at least more than a few dozens of names.

Moreover, there can be some implausible scenarios. For example, if there is a display next to a user and she or he wants to select it. Obviously the user knows which display she or he wants to select and can see the object by own eyes. But because the user does not remember its identifier, it is hard to select it through the traditional interface. Indeed it happens commonly in working area. When considering such scenario and limitation, the traditional method is not appropriate for the augmented space.

Among existing many selection techniques, the techniques that share the concept of *see-and-select* are likely appropriate in the augmented space. The terminology see-and-select

Table 3.1: Selection techniques that rely on spatial locations and identifiers

Spatial Location	Identifiers
Pointing Gesture	Command line interfaces
Map with live video	Graphical user interfaces
	Speech recognition

in this study is defined as *the selection that can occur while users are seeing the objects with their own eyes.*

Indeed selecting an object means that users distinguish a characteristic of an object from others. For example, common GUIs require users distinguishing objects with their identifiers. On contrast, the selection techniques that share the concept of see-and-select are different; they rely on spatial locations. When in particular there are many similar objects, users can distinguish an object with their locations (e.g., normally people designate a light on the ceiling by describing its location: the left one in the middle). Table 3.1 summarizes those techniques.

Pointing gestures and map with live video are representative techniques of the see-and-select and Figure 3.2 illustrates them. Pointing gesture is generally considered one of intuitive selection techniques. Users can select an object by designating with hand or a device while seeing them directly (Fig. 3.2(a)). Live video is similar but it uses an image that covers whole range of space. For example, when installing a camera with wide view angle lenses it can show whole range of the space, users can select an object by touching the location of the object. In Figure 3.2(b), it touches an object around the location at 3.2(b)1.

When reflecting the aforementioned scenario (i.e., selecting a light or a display), the user can select the display by designating using pointing gesture or by touching the screen; users do not need to remember their identifiers. Indeed it is much easier than traditional GUI or CUI based techniques.

However, both techniques can be problematic in specific situations. For pointing gesture, it can face occlusion problems. Indeed, pointing gesture technique can be valid only when the users can see objects with own eyes directly; obviously users are hard to select an object if they cannot see it directly. Indeed, such occlusion problem of pointing based techniques has been pointed out in several related studies [43][44][45].

Another problem of pointing gesture is its accuracy is not high. In virtual reality researchers evaluated this pointing based techniques and most of researches concluded that

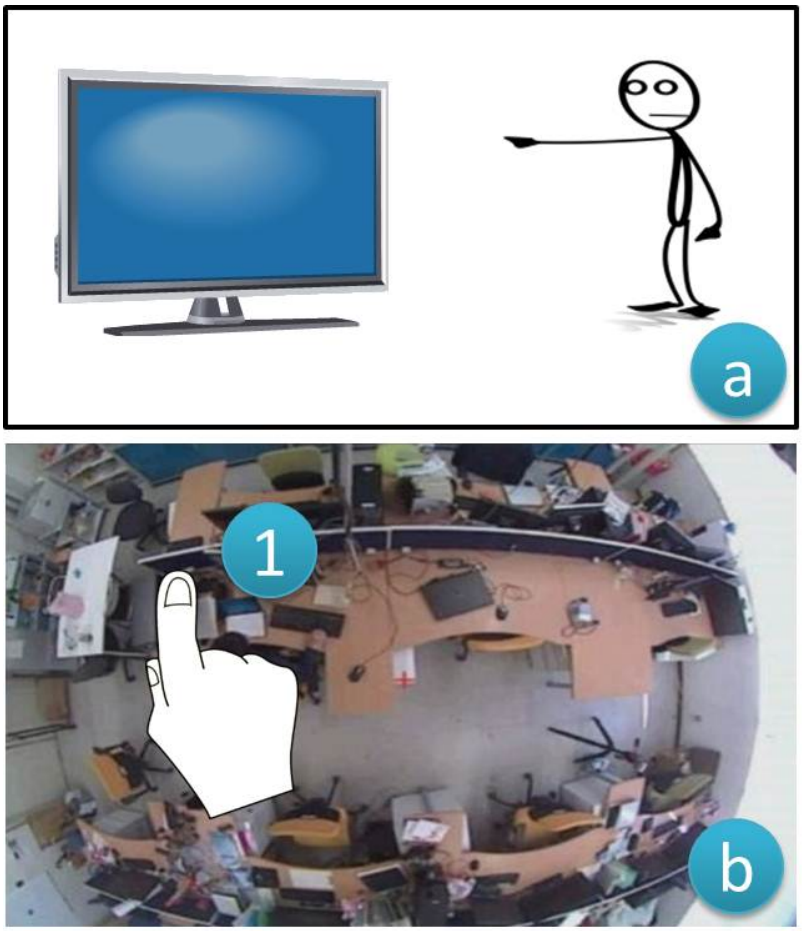

Figure 3.2: Illustration of selection techniques using pointing gesture (a) and map with live video.

pointing gesture is very powerful technique when there are not many object but it can be problematic when the high accuracy is necessary [14][46][47][48].

Generally more objects are shown in a mobile screen, it requires more precise pointing [49][50][51][52]. Live video based technique can have problems when the video is shown at small devices (e.g., mobile screen) and there are many selectable objects. When the density (i.e., the number of selectable objects) is high, obviously it requires the pointing with high fidelity. Thus, users may have some difficulty on it.

To address such problems two techniques are developed, which are called *Point-Tap* and *Tap-Tap* respectively. In this chapter, two techniques are introduced, and it validates their concept by comparing to related techniques.

3.2 Related Work

Here first it summarizes related work and clarifies why two techniques (i.e., pointing gesture and live video based) are selected as the subjects.

3.2.1 Pointing Gesture

Users can select an object naturally with a pointing gesture if there is no barrier between the users and the object. Some psychology studies examined the pointing gesture with children, and they found that even 1 - 2 years old children can use the pointing gesture accurately [53][54][55]. Therefore it is considered a powerful and intuitive technique and has been exploited in various domains such as large displays, virtual reality, and ubiquitous environment. Usually on those environment, there is some object that is out of users' reach; in this case the pointing gesture can be used effectively.

Users normally are hard to cover all areas in physically large display. The work by Bolt is one of early work of using pointing gesture in large display environment [56]. In its demonstration, users could move an object in large screen with pointing gesture and speech based commands. Vogel and Balakrishnan intensively studied the effect of pointing based techniques in large display environment [57]. They concluded that ray casting based technique (i.e., general pointing gesture) was the fastest among other related techniques (e.g., eye gaze or indirect pointing). However, the pointing gesture showed high error rate, which is a generally referred problem.

In virtual reality, pointing gesture was extended to various forms with respect to application scenarios [46]. A feature of virtual environment is it is fully graphically generated world; rich graphical representations of pointing ray can be given. Using such rich graphics, virtual reality provided 3D pointing or flexible pointing [45]. Indeed pointing gesture in real environment is 2D interaction. We can only designate X and Y coordinate. However, in virtual environment it also can manipulate the length of pointing ray; it is 3D pointing. In 2D and 3D pointing, a common problem is the occlusion. To avoid such occlusion problems, it provided flexible pointing, which can bend the pointing ray.

In ubiquitous environment, pointing gesture was used for allowing users to designate a target object. For example, Kemp et al. used the pointing gesture for designating an object so that it makes a robot to move to the object [58]. Wilson and Shafer prototyped a special wand, which can track its orientation and location [59]. Indeed the wand can work

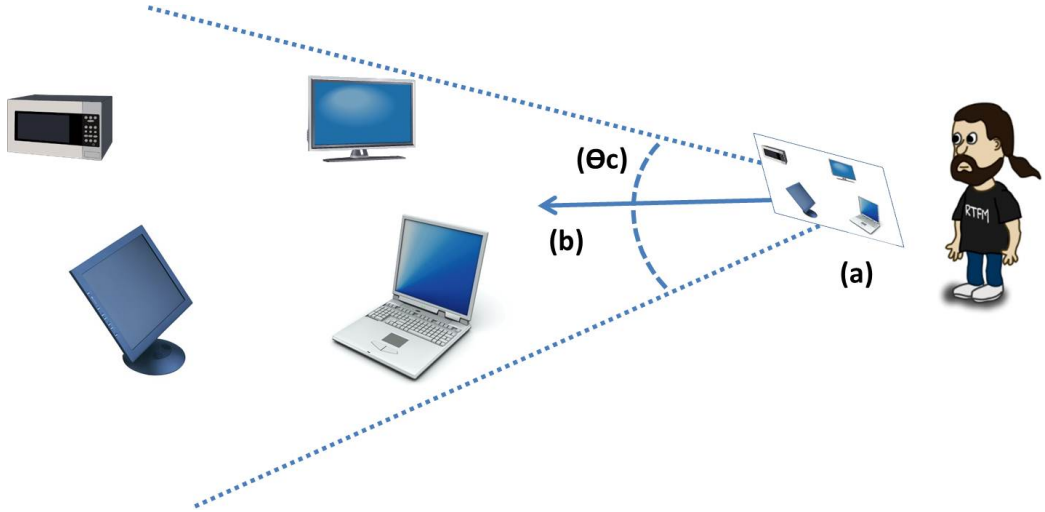

Figure 3.3: Illustration of mobile AR based technique.

like a virtual wand in virtual environment. Therefore the users could designate any object in the space using the wand. However, it also could face occlusion problems.

Wilson and Pham noticed this problem and they provided a modified version, which is called WorldCursor [60]. In this system, they installed a steerable laser pointer on the ceiling. When a user is designating a point, the steerable laser pointer provides the visual feedback on the designated point (i.e., light dot). Therefore it could resolve the problems of occlusion. However, indeed it still does not solve the occlusion problem perfectly; it does not provide the view of the hidden area to users.

3.2.2 Mobile Augmented Reality (AR)

The principle of mobile AR is simple. First the posture of a mobile device is fully trackable in the space (i.e., the space can recognize its location and orientation). When a mobile device has a front camera, using such geometrical properties (i.e., location and orientation) and camera lens' specification (i.e., field of view), the system can recognize the object on the screen. Figure 3.3 illustrates this mobile AR technique. When Figure 3.3(a) is the mobile device and its orientation (i.e., Fig. 3.3(b)) and filed of view (i.e., 3.3(θc)) are known, it is easy to recognize that which object is drawn on which location on the mobile screen (Fig. 3.3(a)). If such tracking capability is not given, generally mobile device processes the images from the front camera. Most of mobile AR techniques are using one of either approaches [61][62][63].

The mobile AR system can be considered one type of pointing-based selection from an interaction-centric perspective. Indeed, mobile AR also relies on the direction from the device to target objects and the device is always near to users (Fig. 3.3(b)).

3.2.3 Map with Live Video

As shown in Figure 3.2(b), it shows a digital image that covers the whole range of a space on a digital surface. Then, users can select one of objects through that surface. This map based technique have used on both mobile and static displays.

CRISTAL installed a camera with a wide view angle lens on the ceiling and shows images on large sized table top [64]. In here users could select an object on the table and could move the objects on the other appropriate devices (e.g., dropping a DVD title onto a DVD player).

Sketch and Run provided similar features on mid-size mobile device (i.e., tablet PC like size) [65]. They set up a similar environment of CRISTAL and proposed a technique that allows users to generate path based (i.e., sketch) commands for robot. Users could draw a curve based line on the mobile device, and a robot follows the path.

3.2.4 Proximity-based Techniques

Another area of enabling see-and-select is proximity based technique [66]. With this technique, the selection occurs when the distance between two devices become shorter than a certain threshold. Swiping RFID (Radio Frequency IDentification) [67] on public transportation system is an example. This technique can be implemented in relatively cheap cost with high fidelity.

3.2.5 Summary

In related work section, four domains of technique were introduced: pointing gesture, mobile AR, map with live video, and proximity-based. Indeed, mobile AR related approach can be considered one of pointing gesture based techniques. This is because mobile AR also relies on the direction from users to target devices. Only difference is it requires a device. Therefore in interaction-centric viewpoint, it can be included in pointing based technique.

First three techniques (i.e., pointing gesture, mobile AR, and map based) are able to support remote selection. Indeed, this is one of critical requirement in mid to large size

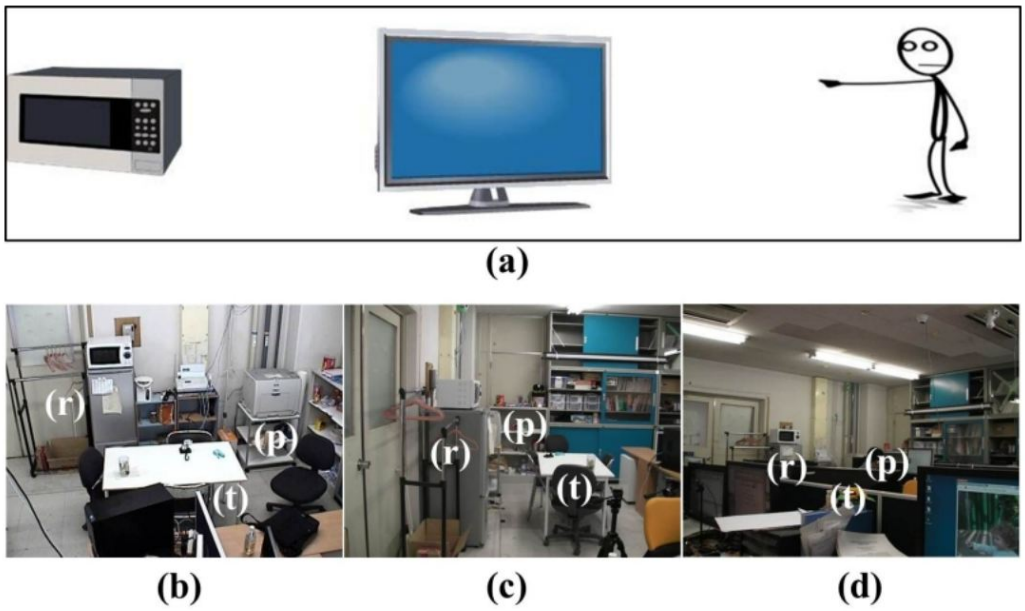

Figure 3.4: Illustration of occlusion problem of pointing gesture.

environment. However, proximity based technique does not provide this feature. Therefore, this study deals with pointing and map based techniques.

