RWTH AACHEN UNIVERSITY

A Classification of Interaction Styles that Span Multiple Systems

Master's Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Ying Zhang

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Irene Mittelberg

Registration date: 27th October, 2011
Submission date: 29th May, 2012

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, May 2012
Ying Zhang

Contents

List of Figures

List of Tables

Abstract

With the development of techniques and the growing interest of user-centered design, more and more new ideas of advanced interaction technologies are proposed and implemented. And the use of Multiple Display Environment (MDE) makes collaborations and information exchange more efficient. However, when heterogeneous portable devices are involved, how to choose a suitable technique to meet users' requirements becomes an issue. In fact, only few literatures present classifications about cross-device interaction techniques. The existing ones are not complete enough to describe the major properties of each technique, especially from the capability aspect. In this thesis work, we present a design space with 10 dimensions and the corresponding graphical representations, which are sufficient to depict the properties of the existing interaction techniques. In our design space, the properties of input methods, the level of utilizing spatial information of the involved devices, the control mechanism, and parallelism aspects are of most concern. Moreover, 13 major techniques supporting cross-device object relocation tasks are introduced, and each of them is fitted in the design space. The graphical representation visualizes the distinctions between techniques, and indicates a comprehensive and intelligent classification. The dimensions we identified reflect user preference of the selected techniques and have different influences on users' performances. Among them, some dimensions are context dependent. These hypotheses have been proved by our evaluation.

Überblick

Mit der Entwicklung von Technologien und wachsende Interesse an User basierte Entwurf, immer mehr Konzepte von fortgeschrittene Technologien sind entwickelt und implementiert. Und die Anwendung von Multiple Display Enviroment (MDE) hat die Effektivität von Zusammenarbeit und Datenaustausch deutlich erhöht. Allerdings es ist nicht einfach festzustellen welche Technik ist optimal für ein bestimmt Anwendung wenn heterogenische mobile Geräte sind involviert. In der Literaturen nur ein beschränkt Menge von Klassifizierung der Interaktionstechniken ist zu finden. Vorhandenen Techniken sind nicht befähigt für eine vollständige Beschreibung der Hauptmerkmale von einzelner Technik, besonders die Kapazitätsmerkmale. Wir haben ein Design Space mit 10 Dimensionen und dazugehörige graphische Darstellung entwickelt, der die Eigenschaft von existierenden Techniken erfolgreich beschreiben kann. In unsere Design Space, grose Werte ist auf die Eigenschafte von Inputmethoden, das Anwendungsniveau von Information der involvierten Geräte, der Mechanismus von Kontrolle und Synchronisation gelegt. Außerdem werden 13 Hauptmethoden, die das Objektpositionieren bei Multigeräte unterstützt, vorgestellt und in die Design Space klassifiziert. Die graphische Darstellung hat die Unterschiede zwischen Techniken visualisiert und klassifiziert. Die von uns gewählten Dimensionen reflektieren die Bevorzugung des Users und gleichzeitig beeinflusst die Performance von User. Unter diese Dimensionen, einige davon sind Kontext relevant. Diese Hypothese ist auch bewiesen in unsere Bewertung.

Acknowledgements

It is a pleasure to thank those who made this thesis possible:

I must first express my gratitude towards Prof. Jan Borchers for offering many wonderful and interesting courses which lead me to the HCI field. I would like to thank my advisor Jonathan Diehl, who offered me this interesting thesis topic, and guided me to develop an understanding of the subject throughout the work. Without your guidance, patience and support I can't complete this work. Many thanks to Chat Wacharamanotham. You gave me a lot of ideas and suggestions especially in the initial level of this work and guided me in aspects from theory and user test design. I also thank Sukeshini Grandhi who advised me in understanding the purpose of evaluation from social computing aspect. I learned so many things from all of you in this work.

Special thanks to some of my fellow master students and friends: Can Liu, Yang Yang, Siyuan Zhang. You gave me a lot of support and helped me when I have difficulties.

Thanks to all my friends and students participated in my user test. I appreciated your patience and support. Finally, I thank my parents for supporting me to finish the study in Germany.

Conventions

Throughout this thesis we use the following conventions.

The names of interaction techniques, sub-dimensions, special widgets or devices, are written in italic characters.

The whole thesis is written in American English.

"He/she"is used to describe the unidentified third person.

The plural "we" will be used throughout this thesis instead of the singular "I", even when referring to work that was primarily or solely done by the author.

Chapter 1

Introduction

Classification of interaction techniques plays a critical role in the research field of human computer interaction. It provides a better understanding of the previous works and a criteria of evaluating various techniques. With different design disciplines and dazzling innovations, modern interaction techniques have more and more new features and are developed rapidly. For input methods, more choices as pen/stylus, finger-touch screen have been offered, mouse and keyboard are no longer the only options. It is the problem now that how to make the choice of the proper technique under certain situation, and has been focused and discussed intensively.

Heterogeneous devices and diversity of today's interaction techniques bring the choosing problem.

Much more efficient work environment has been provided by multiple displays with different sizes and form, as well as heterogeneous peripheral devices. The concept of MDE (Multiple Display Environment) has been brought up to describe