3.3 Point-Tap and Tap-Tap

Here it introduces two proposed techniques, which are called Point-Tap and Tap-Tap respectively. First, the detail of problems is described.

3.3.1 Problem of Pointing Gesture

Figure 3.4 shows the problem in detail. In Figure 3.4(a) the user is difficult to designate a microwave because a big television occludes it. Figure 3.4(b)-(d) illustrates the problem in real environment. Figure 3.4(b)-(d) were taken for a same place from different locations in the space, and labels (r), (p), and (t) denote refrigerator, printer, and table. In Figure 3.4(b), three objects are shown well, but in Figure 3.4(c) or (d) they are occluded by each other or other objects (i.e., partition and displays in Fig. 3.4(d)).

It implies that users may have difficulty on designating an object accurately because the users cannot see the object well. Indeed such occlusion problem can be addressed

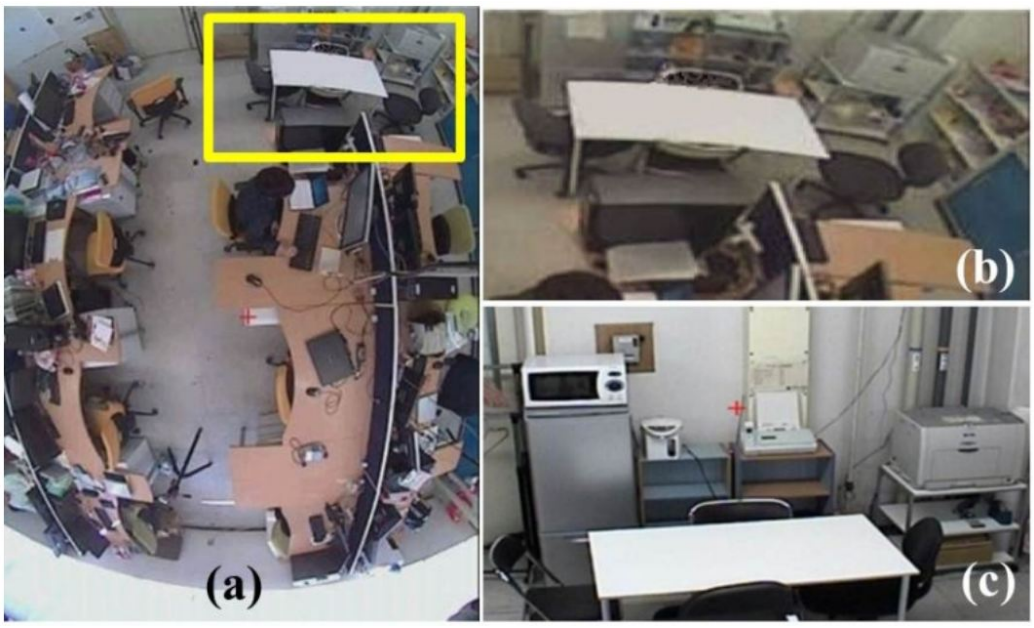

Figure 3.5: Problem of naïve magnification.

in virtual environment using flexible pointing. However, in real environment such rich graphical representations are not available.

Wilson and Pham presented the solution using a steerable laser pointer on the ceiling [68]. Indeed it can avoid the occlusion problem however, it cannot provide the view of the hidden region. As mentioned in the above, if users are difficult to see objects directly, there is no means of designating an object precisely.

3.3.2 Problem of Map with Live Video

The problem of live video based technique occurs when there are lots of selectable objects. When there is a camera and it can capture whole range of a space (Fig. 3.5(a)) but the image is shown on small mobile device, it may require precise pointing. In Figure 3.5(a), there are many objects (e.g., displays and computers). When such image is given on small mobile device, users may have difficulty on picking up an object (i.e., precise pointing is required).

Obviously, it can provide a naïve magnification technique but the magnification also has some problems. Figure 3.5(b) shows the magnification of the yellow rectangle in Figure 3.5(a). To cover whole range of a space it generally requires a wide view angle lens. And the

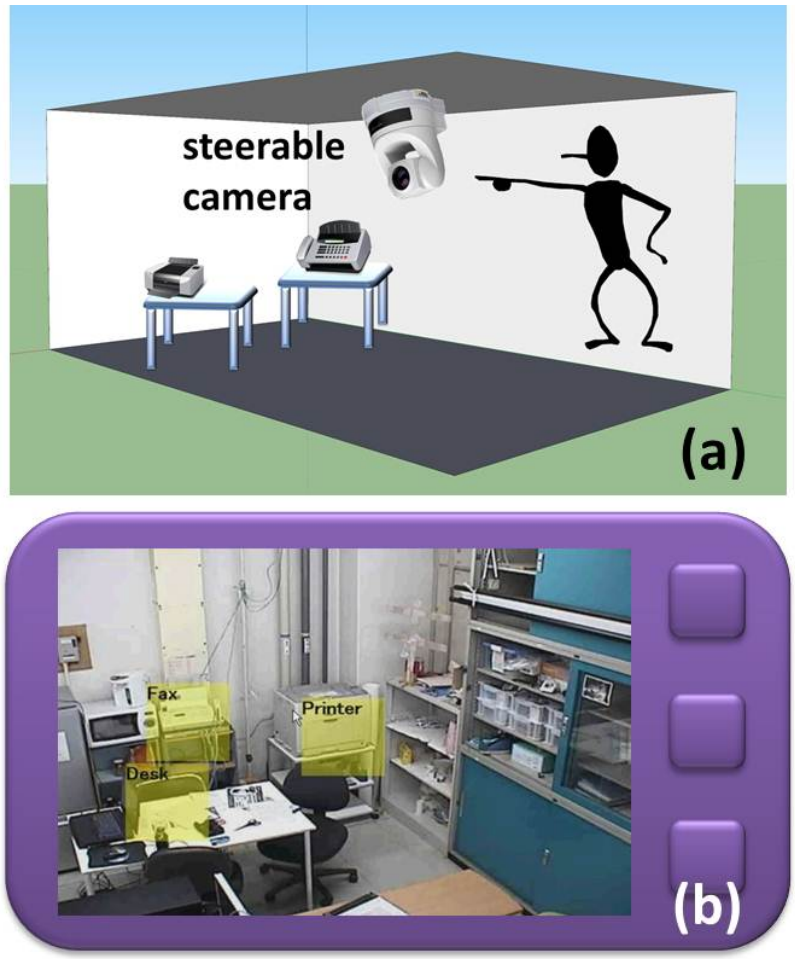

Figure 3.6: Illustration of Point-Tap.

images from such lens are distorted severely in particular at around the edges. Thus Figure 3.5(b) provides the unnatural view. Also, with respect to the capability (i.e., the field of view) of lens, some objects around the edge might not be seen completely (see 3.5(b)). Therefore there should a solution to provide natural view with broader coverage.

Figure 3.5(c) shows a clear view of the same place of Figure 3.5(b) and it was given by a normal camera. As shown in here, all objects are shown completely with more natural view. If there is a means of providing such natural view, it will benefit to use the live video based techniques more efficiently.

3.3.3 Point-Tap

Point-Tap is a technique that is designed to solve the problem of pointing gesture. It exploits an additional view, and the additional view is given by a steerable camera on the ceiling. Figure 3.6 illustrates it. The steerable camera is installed on the ceiling and it can change its heading direction as its name literally indicates.

Generally occlusion problem occurs when there is some objects that hides target objects from users' view. However, the steerable camera is installed on the middle of the ceiling, which is a good location to overcome the obstacle objects. In next section, it demonstrates how the Point-Tap is used by a user.

For the interaction, first a user is holding a mobile device. Then the user designates a place where the target object is around. After making pointing gesture, he taps the screen of the mobile device (Fig. 3.6(a)). This tapping action confirms the pointing direction, and the direction is sent to the system.

Then the system makes the steerable camera change the direction to the designated point and sends back the image to the mobile device. Then the system draws transparent rectangles on selectable objects with their names as shown in Figure 3.6(b) (i.e., desk, fax, and printer). This is for helping users the selection. Then the user is able to complete the selection by picking up one of rectangles.

Because this technique requires a pointing gesture and a tapping (i.e., touch), it is called Point-Tap.

3.3.4 Tap-Tap

Figure 3.7 illustrates Tap-Tap technique. This technique also exploits a steerable camera on the ceiling. In the beginning the mobile device shows 3.7(a). In here users do not need to make precise tap on a target object. Users can roughly tap the mobile device where a target object is. The system recognizes the tapped point and finds the nearest selectable object. It causes the camera change its direction to the nearest object. Then the image from the camera is sent to the mobile device.

Indeed, the image in Figure 3.7(b) is same to Figure 3.6(b). The remainder of the interaction is identically same to Point-Tap; the users can tap one of rectangles and can complete the selection.

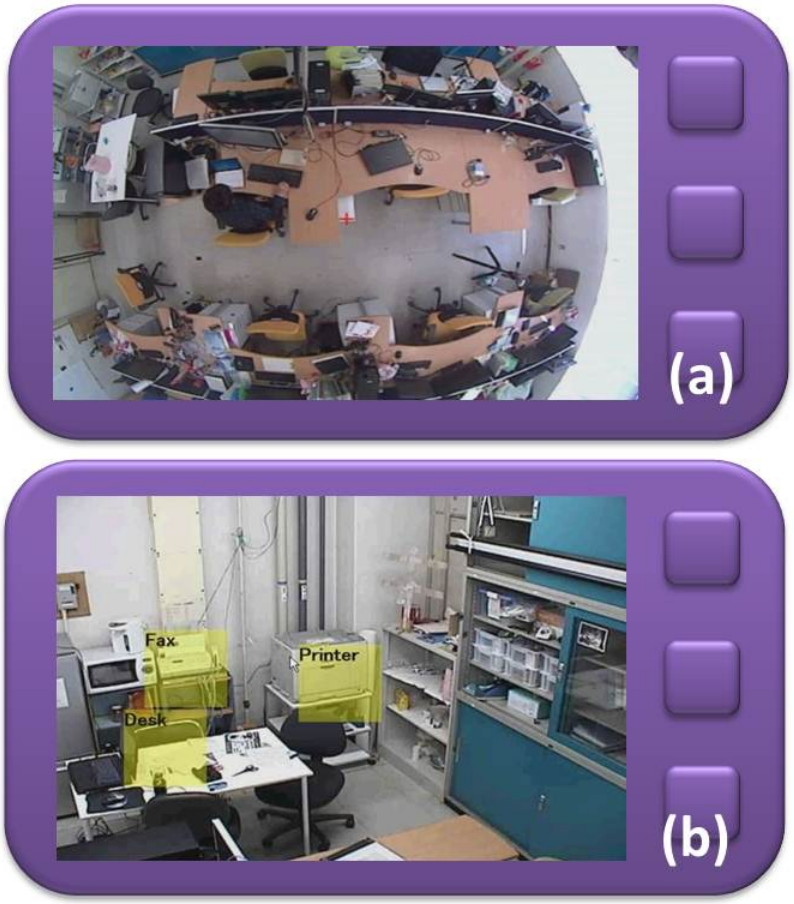

Figure 3.7: Illustration of Tap-Tap.

This technique requires two times tapping of the screen. This is why this technique is called Tap-Tap.

3.3.5 Discussion of Two Techniques

So far proposed two techniques have been introduced. Here some expected issues around the two techniques are discussed, and it explains the benefits that would be gained through the featured approach (i.e., exploiting a steerable camera on the ceiling).

3.3.5.1 Can Users Designate Hidden Objects well?

The Point-Tap technique is designed to enable users to select objects even if the objects are not directly shown to the users. Indeed it is valid under an assumption that humans can

points out a location roughly even though the object is not shown well. Therefore, it raises the question as to whether the users can designate the location of hidden objects well.

An experiment by Berkinblit et al. confirmed this ability in humans well [69]. In their experiment, users were asked to make pointing gestures to hidden objects with or without vision. When the user had the test without vision, they made 5° angular errors at most in azimuth or elevation.

This angular error corresponds to about a 43 cm when the distance between the user and the object is about 5 m; it is fairly accurate. This experiment implies that humans are hard to make highly accurate designation when they cannot see targets by eyes but rough designation is possible within the aforementioned error.

Indeed, Point-Tap does not require highly accurate pointing gesture. Even though the first pointing gesture is not highly accurate the users can have one more chance of confirming the selection (i.e., tapping one rectangle). It is expected that the small error (i.e., about 5°) will be reasonably accurate to use Point-Tap.

3.3.5.2 Live Video versus Rendered Map

Tap-Tap exploits live video as a view source. The users select an object through the images of the video camera. Indeed it is true that a well-designed map might be more understandable than a raw video in some scenarios. Therefore a rendered map image may be used instead of an interface with live video. However there are mainly two benefits that a live video can have over the rendered map images.

The first benefit is the feedback. The live video can provide feedback immediately if the feedback is visually noticeable. For example, when interacting with lights or displays, they reflect the result of the manipulation soon (i.e., turning on lights or changing contents of a display). The visual feedback can be given through the video in real-time. But a map cannot provide the such feedback.

The second is the cost. Indeed the map must be produced through somewhat artificial way and it naturally requires at least small cost. On contrary, the live video can be given without additional cost if there is a video camera. Also the cost of map becomes higher when the layout of the space is changed more often. But the live video dos not have any concern of the layout change in terms of the cost.

3.3.5.3 Occlusion-Free

By exploiting a steerable camera on the ceiling, the system has a lower probability of facing occlusion problems. The occlusion occurs when there are physical barriers between users and objects. In this situation, if the system can use other view sources, based on different locations, it has a higher probability of avoiding the occlusion. Also, the steerable camera is installed on the middle of the ceiling, which is generally the best spot that can avoid the occlusion.

One issue related to this feature is the camera shows an image based on its location, which can be different from the view perceived by the users. Therefore, users may have disorientation from such view sources based on different locations. Chan et al. and Rohs et al. conducted a series of experiments related to this issue [70][71]. In the experiments in both papers, the users were able to overcome the different views.

3.3.5.4 Less Complexity in Object Recognition

Most of augmented reality based system requires image processing techniques for recognizing objects; sometimes unnatural markers are necessary [71][72][73]. Indeed such unnatural markers can break the imersiveness. One way around this is to try to recognize objects through markerless approach [74][75][76], but this is not easy to implement in terms of image processing and is prone to errors.

However, when using fixed and steerable cameras, it does need to have such complex image processing. Because the location, orientation, and field of view of the camera are known, the system can detect objects by referring to geometrical properties rather than relying on image processing.

Obviously, when an interactive object is moving, some other methods of object recognition are still necessary. However, there are many scenarios that deal with statically fixed objects (e.g., large displays, lights, and common desks).

3.3.5.5 Hand Jitter-Free

Another advantage of the view from a fixed camera is that it can be free from the hand jitter problem. When users rely primarily on the view from a mobile device's own camera, hand jitter can cause the image to be unstable. Such systems need to provide methods for

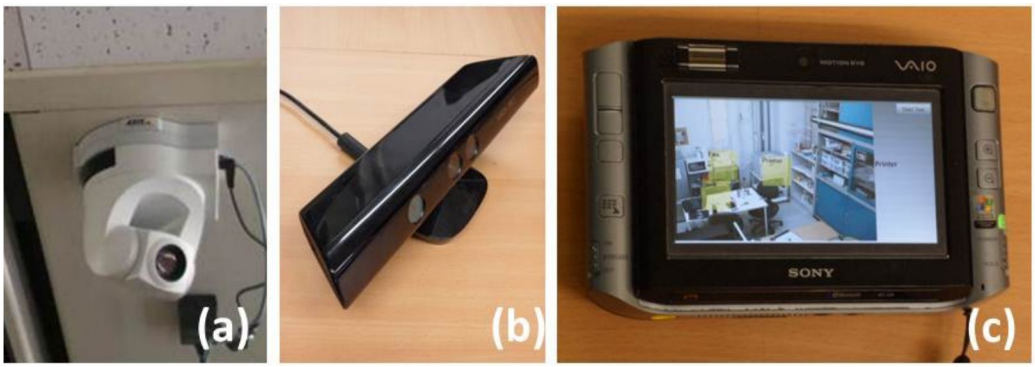

Figure 3.8: Hardware for the implementation.

overcoming the jitter (i.e., stabilizing the image) [77][78]. In contrast, with our setup, such stabilization is not required, because the steerable camera is fixed on the ceiling.

3.4 Implementation

Here the detail of the implementation for prototype is described.

3.4.1 Hardware

There were mainly three hardware for implementing a proof-of-concept system, and those were a steerable camera, two depth cameras, and a mobile device. For the steerable camera, an AXIS 214[1] network camera was used (Fig. 3.8(a)). This camera includes a small web server, which accepts specific messages to control its direction. Therefore, the system can change its heading direction by sending messages; it fits to the purpose of Point-Tap and Tap-Tap. Microsoft's Kinect[2] cameras were used as depth cameras (Fig. 3.8(b)). Now it is easily purchasable with reasonable cost. For the mobile device, we used a Sony UX50[3] (Fig. 3.8(c)). It runs on common Microsoft's Windows desktop operating system (i.e., Windows XP), which is more stable and development friendly than common mobile operating systems [79].

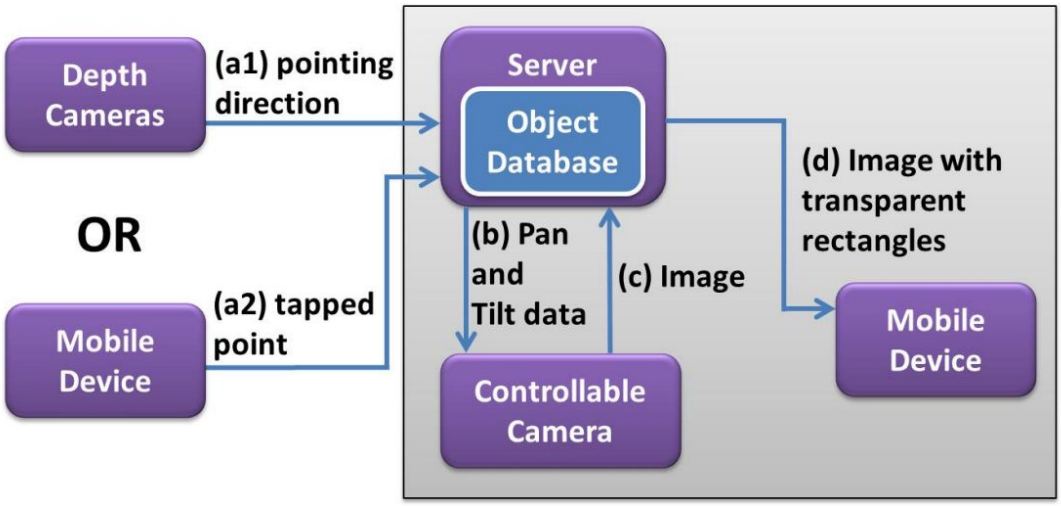

Figure 3.9: Architecture of the implementation.

3.4.2 Overview

Figure 3.9 illustrates the overall architecture and execution flow of Point-Tap and Tap-Tap together. The difference between the two methods is the beginning of the execution. In Point-Tap, the execution starts by sending a pointing direction (Fig. 3.9(a1)), but Tap-Tap starts by sending a tapped point (Fig. 3.9(a2)). Here, we provide an overview of the two techniques.

For Point-Tap, it is possible to track pointing direction by using depth cameras [80]. Tracked pointing data are sent to the server (Fig. 3.9(a1)), which starts to find the nearest object from the pointing direction. This is possible because the server maintains an object database that stores the location of each object. Then, the server sends the pan and tilt values of the detected object to the steerable camera (Fig. 3.9(b)), causing the camera to change its aim towards the designated object. The camera starts to take images and sends them back to the server (Fig. 3.9(c)). The server adds transparent rectangles for marking selectable objects and sends that image to the mobile device (Fig. 3.9(d)). Finally, the mobile device can display the image shown in Figure 3.6(b).

[1] http://www.axis.com/products/cam_214
[2] http://www.xbox.com/kinect
[3] http://www.vaio.sony.co.jp/Products/UX1/feat1.html

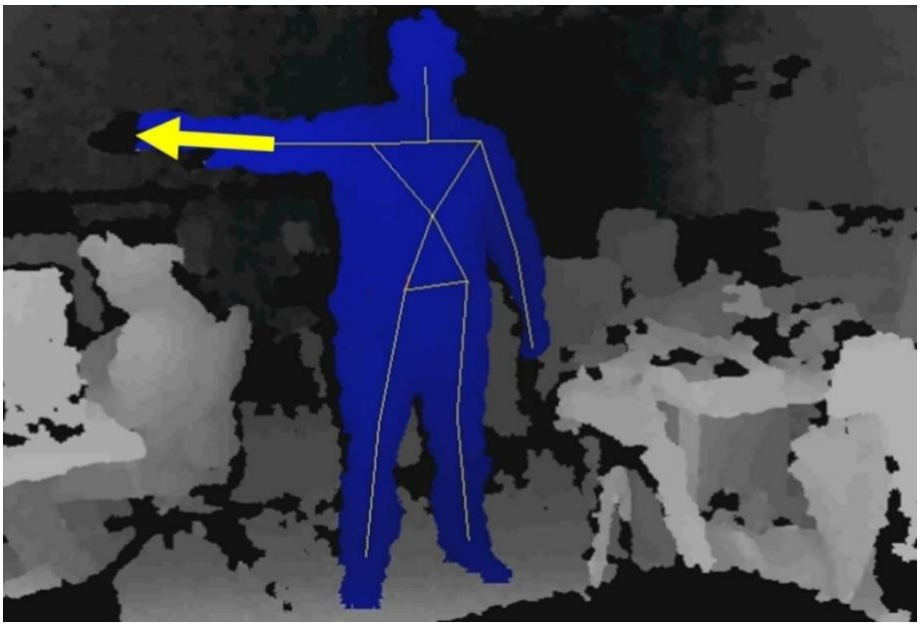

Figure 3.10: Visualization of a tracked user. The blue are indicates the tracked user's body, and the yellow arrow (i.e., from elbow to hand) designates the user's pointing direction.

The flow of Tap-Tap is essentially the same as Point-Tap, except at the beginning. When the user taps a point in the image (Fig. 3.7(a)), the tapped point is sent to the server (Fig. 3.9(a2)). The server finds the nearest object to the tapped point. The remainder of the process is the same as in Point-Tap (Fig. 3.9(b)-3.9(d)).

3.4.3 Tracking Pointing Gesture

For tracking pointing gesture there are two considerations: tracking pointing gesture itself and synchronizing two different camera's 3D spaces.

Tracking pointing gesture means that the system needs to construct a pointing ray (i.e., a vector from a body part to another), and the system exploits the elbow and the hand to construct the pointing ray. Figure 3.10 shows it.

Obviously the system needs to track the skeleton points (i.e., the elbow and the hand). Fortunately, OpenNI [80], which is an implementation of utilities for Microsoft's Kinect camera, provides most of human skeletons and Figure 3.11 shows it.

The next task is to merge two different cameras' 3D spaces into one space (i.e., synchronization). Indeed, it is difficult to track users' pointing movements in different directions

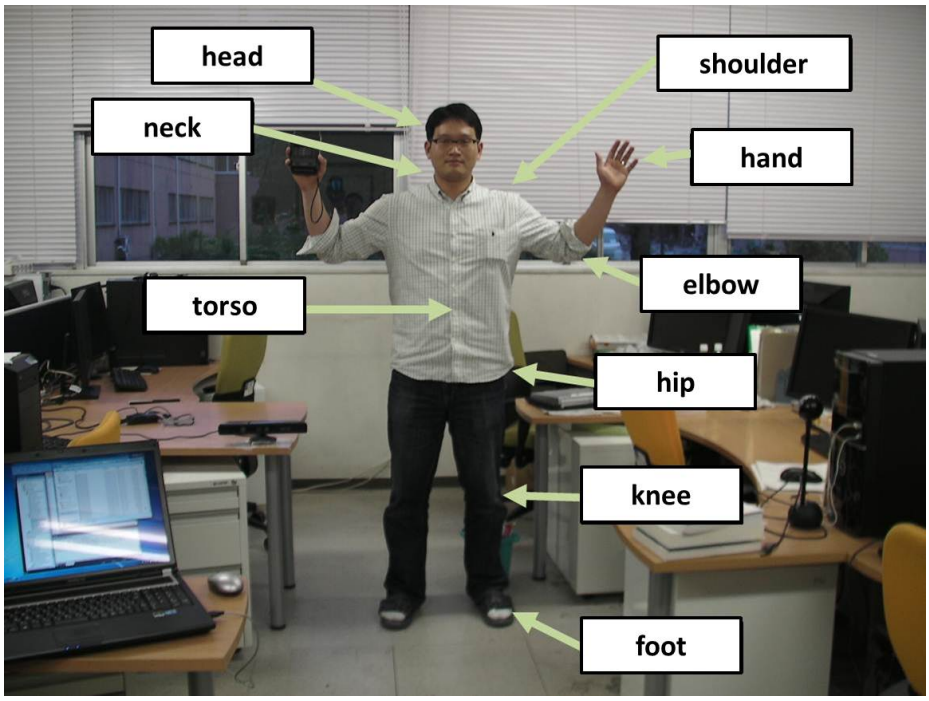

Figure 3.11: Skeletons tracked by Microsoft's Kinect.

accurately if using only one camera [81]. To address this, the system is using two Microsoft Kinect depth cameras, and they were set up to face each other. The depth cameras have its own 3D space with respect to its posture (i.e., position and orientation) [82][83]. Therefore when their spaces are seamlessly merged, the system can track user's pointing gesture for all directions. In next section, the detail is given.

3.4.3.1 Synchronizing Two Depth Cameras

To combine the two 3D spaces of the depth cameras, we arbitrarily picked three points in the shared area (i.e., commonly covered area by two cameras). And, it gathers the measured values based on the two different coordinate systems of the cameras. Even if the measured values are different with respect to the location and orientation of the cameras, they were located in the same place. The synchronization is achieved by using this feature.

Figure 3.12 illustrates the concept. Camera 1 and camera 2 in the Figure can be considered two depth cameras for the system, and they are heading to the directions of Figure 3.12(v1) and (v2) respectively. Then, ultimately what the system needs to build is an equation to convert a location based on camera 1's coordinate into a location in camera

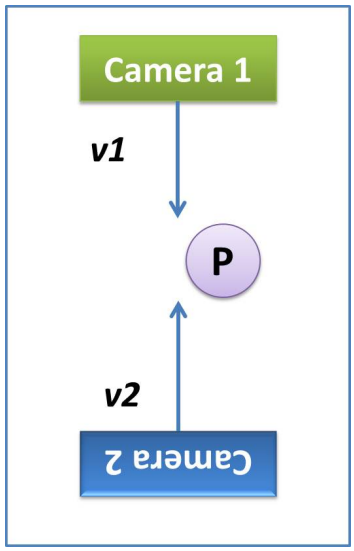

Figure 3.12: A point (P) between two cameras.

2's coordinate or vice versa. Therefore, finding a matrix M in Equation 3.1 is the specific task of synchronization.

$$\begin{aligned} &M(x, y, z) = (x', y', z') \\ &\text{where } x,\ y,\ z \text{ are the position of P based on camera 1's coordinate} \\ &\text{and } x',\ y',\ z' \text{ are the position of P based on camera 1's coordinate} \end{aligned} \tag{3.1}$$

The Equation 3.1 conceptually describes it simply, but it requires more complex, in terms of calculation, process. Indeed, we need to have three reference points (i.e., the point P in Fig. 3.12). While monitoring the images from both cameras, we select three points arbitrarily, and the following Equation 3.2 can build a rotation matrix (i.e., MR) that synchronizes the direction of two cameras.

$$
\begin{aligned}
&\vec{VA} = pa1 - pa2 \\
&\vec{VB} = pb1 - pb2 \\
&angle1 = acos(\vec{VA} \cdot \vec{VB}) \\
&axis1 = \vec{VA} \times \vec{VB} \\
&M1 = BuildRotationMatrix(angle1, axis1) \\
&\vec{VA} = (pa1 - pa2) \times (pa2 - pa3) \\
&\vec{VB} = (pb1 - pb2) \times (pb2 - pb3) \\
&angle2 = acos(\vec{VA} \cdot \vec{VB}) \\
&axis2 = \vec{VA} \times \vec{VB} \\
&M2 = BuildRotationMatrix(angle2, axis2) \\
&MR = M1 \times M2 \\
&\\
&\times : CrossProduct \\
&\cdot : DotProduct \\
&\text{where } pa1, pa2, \text{ and } pa3 \text{ are points from camera1} \\
&\text{and } pb1, pb2, \text{ and } pb3 \text{ are points from camera2}
\end{aligned}
\tag{3.2}
$$

The *RotationMatrix* that is given from the Equation 3.2 enables three dimensional spaces of two cameras to be aligned. The next step is to build a translation matrix. The translation matrix can be given easily through following Equation.

$$
\begin{aligned}
&\vec{VA} = pa1 \\
&\vec{VB} = pb1 \\
&MT = Buildtranslation Matrix(p1.X - p2.X, p1.Y - p2.Y, p1.Z - p2.Z) \\
&M = MR * MT \\
&\text{where } p1 \text{ is a point from camera1} \\
&\text{where } p2 \text{ is a point from camera1} \\
&MR \text{ is from Equation 3.2}
\end{aligned}
\tag{3.3}
$$

```
<Space>
  <Object Name="Printer"
          PosX="532.3" PosY="-253.5" PosZ="1352.4"
          Pan="-25.5712" Tilt="19.65"/>
</Space>
```

Figure 3.13: Example of object database in XML format.

Finally, we can build a matrix M in Equation 3.3 (i.e., also in 3.1) by applying MR in Equation 3.2 and MT in Equation 3.3.

After the synchronization, the system could track a user's pointing gesture in all directions. The system tracks the points of the hand and elbow, and the vector is used as a pointing ray. Figure 3.10 shows the visualization of a tracked user, and the yellow arrow illustrates a tracked pointing ray.

3.4.4 Server

There are two main roles in the server: maintaining object database and marking objects. Here the details are described.

3.4.4.1 Object Database

Figure 3.13 shows an example of an object database. The database is stored in XML format. The element *Object* contains six attributes. The attributes $PosX$, $PosY$, and $PosZ$ are the position of objects in the camera's space (i.e., synchronized through the process mentioned in the above). This position is used when a pointing vector is detected. Using the direction of pointing ray and the position of objects it finds the nearest object (i.e., using orthogonal distance between line and point).

The name attribute indicates the name of the object, and it is used to mark the names of the objects on transparent rectangles (Fig. 3.6(b)). The attributes pan and tilt represent absolute horizontal and vertical rotation of the steerable camera. When user's pointing gesture is recognized, the server sends those numerical values (i.e., the pan and tilt) to the camera and makes it change the direction.

Indeed construction of such object database is a tedious routine; the manual way of gathering locations and editing XML file is time-consuming. To ease this process, it provides

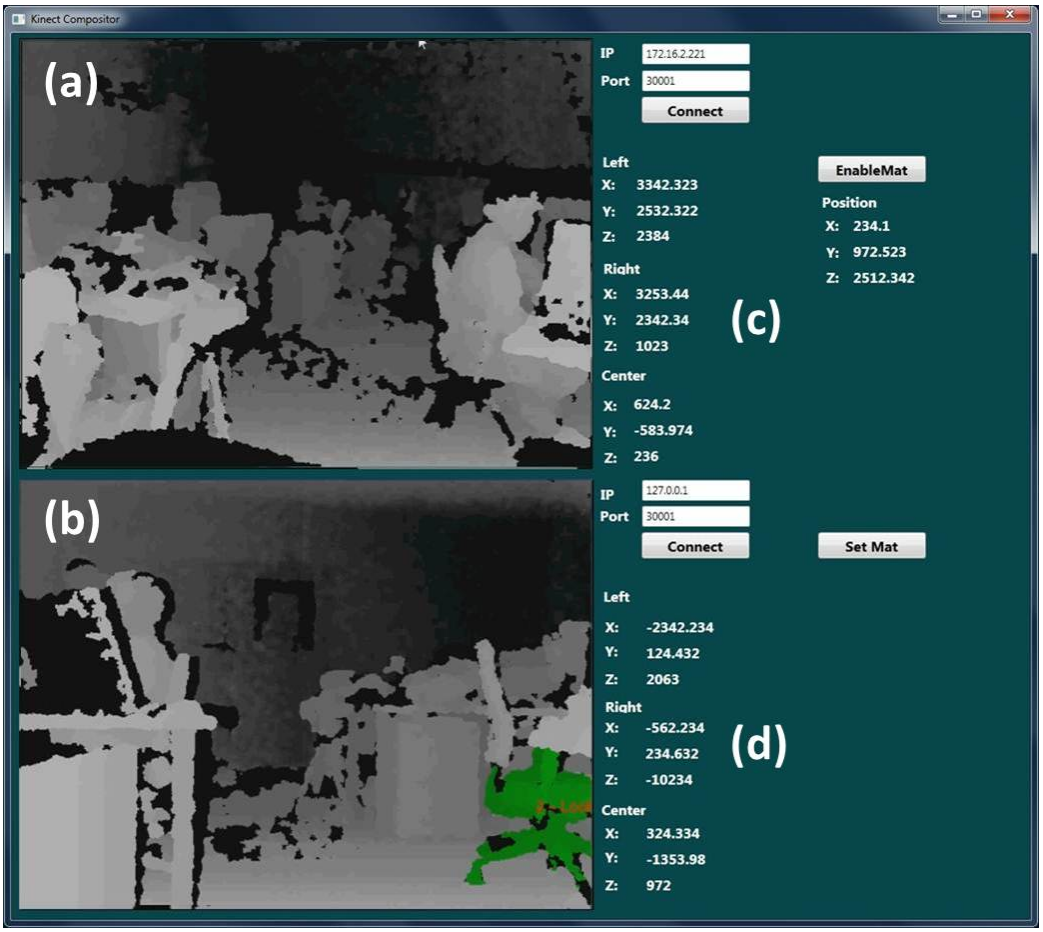

Figure 3.14: A toolkit for synchronization and object database construction.

a GUI toolkit and Figure 3.14 shows it. As mentioned, the prototype system uses two depth cameras. Therefore, this application has two graphical panes for each cameras (i.e., Fig. 3.14(a) and (b)). When users click a point of the graphical panes, the location is shown around Figure 3.14 (c) and (d), which are for cameras of Figure 3.14(a) and (b) respectively. Consequently users are able to collect the locations of objects through by clicking the points and input its name. Then, it generates the XML database file automatically.

The steerable camera's API enables accessing to the absolute pan and tilt values. After constructing the database, we used the software given from the camera's vendor. First we adjust the camera to direct a target object. Then, we could gain the pan and tilt value through camera's API. Then we manually input the values on the automatically generated database file.

Table 3.2: Summary of related selection techniques based on see-and-select. Each method has advantages and disadvantages. With Point-Tap and Tap-Tap, it is potentially able to satisfy all attributes in the table.

	Pointing based [58] [62][68][59]	Proximity based [66]	Live Video on Tabletop [64]	Live Video on Mobile [70]	*Point-Tap*	*Tap-Tap*
Occlusion	hard to avoid	avoidable	avoidable	avoidable	avoidable	avoidable
High Density	hard to avoid	avoidable	avoidable	hard to avoid	avoidable	avoidable
Remote Selection	feasible	infeasible	feasible	feasible	feasible	feasible
Mobility	dependent	sufficient	limited	sufficient	dependent	sufficient

Dependent : It depends on the capability of tracking system.

3.4.4.2 Marking Objects

As explained above, the system marks selectable objects with half-transparent rectangles. The system draws these by considering the pan and tilt values of all objects in the database. Because pan and tilt describe absolute directions from the camera, it is possible to determine whether the object is in view with respect to the current pan and tilt values (i.e., absolute direction) and the field of view of the camera. Indeed, it is similar to common techniques of projecting 3D point onto a 2D plane, which are commonly dealt in computer graphics or linear algebra textbooks [84][85]. If the object is determined to be shown in the image, the system draws a rectangle with its name at the appropriate position.

3.5 Comparing with Existing Techniques

Table 3.2 compares Point-Tap and Tap-Tap with related techniques. Each technique has advantages and disadvantages. By adding one more live video view from the camera on the ceiling, the designed techniques can satisfy all of the attributes in Table 3.2.

The pointing based techniques can face occlusion problems and this was described in the above in detail. The pointing gesture also can be problematic when there are numerous objects (i.e., high density). Generally humans' pointing gesture is not much accurate [57], but the environment of high density requires precise pointing. Therefore it can be problem. Obviously pointing gesture supports remote selection. Its mobility is dependent on its

implementation. If the system can track users or pointing device (e.g., XWand by Wilson and Shafer[59]) fully in the space, its mobility can be guaranteed. Otherwise, it would be limited.

Proximity based techniques can avoid occlusion and high density problem. Obviously users need to be near to the target object because the selection occurs the distance between devices within a certain threshold. However, it cannot support remote selection. This proximity based technique can sufficiently support the mobility. Generally users should be with a RFID tag; the mobility is guaranteed if the user is with the RFID.

Live video on large tabletop and on mobile can face occlusion problem. Because both techniques mainly rely on the camera on the ceiling, which is a good location for avoiding occlusion problem. Live video on large tabletop may not face problems of high density. Because it draws the image on large platform. However, live video on mobile device is hard to avoid this problem. Both techniques can support remote selection. The large tabletop is usually not portable; it is hard to support the mobility. However, live video on mobile can obviously support the mobility.

3.6 Conclusions and Future Work

Point-Tap and Tap-Tap are based on the pointing gesture and a map with live video techniques, respectively. The new techniques were designed to enhance the capabilities of both the pointing gesture and a map with live video systems, and the concept was verified by comparing them to related techniques.

Current prototype system exploited only two cameras, which is obviously not enough to cover a space wholly. However, the architecture of the system is developed in distributed way; it can be extensible with more cameras. Developing such system and finding more sophisticated architecture is a future direction around the implementation.

Chapter 4

The Effect of Familiarity of Point-Tap and Tap-Tap

4.1 Introduction

The purpose of this chapter is to introduce a human factor that has significant effect to proposed two techniques (i.e., Point-Tap and Tap-Tap), and the factor is the familiarity with the space. A controlled user experiment was conducted and it confirmed that the familiarity has the significant effect. Here the motivation, the detail of the experiment, and the implication from the result are presented.

Indeed picking an object among multiple objects implies that humans should distinguish an object from the other objects first. For example, we humans can easily pick up an apple when the apple is among tens of water melons. It means we refer to the color or shape of the apple, even without consciousness, therefore we can distinguish the apple from the water melons.

This implies that the referable property, which is engaged to selection techniques, can have effect on the performance of the selection techniques. For example, if we humans are good at distinguishing colors rather than characters, basically selection techniques that refer to colors will be better than the techniques that refer to characters.

Such referable property (i.e., color or characters in the examples) is determined with respect to the characteristics of selection techniques. For example, we are asked to pick an

Figure 4.1: Three components of selection techniques.

Table 4.1: Different referable properties of Point-Tap and Tap-Tap at different spaces.

Many similar objects (e.g., office / laboratory)	Not-many similar objects (e.g., common living room)
Locations	External shape

apple without vision, and then we need to rely on the tactile sense. Therefore first we need to clarify *which property is referred in Point-Tap and Tap-Tap.*

The next is to determine the interaction between humans' cognition and the referable property. Again with the aforementioned example (i.e., if humans can distinguish colors better than characters), indeed such assumption (if it is true) implies that there are some relations between humans' cognition and referable property. Why are humans better to recognize colors than characters? The investigation on the interaction between such properties (i.e., in here the color and characters) and human's cognition may give answers. Therefore *the relation between referable property of Point-Tap and Tap-Tap and humans' cognition should be investigated.*

Figure 4.1 depicts the relations between three components, which are selection techniques, referable property, and human's cognition. Obviously specific selection technique requires specific referable property (Fig. 4.1(a)). And psychologically there might be relations between humans' cognition and referable property (Fig. 4.1(b)). Therefore the links between three components (i.e., Fig. 4.1(a) and (b)) should be clarified.

As mentioned in the earlier, this chapter presents the result of the experiment that investigated the effect of the familiarity to Point-Tap and Tap-Tap. First, it clarifies the referable property of Point-Tap and Tap-Tap in specific situation. And next the relation between the referable property and the familiarity is described.

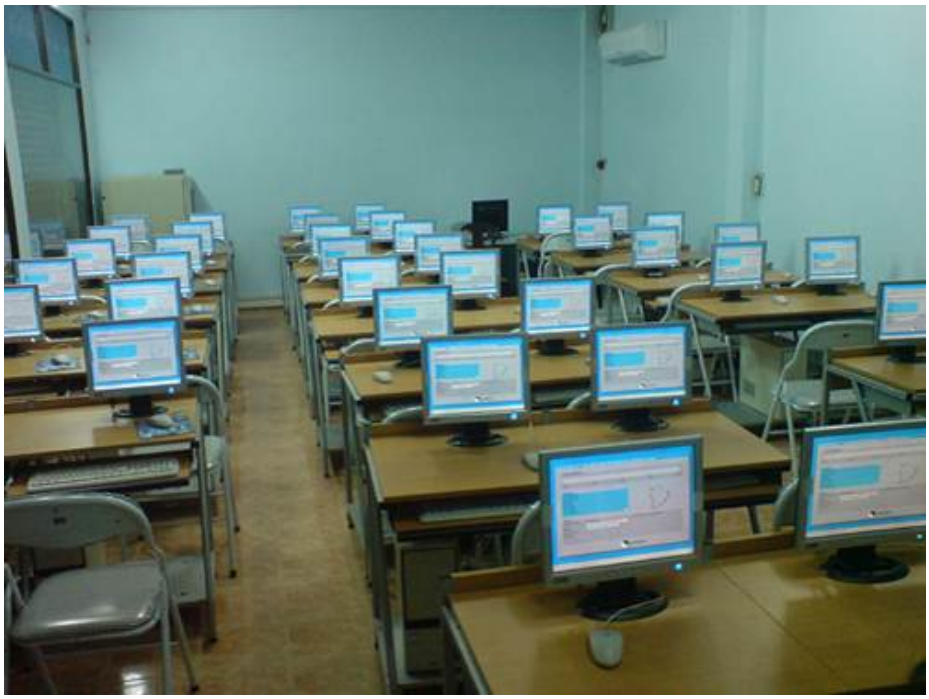

Figure 4.2: A common laboratory.

4.2 Referable Property Point-Tap and Tap-Tap at a Scenario

The scenario that this study focuses is that *the users rely on the locations of objects as a referable property when using Point-Tap and Tap-Tap.* The detailed background of this scenario is given in this section.

Indeed, the referable property of Point-Tap and Tap-Tap becomes different with respect to the characteristic of the space, and Table 4.1 shows it. If there are not many similar objects, users are able to refer only outer shape. For example, in common living room users can recognize a microwave or a television easily by seeing its outer shape. However, if there are many similar objects, users are hard to refer the external shape any more. For example, in common offices or laboratories, there are many computers or displays and Figure 4.2 shows an example [1]. In this case, indeed there is almost no means of designating a display except referring locations.

Point-Tap and Tap-Tap can have more benefit when there are many similar objects, because it mainly relies on the locations (see Table 3.1). Indeed when there are not many

[1] http://preilly.files.wordpress.com/2008/10/computer-lab1.jpg

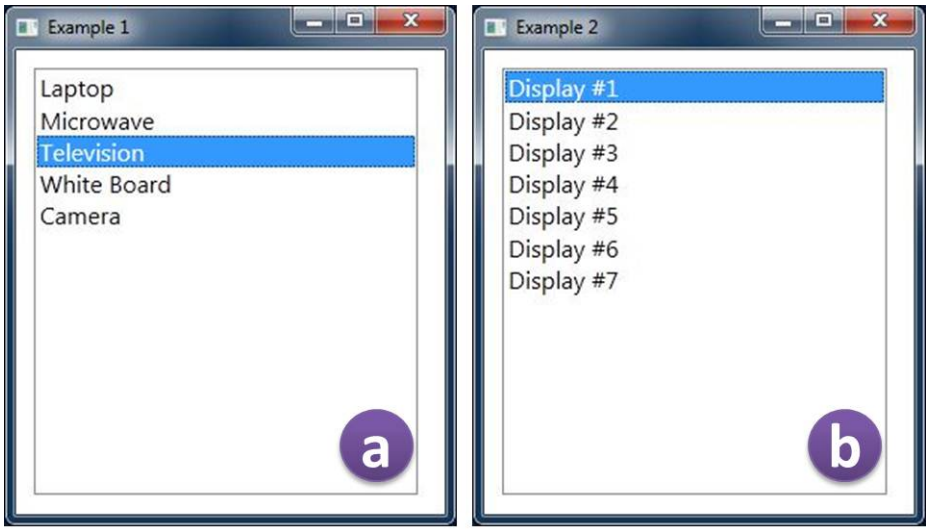

Figure 4.3: Examples of showing lists of devices in different spaces.

similar objects, they also can provide easy identifiers. Figure 4.3 shows it. Both images (i.e., Fig. 4.3(a) and (b)) exemplify common GUIs, and they show the list of devices in a space. As shown in here, in common living room (Fig. 4.3(a)), users can easily pick up a television from the list because literally it represents the objects well. However, in office like environment (Fig. 4.3(b)), it is hard to select a display identifier from the list. Therefore, in this case Point-Tap or Tap-Tap is more preferable. And, in this scenario it requires referring the locations.

4.3 Theoretical Background of Human Cognition on Location Recognition

The next is to find the relation between humans' cognition and referable property (i.e., the location).

The users should refer to locations of objects when using the Point-Tap or Tap-Tap in the aforementioned scenario. Indeed referring to the locations implies that the users who memorize the locations better can use the techniques more efficiently.

Obviously we can expect that users who use a space in everyday (i.e., familiar) may be memorizing the locations better than users who occasionally visit the space (i.e., unfamiliar). Therefore, it raises the question as to whether the users who are familiar with the

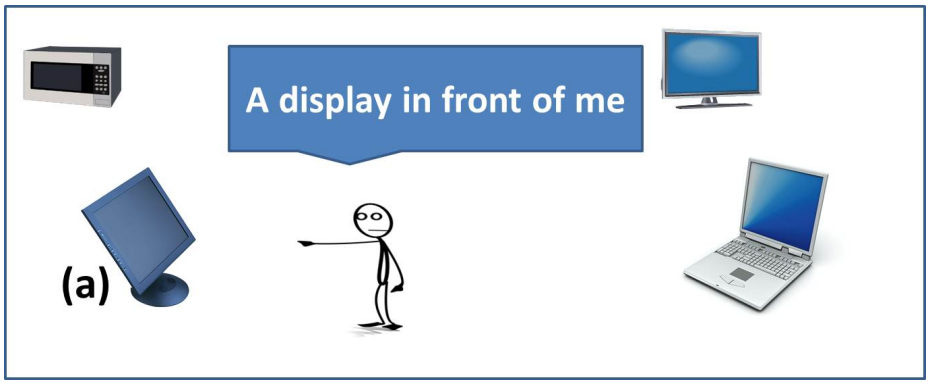

Figure 4.4: Illustration of egocentric representation.

space exploit the see-and-selection techniques more efficiently. Another aspect on specific techniques (i.e., Point-Tap and Tap-Tap) is whether the users who familiar and unfamiliar with the space show similar performance on both techniques or not.

To study the effect of such familiarity we need to understand how humans recognize the location in brain. Theoretically, there are mainly two representations that humans recognize a location in the brain and details are given in the following section.

4.3.1 Two Representations

Humans recognize a location of object in two ways [86]. Those are *egocentric* and *allocentric*. The details of two representations are given in this section.

4.3.1.1 Egocentric

In egocentric way humans recognize a location with the relative distance and direction from him/herself to an object. For example, when there is a video player, it can be memorized with a phrase "a video player at the left side of myself". Figure 4.4 illustrates it. When there are four devices in the space and a user is placed at the center. Then, the user designates a display (Fig. 4.4(a)) with egocentric way (i.e., a display in front of me).

4.3.1.2 Allocentric

Allocentric is different. In this way, users memorize the location of an object by analyzing the relative distance and direction between objects. For example, such a phrase "a video player at the left side of a television" can be used for memorizing a location. Figure

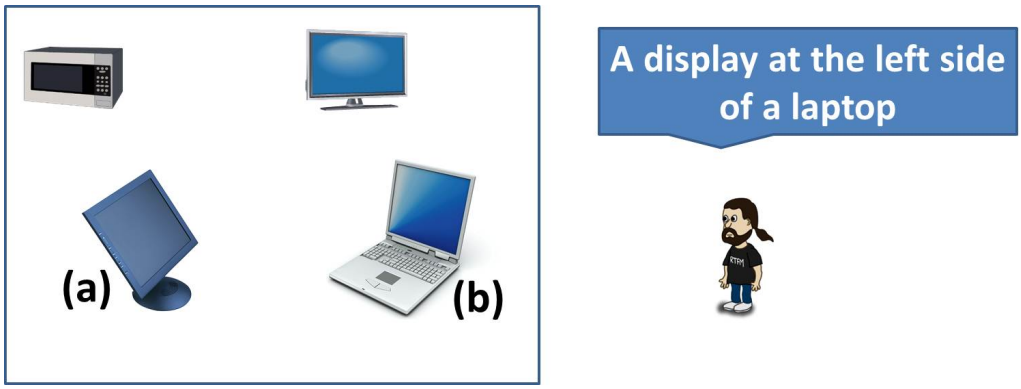

Figure 4.5: Illustration of allocentric representation.

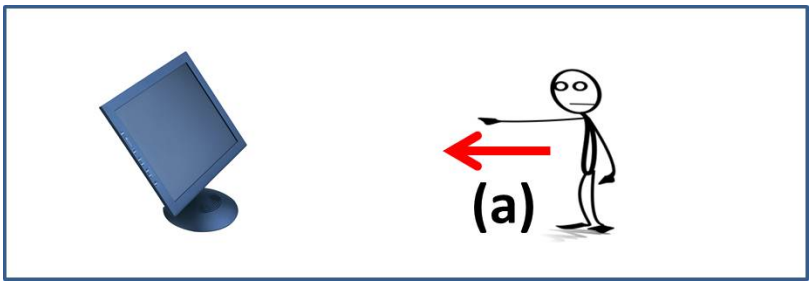

Figure 4.6: Pointing vector in egocentric representation.

4.5 illustrates it. In here the user is not in the space. Therefore he is not able to describe an object in egocentric way. In here the user designates a display (Fig. 4.5(a)) by describing a relative location from a laptop (4.5(b)).

4.3.2 The Relations between the Representations and Point-Tap, Tap-Tap

Point-Tap and Tap-Tap have strong relations with two representations respectively. Indeed Point-Tap is a technique of using pointing gesture mainly and Tap-Tap relies on a map-like interface. Therefore users need to have egocentric representation more when using Point-Tap. Pointing gesture is to designate an object by making a vector from the user to the object, and the vector always starts from the user (see Fig. 4.6(a)). Therefore it requires egocentric memorization for making pointing gesture.

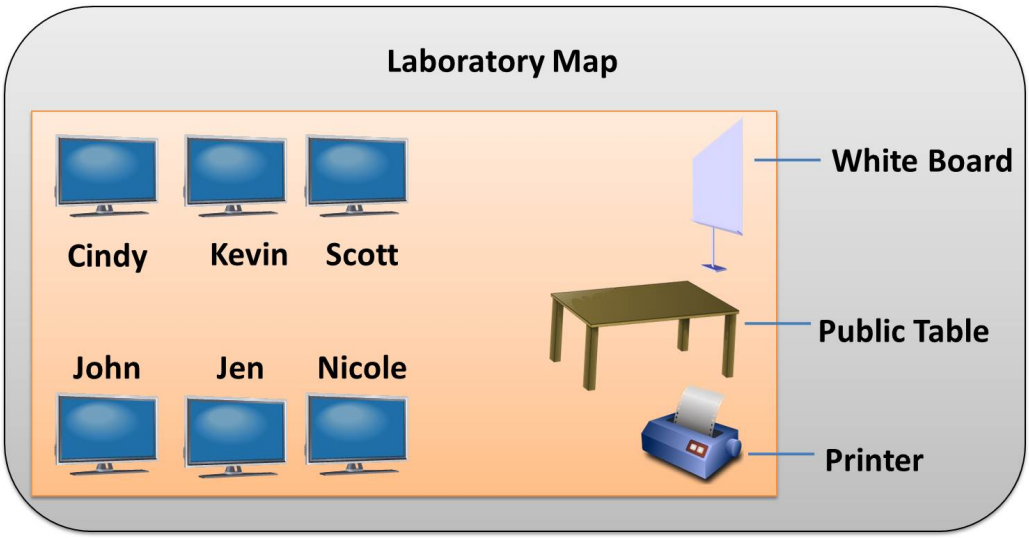

Figure 4.7: Example of a laboratory map.

Tap-Tap is different. Indeed Tap-Tap relies on a video from a camera that covers whole range of the space and it can be considered a map from the interaction centered view. Figure 4.7 shows it. When such map is given, there is no user in there.

Such maps can be considered a WIM (World In Miniature) in virtual or augmented reality studies [87][88], and those WIM techniques are regarded as exocentric metaphor [89]. Therefore, the user should remember a location of an object by memorizing relative direction and distance from an object with this scenario; it is the same way of allocentric representation.

In summary when considering the features of Point-Tap and Tap-Tap, they require different representations; *Point-Tap relies on egocentric representation but Tap-Tap relies on allocentric representation.*

4.3.3 The Representations, Familiarity, and Hypothesis

There is an interesting relation between the aforementioned two representations and different familiarity; *the available amount of allocentric representation is being more after users become familiar with the space* [90]. It implies that there will be significant difference between users who have different familiarity to the space if technique relies on allocentric representation mainly.

Table 4.2: Summary of two user groups.

Familiar group	*Unfamiliar group*
Users who work at the space in everyday	Users who have visited the space less than five times in a month

As mentioned in the previous section, Point-Tap and Tap-Tap rely on egocentric and allocentric representations respectively. Therefore we can draw the following hypothesis.

The users who are familiar with the space will use Tap-Tap more efficiently; the different familiarity will have significant effect on Tap-Tap.

4.4 Experiment

4.4.1 Overview and Objective

There are two objectives in the experiment. The first is to verify the hypothesis. The experiment measures the quantitative data of performance of Point-Tap and Tap-Tap from the users who have different familiarity with the space. If a significant effect is found, then we can argue that the familiarity with the space should be considered when deciding a selection technique among pointing gesture and map based selection techniques.

The second is to observe user satisfaction. After the experiment, we had a small questionnaire, which asks preference.

After having data from two types of experiment, it is expected that a guide of considerations for deciding appropriate selection techniques with respect to higher priority of the space (i.e., performance or user satisfaction) will be given.

The detail of the design is described from the next section, and the prototype system explained in chapter 4 is used for the experiment.

4.4.2 Experiment Environment

4.4.2.1 Selection and Arrangement of Target Objects

Figure 4.8 shows the installation layout and real environment. In Figure 4.8(a), the numerical values in parentheses below each object describe its location, and the origin of coordinate is the top-left corner. Please note that the numerical values are exact but the locations of the objects in the figure were adjusted to include all objects within one image.

There were eight objects for the experiments, all of which were displays. This was intended to make users to refer only locations when making selections. If the experiment is given with different objects (e.g., a television and a microwave), then the users are able to distinguish the objects by referring their external shapes rather than by their locations. Again a critical point of this experiment is to measure the performance of location recognition. Therefore, similar objects should be given as targets for making users refer to only the locations.

Figure 4.8(b) shows the real environment. Displays were located on top of desks (Fig. 4.8(b1) - 4.8(b3)), and the desks were used by real users. For the experiments, we did not adjust the original layout of the space, and this was because users who are familiar with the space can become unfamiliar if the objects are arbitrarily moved for the experiment. Another critical point of the experiment is to examine the effect of different familiarity. Therefore, the original layout of the space should be maintained.

To designate target displays, real owner's names were used (e.g., Nicole's display or Scott's display). The application for the experiment displayed the users' names. When there are multiple displays on a desk, we asked users to designate the middle of the displays.

4.4.2.2 User Groups

One of objectives in this experiment is to examine the relations between two techniques and users who have different familiarity with the space. Therefore, we recruited two user groups. Users in first group basically work in the space every day; therefore they are familiar with the space. In this paper *unfamiliar group* denotes it. The other is *unfamiliar group*. The users in this group had visited the space less than five times; therefore they are not familiar with the space. Table 4.2 summarizes two user groups.

There were seven users in each group. In total, 14 users between the ages of 24 and 30 years (average, 26 years) participated. There were 12 males and 2 females.

4.4.2.3 Tasks and Measured Values

For the experiment, the application shown in Figure 4.9 was developed. When the user pushes the start button (Fig. 4.9(a)), the system shows the name of a target object (Fig. 4.9(b)). Then users start to make selections for each target object. After one selection is completed, the name of the next target object is shown automatically.

Table 4.3: Overall performance and preference.

Method Type	Time (s)	Preference
Point-Tap	11.54	10 users (71%)
Tap-Tap	10.52	4 users (29%)

We asked the users to make selections for each object using Point-Tap and Tap-Tap. The order of the selection for objects was identical for all users and both types.

Obviously users in unfamiliar group do not know the locations of objects. Therefore, they need some time to memorize it. Before having test, all users (in familiar and unfamiliar groups) had 10 min to practice both methods. In this time, the users who are not familiar with the space were asked to memorize the locations of the objects.

For all cases, we measured the time taken to complete the selection. All users had experiment with Point-Tap first and then had with Tap-Tap.

4.4.3 Result

4.4.3.1 Effect of Familiarity

Familiarity had a significant effect on Tap-Tap but not on Point-Tap. Figure 4.10 shows graphs illustrating the effect of familiarity on Point-Tap and on Tap-Tap broken down by group. The groups performed similarly for Point-Tap, where the familiar group took 12.8 s and the unfamiliar group took 12.9 s, on average. The standard errors of the mean were 0.67 s and 1.28 s respectively. However, they showed relative big difference with Tap-Tap (8.2 s and 13.8 s in familiar and unfamiliar groups, standard errors of the mean were 0.37 s and 0.86 s).

To validate this difference in average we had one-way analysis of variance (ANOVA) test, and it showed that $F(1,14) = 0.0006$, $P = 0.97$ in Point-Tap case; it implies that the familiarity has no significant effect statistically also. In contrast, the results for Tap-Tap showed the significant effect, with $F(1,14) = 13.37$, $P < 0.01$.

Those statistic data means that the familiarity with the space showed significant effect on Tap-Tap (map based interface) but not with Point-Tap (pointing based). Therefore we could confirm that the theory in spatial cognition [90] is valid in this specific case (i.e., with Point-Tap and Tap-Tap).

4.4.3.2 Point-Tap versus Tap-Tap

The two techniques showed similar performances, but more users preferred Point-Tap than Tap-Tap. As shown in Table 4.3, Point-Tap and Tap-Tap took 11.54 and 10.52 s, respectively on average. Point-Tap took slightly longer. This would be due to the time taken for the physical movement of the pointing gesture.

4.4.3.3 User Satisfaction

As mentioned in the above, another aspect that this experiment confirms is user satisfaction. After the experiment (i.e., having selected all objects using both methods), a small questionnaire was given. This questionnaire asked more preferable method to participants.

Ten users (71%) responded that they preferred Point-Tap over Tap-Tap (see Table 4.3). Most of participants who preferred Point-Tap reported that they felt more intuitive and natural when making pointing gestures. The users who preferred Tap-Tap complained the physical fatigue; they have felt pain at the shoulder because they needed to make ten times pointing gestures. In particular two female participants preferred Tap-Tap over Point-Tap.

4.5 Implications and Discussion

Users' familiarity with the space had a significant effect on Tap-Tap but not on Point-Tap. This result is consistent with the theory of cognition science [90].

First, with Point-Tap (Fig. 4.10), both familiar and unfamiliar users showed almost similar performance, which means even users who are not familiar with the space had no problem of remembering the locations well. In contrast, with Tap-Tap two user groups showed different performance and it was statistically significant also.

It implies that the users who are not familiar with the space (i.e., unfamiliar group) were the locations of the objects with egocentric representations, i.e., the direction and distance from themselves to the objects, and thus they had no significant problem using a pointing gesture because it relies on the same representation. However, Tap-Tap requires different representation (i.e., allocentric), and this representation is not readily available until users have become familiar with the space [90]. Thus, it confirms that the familiarity with the space had a significant effect on Tap-Tap; the expected familiarity with the space should be

Table 4.4: Summary of beneficial places for Point-Tap and Tap-Tap.

	Familiar	*Unfamiliar*
Satisfaction	Point-Tap Ex) Staff lounge	Point-Tap Ex) Museum
Effectiveness	Tap-Tap Ex) Office	Point-Tap Ex) Lecture room

considered at least when deciding a selection technique among two techniques (Point-Tap and Tap-Tap).

With this observation, it can deduce a guide of deciding a selection technique with respect to user's familiarity with the space and higher priority.

Table 4.4 summarizes the choices. When user satisfaction is more important, Point-Tap will be more promising for both types of user (i.e., familiar and unfamiliar with the space). This is because 71% of users preferred Point-Tap over Tap-Tap (see Table 4.3). When effectiveness has a higher priority, Tap-Tap will be better if the users are familiar with the space because users in the familiar group took less time with Tap-Tap than Point-Tap (Tap-Tap: 8.2 s, Point-Tap: 12.8 s; see Fig. 4.10). However, if the users are unfamiliar with the space, Point-Tap will be more promising because they took 12.9 s with Point-Tap but 13.8 s with Tap-Tap (see Fig. 4.10).

An important factor for a usability test is the failure rate. In these experiments, there was no case of failure. In the experiment, users were asked to make a rough pointing gesture (or tap in Tap-Tap) and complete the selection by picking one rectangle on the mobile device screen. These were relatively easy tasks and the participants were hard to make failure cases without mistakes. Another reason of no failure case is that the density of target objects was low. Four objects were shown at the same time at maximum and there were no overlaps among objects. Those are the main reasons why there was no case of failure. It is expected that higher densities would affect the techniques differently, and further investigations in such an environment should be conducted.

The order of the experiment is important because it can have effect on its result. Therefore most of user experiments have a counterbalanced design. However, in this experiment has one order: all users first had the experiment with Point-Tap and after then had it with Tap-Tap.

However, the result of the experiment is still valid because the users who are not familiar with the space can become more familiar with the space through the test with Point-Tap.

The main reason of poor performance of unfamiliar group with Tap-Tap is they did not have much memory of allocentric representation and it can be gained after becoming familiar with the space. In the experiment, even though the users have a session with Point-Tap but still showed significant difference with Tap-Tap, which means even though the users have more time to become familiar with the space, indeed exactly the time of the first session, they still have less allocentric representation. Therefore, the result of experiment is meaningful and the implication is valid.

In general, males have better spatial cognition [91]. In our test, there were two female participants and they were in the unfamiliar group. We compared the results of the two females to the results of the male participants in the unfamiliar group, and found no significant difference. With Point-Tap, male and female users took 13.8 s and 12.1 s, respectively, on average. With Tap-Tap, they took 13.2 s and 15.2 s. Hence, female users showed a slightly faster speed with Point-Tap and a slower performance with Tap-Tap. We conducted a one-way ANOVA test with the averaged results of Tap-Tap, but there was no significant effect ($F(1,14) = 0.91$, $P = 0.38$).

4.6 Conclusions

The experiment focused on two factors: users' familiarity with the space and user satisfaction. The result showed that Tap-Tap has the significant effect from the familiarity but Point-Tap is not. This result is consistent to a theory in spatial cognition science (i.e., users have the more memory of allocentric representation after becoming familiar with the space). Therefore users' expected familiarity with the space should be considered at least when deciding an appropriate technique among two techniques.

In user satisfaction questionnaire, Point-Tap marked higher score than Tap-Tap. Most of users reported Point-Tap is likely more natural and intuitive. On contrast, some of users preferred Tap-Tap because pointing gesture in Point-Tap imposes some physical pain.

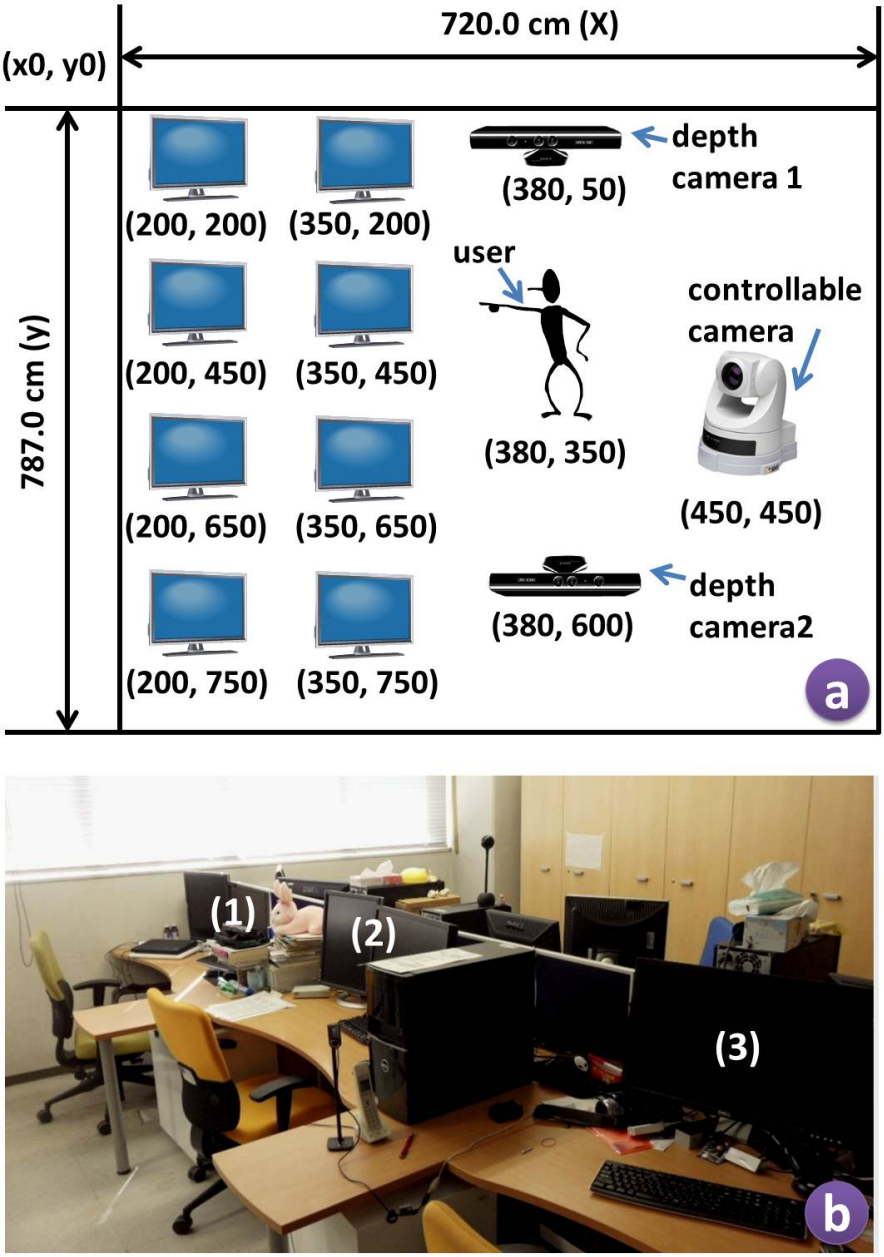

Figure 4.8: Experiment environment.

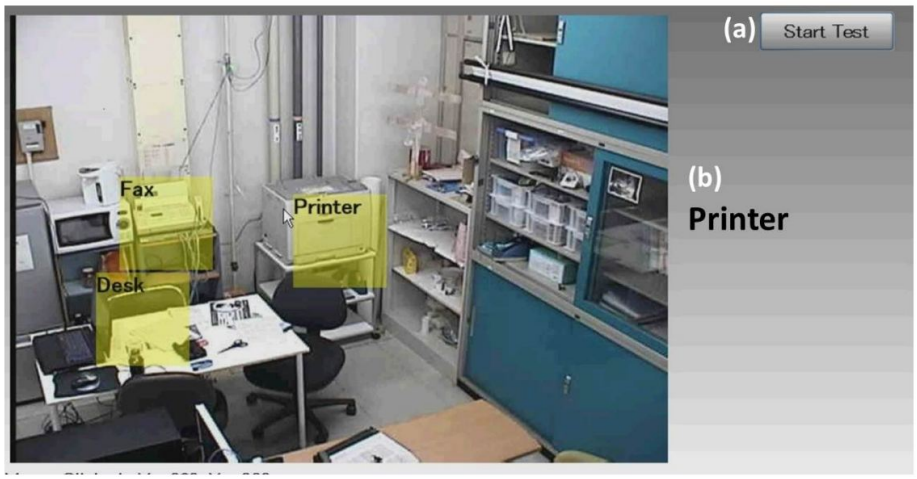

Figure 4.9: The application used for the experiments with both techniques.

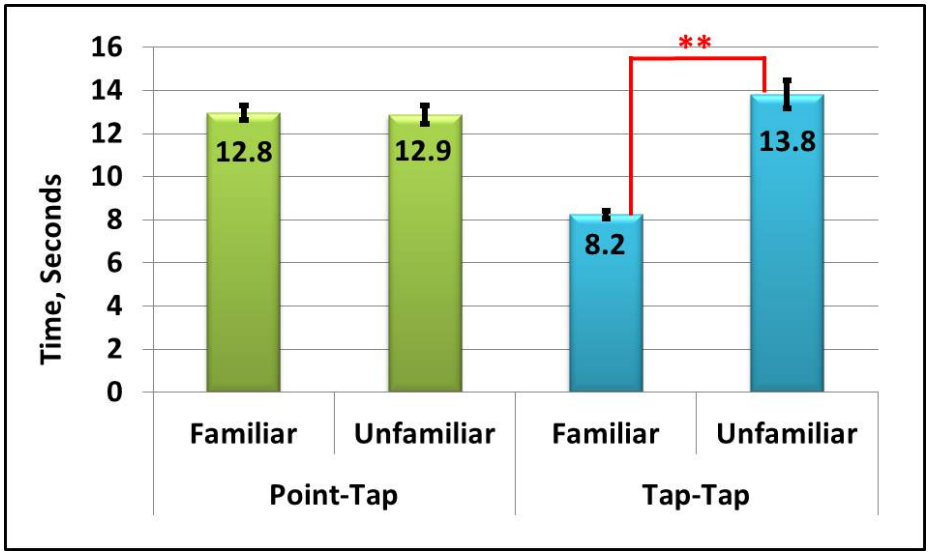

Figure 4.10: Result of the experiment.

Chapter 5

Conclusions

This study focused on providing more convenient tools for target acquisition in the augmented space. In augmented space, it is expected that there will be more and more devices; all of them are interactive, and users in the space need to access the devices more frequently. For having the interaction with such devices, obviously the users need to access the devices physically or have an interface connected to a target device. The process of accessing the devices is called the target acquisition.

Obviously the process of accessing a device requires some physical movement (i.e., holding a controller or approaching a physical device). To remove the physical movement, this study proposed an always available interface, which is called the palm display. The palm display is designed to enable an always available interface in a space without any temporal or spatial constraints. Therefore, the users were able to access to the palm display always; the users do not need to move physically to hold an interface.

Indeed, there have been many related studies to provide such always available interfaces. However, most of them were based on wearable computers; it requires at least small physical burden of holding a device. Or, some work provided such always available interfaces using infrastructure, which does not impose the physical burden. However, the existing work using infrastructure has not provided the direct manipulation metaphor, which is usually employed in common mobile device. Rather than, they provided some gesture or physical metaphor based interaction schemes, which may require some learning curve.

The palm display can be considered an infrastructure based always available interface and it provides the direct manipulation metaphor, which requires almost no learning curve to users. Indeed the palm display is the first work that provides such features, and this

is the main contribution of the palm display. With this concept, it enabled a prototype and proposed an implementation method of enabling it and the user study found that the current implementation can be used for applications with simple layout. The concept of the palm display will open the door to study this domain (i.e., infrastructure based interface that provides direct manipulation metaphor) and the proposed technology for enabling it will be a basic technology for improving the capability.

When such always available interfaces are freely given in a space, obviously users need to establish the connection between the interface and a target device (i.e., selection). Among existing selection techniques, the techniques with see-and-select concept are appropriate in particular when there are many similar objects (e.g., common offices or laboratories).

There are many techniques that provide the see-and-select features, and pointing gestures and live video based techniques among them can support the remote selection and the mobility; those two techniques (i.e., pointing gesture and live video on mobile) were selected as the study subjects.

The problem of pointing gestures is occlusion; indeed the pointing gestures can be valid only when there is no physical barrier between users and target objects. Problem of live video on mobile device is it can require precise pointing when there are too many selectable objects; obviously user should pick up an object precisely when there are too many objects. The problems on both techniques can be resolved when there is an additional view because the additional view can provide different perspective based on different locations; it can at least provide more plausibility of avoiding the occlusion problem. Also, this additional view can be used as a magnified view for small live video.

With those concepts, two techniques were proposed which are called Point-Tap and Tap-Tap respectively. Both techniques rely on a view from a steerable camera. The steerable camera was installed at the middle of the ceiling, which is the most promising position that can avoid the occlusion (i.e., for pointing gesture), and it can provide a naturally magnified view for small live video. They contribute to address the aforementioned problems and the concept was verified by comparing them with existing techniques.

The evaluation confirmed the significant effect of the familiarity with the space when using two proposed techniques. Two techniques rely on egocentric (Point-Tap) and allocentric (Tap-Tap) representations respectively. A theory in spatial cognition is that users can have more memory based on allocentric representation after they become familiar with the

space [90]. This theory enabled us to build a hypothesis: the users who are familiar with the space will use Tap-Tap more efficiently.

To verify the hypothesis, a controlled user experiment was conducted and it found the significant effect. It implies that the theory in spatial cognition study is still valid for this specific case (i.e., with Point-Tap and Tap-Tap) and this factor (i.e., the familiarity) can be considered when deciding a selection technique among two techniques. With this implication this study suggested a guide for deciding selection techniques with respect the priorities of spaces (see Table 4.3).

In conclusion this study found the followings.

1) It is able to provide an always available interface that supports direct manipulation metaphor but based on the infrastructure. Therefore the users can use the interface without any physical burden of holding a device because it relies on the infrastructure not on wearable computers. And also the users do not need to learn its usage because it provides the commonly used direct manipulation metaphor.

2) The problems of selection techniques with pointing gesture (i.e., occlusion) and live video on mobile (i.e., high density of objects population) can be addressed by using a steerable camera on the ceiling. The images from the steerable camera at least have more plausibility of avoiding the occlusion, because the camera is installed on the ceiling. Also, this view can be used for providing the naturally magnified view for the small live video based technique.

3) The familiarity with the space should be considered when deciding a selection technique for a space among proposed two techniques (i.e., Point-Tap and Tap-Tap). A controlled user study found that the significant effect on the familiarity. The user's familiarity with the space can be estimated by expecting the usage scenario of the space; the familiarity with the space is possible to estimate and should be considered.

Bibliography

[1] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, "Easyliving: Technologies for intelligent environments," in *Handheld and Ubiquitous Computing*, ser. Lecture Notes in Computer Science, P. Thomas and H.-W. Gellersen, Eds. Springer Berlin Heidelberg, 2000, vol. 1927, pp. 12–29.

[2] A. D. Cheok, X. Yang, Z. Z. Ying, M. Billinghurst, and H. Kato, "Touch-space: Mixed reality game space based on ubiquitous, tangible, and social computing," *Personal and Ubiquitous Computing*, vol. 6, no. 5-6, pp. 430–442, Jan. 2002.

[3] S. Helal and C. Chen, "The gator tech smart house: enabling technologies and lessons learned," in *Proceedings of the 3rd International Convention on Rehabilitation Engineering & Assistive Technology*, ser. i-CREATe '09. New York, NY, USA: ACM, 2009.

[4] D. Molyneaux and H. Gellersen, "Projected interfaces: enabling serendipitous interaction with smart tangible objects," in *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, ser. TEI '09. New York, NY, USA: ACM, 2009, pp. 385–392.

[5] D. Molyneaux, H. Gellersen, G. Kortuem, and B. Schiele, "Cooperative augmentation of smart objects with projector-camera systems," in *Proceedings of the 9th international conference on Ubiquitous computing*, ser. UbiComp '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 501–518.

[6] C. Pinhanez, R. Kjeldsen, L. Tang, A. Levas, M. Podlaseck, N. Sukaviriya, and G. Pingali, "Creating touch-screens anywhere with interactive projected displays," in *Proceedings of the eleventh ACM international conference on Multimedia*, ser. MULTIMEDIA '03. New York, NY, USA: ACM, 2003, pp. 460–461.

[7] M. Weiser, "Some computer science issues in ubiquitous computing," *Commun. ACM*, vol. 36, no. 7, pp. 75–84, Jul. 1993.

[8] A. K. Dey and A. Newberger, "Support for context-aware intelligibility and control," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 859–868.

[9] L. Barkhuus and A. Dey, "Is context-aware computing taking control away from the user? three levels of interactivity examined," in *In Proceedings of Ubicomp 2003*. Springer, 2003, pp. 149–156.

[10] A. Frescha, *Implicit Interaction*. CRC Press, 2009.

[11] O. Kaufmann, A. Lorenz, R. Oppermann, A. Schneider, M. Eisenhauer, and A. Zimmermann, "Implicit interaction for pro-active assistance in a context-adaptive warehouse application," in *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, ser. Mobility '07. New York, NY, USA: ACM, 2007, pp. 729–735.

[12] A. Wilson and N. Oliver, "Multimodal sensing for explicit and implicit interaction," in *The 11th International Converence on Human-Computer Interaction*, ser. HCII'05. Lawrence Erlbaum, 2005.

[13] G. Shoemaker, A. Tang, and K. S. Booth, "Shadow reaching: a new perspective on interaction for large displays," in *Proceedings of the 20th annual ACM symposium on User interface software and technology*, ser. UIST '07. New York, NY, USA: ACM, 2007, pp. 53–56.

[14] I. Poupyrev, S. Weghorst, M. Billinghurst, and T. Ichikawa, "Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques," vol. 17, no. 3, pp. 41–52, 1998.

[15] M. Kobayashi and T. Igarashi, "Ninja cursors: using multiple cursors to assist target acquisition on large screens," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 949–958.

[16] T. Grossman and R. Balakrishnan, "The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '05. New York, NY, USA: ACM, 2005, pp. 281–290.

[17] A. Cockburn, P. Quinn, C. Gutwin, G. Ramos, and J. Looser, "Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback," *Int. J. Hum.-Comput. Stud.*, vol. 69, no. 6, pp. 401–414, Jun. 2011.

[18] S. Zhai, W. Buxton, and P. Milgram, "The "silk cursor": investigating transparency for 3d target acquisition," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '94. New York, NY, USA: ACM, 1994, pp. 459–464.

[19] C. Randell and H. L. Muller, "The well mannered wearable computer," *Personal Ubiquitous Comput.*, vol. 6, no. 1, pp. 31–36, Jan. 2002.

[20] T. Martins, C. Sommerer, L. Mignonneau, and N. Correia, "Gauntlet: a wearable interface for ubiquitous gaming," in *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, ser. MobileHCI '08. New York, NY, USA: ACM, 2008, pp. 367–370.

[21] C. Harrison, D. Tan, and D. Morris, "Skinput: appropriating the body as an input surface," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 453–462.

[22] J. W. Summet, M. Flagg, J. M. Rehg, G. D. Abowd, and N. Weston, "Gvu-procams: enabling novel projected interfaces," in *Proceedings of the 14th annual ACM international conference on Multimedia*, ser. MULTIMEDIA '06. New York, NY, USA: ACM, 2006, pp. 141–144.

[23] C. Pinhanez, "The everywhere displays projector: A device to create ubiquitous graphical interfaces," in *UbiComp*. Springer Berlin Heidelberg, 2001, pp. 315–331.

[24] M. Ashdown and P. Robinson, "A personal projected display," in *Proceedings of the 12th annual ACM international conference on Multimedia*, ser. MULTIMEDIA '04. New York, NY, USA: ACM, 2004, pp. 932–933.

[25] K. Miyahara, H. Inoue, Y. Tsunesada, and M. Sugimoto, "Intuitive manipulation techniques for projected displays of mobile devices," in *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '05. New York, NY, USA: ACM, 2005, pp. 1657–1660.

[26] S. Seo, B. Shizuki, and J. Tanaka, "Clutching and layer-switching: interaction techniques for projection-phone," in *HFT2008: 21st International Symposium Human Factors in Telecommunication: User Experience of ICTs*. Prentice Hall, 2008, pp. 247–254.

[27] F. Echtler, M. Huber, and G. Klinker, "Shadow tracking on multi-touch tables," in *Proceedings of the working conference on Advanced visual interfaces*, ser. AVI '08. New York, NY, USA: ACM, 2008, pp. 388–391.

[28] A. D. Wilson, "Playanywhere: a compact interactive tabletop projection-vision system," in *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ser. UIST '05. New York, NY, USA: ACM, 2005, pp. 83–92.

[29] A. Greaves and E. Rukzio, "View& share: supporting co-present viewing and sharing of media using personal projection," in *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '09. New York, NY, USA: ACM, 2009, pp. 44:1–44:4.

[30] J. C. Lee, S. E. Hudson, J. W. Summet, and P. H. Dietz, "Moveable interactive projected displays using projector based tracking," in *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ser. UIST '05. New York, NY, USA: ACM, 2005, pp. 63–72.

[31] T. S. Saponas, D. S. Tan, D. Morris, R. Balakrishnan, J. Turner, and J. A. Landay, "Enabling always-available input with muscle-computer interfaces," in *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, ser. UIST '09. New York, NY, USA: ACM, 2009, pp. 167–176.

[32] P. Mistry, P. Maes, and L. Chang, "Wuw - wear ur world: a wearable gestural interface," in *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '09. New York, NY, USA: ACM, 2009, pp. 4111–4116.

[33] J. C. Platt, "Advances in kernel methods," B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, ch. Fast training of support vector machines using sequential minimal optimization, pp. 185–208.

[34] C. Harrison, H. Benko, and A. D. Wilson, "Omnitouch: wearable multitouch interaction everywhere," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 441–450.

[35] A. D. Wilson and H. Benko, "Combining multiple depth cameras and projectors for interactions on, above and between surfaces," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, ser. UIST '10. New York, NY, USA: ACM, 2010, pp. 273–282.

[36] C. Harrison, S. Ramamurthy, and S. E. Hudson, "On-body interaction: armed and dangerous," in *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, ser. TEI '12. New York, NY, USA: ACM, 2012, pp. 69–76.

[37] T. C. Alexander, H. S. Ahmed, and G. C. Anagnostopoulos, "An open source framework for real-ti incremental, static and dynamic hand gesture learning and recognition," in *Proceedings of the 13th International Conference on Human-Computer Interaction. Part II: Novel Interaction Methods and Techniques.* Berlin, Heidelberg: Springer-Verlag, 2009, pp. 123–130.

[38] S. Vinoski, "Chain of responsibility," *IEEE Internet Computing*, vol. 6, pp. 80–83, 2002.

[39] K. Gray, *Microsoft DirectX 9 Programmable Graphics Pipeline (Pro-Developer).* Microsoft Press, 2003.

[40] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1531–1536, Nov. 2004.

[41] M. Brown, A. Majumder, and R. Yang, "Camera-based calibration techniques for seamless multiprojector displays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 2, pp. 193–206, Mar. 2005.

[42] A. R. Weeks, C. E. Felix, and H. R. Myler, "Edge detection of color images using the hsl color space," *Proc. SPIE Nonlinear Image Processing*, vol. 2424, pp. 291–301, 1995.

[43] V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn, "Fingartips: gesture based direct manipulation in augmented reality," in *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, ser. GRAPHITE '04. New York, NY, USA: ACM, 2004, pp. 212–221.

[44] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine, "Image plane interaction techniques in 3d immersive environments," in *Proceedings of the 1997 symposium on Interactive 3D graphics*, ser. I3D '97. New York, NY, USA: ACM, 1997, pp. 39–43.

[45] A. Olwal and S. Feiner, "The flexible pointer: An interaction technique for selection in augmented and virtual reality," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, ser. UIST '03. New York, NY, USA: ACM, 2003, pp. 81–82.

[46] D. A. Bowman, D. B. Johnson, and L. F. Hodges, "Testbed evaluation of virtual environment interaction techniques," in *Proceedings of the ACM symposium on Virtual reality software and technology*, ser. VRST '99. New York, NY, USA: ACM, 1999, pp. 26–33.

[47] A. Forsberg, K. Herndon, and R. Zeleznik, "Aperture based selection for immersive virtual environments," in *Proceedings of the 9th annual ACM symposium on User interface software and technology*, ser. UIST '96. New York, NY, USA: ACM, 1996, pp. 95–96.

[48] L. J. and G. M., "Jdcad: A highly interactive 3d modeling system," vol. 18, no. 4, pp. 499–506, 1994.

[49] K. Yatani, K. Partridge, M. Bern, and M. W. Newman, "Escape: a target selection technique using visually-cued gestures," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 285–294.

[50] M. Nancel, E. Pietriga, and M. Beaudouin-Lafon, "Precision Pointing for Ultra-High-Resolution Wall Displays," INRIA, Research Report RR-7624, May 2011.

[51] S. Huot and E. Lecolinet, "Spiralist: a compact visualization technique for one-handed interaction with large lists on mobile devices," in *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*, ser. NordiCHI '06. New York, NY, USA: ACM, 2006, pp. 445–448.

[52] H. Benko, A. D. Wilson, and P. Baudisch, "Precise selection techniques for multi-touch screens," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '06. New York, NY, USA: ACM, 2006, pp. 1263–1272.

[53] D. K. O'Neill and J. C. Topolovec, "Two-year-old children's sensitivity to the referential (in)efficacy of their own pointing gestures," vol. 28, no. 1, pp. 1–28, 2001.

[54] G. Butterworth and P. Morissette, "Onset of pointing and the acquisition of language in infancy," vol. 14, no. 3, pp. 219–231, 1996.

[55] L. Camaioni, P. Perucchini, F. Bellagamba, and C. Colonnesi, "The role of declarative pointing in developing a theory of mind," *Infancy*, vol. 5, no. 3, pp. 291–308, 2004.

[56] R. A. Bolt, ""put-that-there": Voice and gesture at the graphics interface," *SIGGRAPH Comput. Graph.*, vol. 14, no. 3, pp. 262–270, Jul. 1980.

[57] D. Vogel and R. Balakrishnan, "Distant freehand pointing and clicking on very large, high resolution displays," in *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ser. UIST '05. New York, NY, USA: ACM, 2005, pp. 33–42.

[58] C. C. Kemp, C. D. Anderson, H. Nguyen, A. J. Trevor, and Z. Xu, "A point-and-click interface for the real world: laser designation of objects for mobile manipulation," in *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, ser. HRI '08. New York, NY, USA: ACM, 2008, pp. 241–248.

[59] A. Wilson and S. Shafer, "Xwand: Ui for intelligent spaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '03. New York, NY, USA: ACM, 2003, pp. 545–552.

[60] A. Wilson and H. Pham, "Pointing in intelligent environments with the world cursor," in *In the proceedings of Interact 2003*, 2003, pp. 495–502.

[61] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch, "Touch projector: mobile interaction through video," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 2287–2296.

[62] S. N. Patel, J. Rekimoto, and G. D. Abowd, "icam: precise at-a-distance interaction in the physical environment," in *Proceedings of the 4th international conference on Pervasive Computing*, ser. PERVASIVE'06, 2006, pp. 272–287.

[63] N. Pears, D. G. Jackson, and P. Olivier, "Smart phone interaction with registered displays," *IEEE Pervasive Computing*, vol. 8, no. 2, pp. 14–21, Apr. 2009.

[64] T. Seifried, M. Haller, S. D. Scott, F. Perteneder, C. Rendl, D. Sakamoto, and M. Inami, "Cristal: a collaborative home media and device controller based on a multi-touch display," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '09. New York, NY, USA: ACM, 2009, pp. 33–40.

[65] D. Sakamoto, K. Honda, M. Inami, and T. Igarashi, "Sketch and run: a stroke-based interface for home robots," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 197–200.

[66] H. Ailisto, J. Plomp, L. Pohjanheimo, and E. Strommer, "A physical selection paradigm for ubiquitous computing," in *Ambient Intelligence*, ser. Lecture Notes in Computer Science, E. Aarts, R. Collier, E. Loenen, and B. Ruyter, Eds. Springer Berlin Heidelberg, 2003, vol. 2875, pp. 372–383.

[67] K. Finkenzeller, *RFID Handbook, Radio-Frequency Identification Fundamentals and Applications*. John Wiley & Son Ltd, 1999.

[68] A. Wilson and H. Pham, "Pointing in intelligent environments with the worldcursor," in *Proceedings of IFTP Interact*, ser. Interact '03, 2003, pp. 495–502.

[69] M. Berkinblit, O. Fookson, B. Smetanin, S. Adamovich, and H. Poizner, "The interaction of visual and proprioceptive inputs in pointing to actual and remembered targets," *Experimental Brain Research*, vol. 107, pp. 326–330, 1995.

[70] L.-W. Chan, Y.-Y. Hsu, Y.-P. Hung, and J. Y.-j. Hsu, "A panorama-based interface for interacting with the physical environment using orientation-aware handhelds," in *The Seventh International Conference on Ubiquitous Computing, (UbiComp 2005)*, Tokyo, Japan, September 2005.

[71] M. Rohs, J. Schöning, M. Raubal, G. Essl, and A. Krüger, "Map navigation with mobile devices: virtual versus physical movement with and without visual context," in *Proceedings of the 9th international conference on Multimodal interfaces*, ser. ICMI '07. New York, NY, USA: ACM, 2007, pp. 146–153.

[72] Y. Uematsu and H. Saito, "Ar registration by merging multiple planar markers at arbitrary positions and poses via projective space," in *Proceedings of the 2005 international conference on Augmented tele-existence*, ser. ICAT '05. New York, NY, USA: ACM, 2005, pp. 48–55.

[73] J. Park and M.-H. Kim, "Smart pico-projected ar marker," in *SIGGRAPH Asia 2012 Posters*, ser. SA '12. New York, NY, USA: ACM, 2012.

[74] R. Ballagas, J. Borchers, M. Rohs, and J. G. Sheridan, "The smart phone: A ubiquitous input device," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 70–77, Jan. 2006.

[75] B. Lee and J. Chun, "Manipulation of virtual objects in marker-less ar system by fingertip tracking and hand gesture recognition," in *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, ser. ICIS '09. New York, NY, USA: ACM, 2009, pp. 1110–1115.

[76] J. a. P. Lima, V. Teichrieb, J. Kelner, and R. W. Lindeman, "Standalone edge-based markerless tracking of fully 3-dimensional objects for handheld augmented reality," in *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '09. New York, NY, USA: ACM, 2009, pp. 139–142.

[77] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch, "Touch projector: mobile interaction through video," in *Proceedings of the SIGCHI Conference on Human*

Factors in Computing Systems, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 2287–2296.

[78] J. Huber, C. Liao, J. Steimle, and Q. Liu, "Toward bimanual interactions with mobile projectors on arbitrary surfaces," in *In proceeding of: Proceedings of MP^2: Workshop on Mobile and Personal Projection in conjunction with ACM CHI 2011.* New York, NY, USA: ACM, 2011.

[79] S. Kim, Y. Cho, K. S. Park, and J. Lim, "Development of a handheld user interface framework for virtual environments," in *Proceedings of the 2nd international conference on Virtual reality*, ser. ICVR'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 253–261.

[80] N. Villaroman, D. Rowe, and B. Swan, "Teaching natural user interaction using openni and the microsoft kinect sensor," in *Proceedings of the 2011 conference on Information technology education*, ser. SIGITE '11. New York, NY, USA: ACM, 2011, pp. 227–232.

[81] Y. Yamamoto, I. Yoda, and K. Sakaue, "Arm-pointing gesture interface using surrounded stereo cameras system," in *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4 - Volume 04*, ser. ICPR '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 965–970.

[82] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 559–568.

[83] H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz, and D. Fox, "Interactive 3d modeling of indoor environments with a consumer depth camera," in *Proceedings of the 13th international conference on Ubiquitous computing*, ser. UbiComp '11. New York, NY, USA: ACM, 2011, pp. 75–84.

[84] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principle and Practice in C (2nd Edition).* Addison-Wesley Professional, 1995.

[85] G. Strang, *Introduction to Linear Algebra, Fourth Edition.* Wellesley Cambridge Press, 2009.

[86] T. Meilinger and G. Vosgerau, "Putting egocentric and allocentric into perspective," in *Proceedings of the 7th international conference on Spatial cognition*, ser. SC'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 207–221.

[87] R. Stoakley, M. J. Conway, and R. Pausch, "Virtual reality on a wim: interactive worlds in miniature," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '95. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 265–272.

[88] A. Mulloni, H. Seichter, and D. Schmalstieg, "Indoor navigation with mixed reality world-in-miniature views and sparse localization on mobile devices," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ser. AVI '12. New York, NY, USA: ACM, 2012, pp. 212–215.

[89] D. A. Bowman, E. Kruijff, J. Joseph J. Laviola, and I. Popupyrev, *3D User Interfaces Theory and Practice.* Addison-Wesley, 2005.

[90] R. L. Klatzky, "Allocentric and egocentric spatial representations: Definitions, distinctions, and interconnections," in *Spatial Cognition, An Interdisciplinary Approach to Representing and Processing Spatial Knowledge.* London, UK, UK: Springer-Verlag, 1998, pp. 1–18.

[91] D. C. Geary, S. J. Saults, F. Liu, and M. K. Hoard, "Sex differences in spatial cognition, computational fluency, and arithmetical reasoning," *Journal of Experimental Child Psychology*, vol. 77, no. 4, pp. 337 – 353, 2000.

List of Publications

Journals

Seokhwan Kim, Shin Takahashi, Jiro Tanaka, "Point-Tap, Tap-Tap, and The Effect of Familiarity: to Enhance the Usability of See-and-Select in Smart Space", Transaction of Human Interface Society, Volume 14, No. 14, Human Interface Society, pp. 445-455, November, 2012.

Conference proceedings

Seokhwan Kim, Shin Takahashi, Jiro Tanaka, "New interface using palm and fingertip without marker for ubiquitous environment", 9th IEEE/ACIS International Conf erence on Computer and Information Science, pp. 819-824, 2010.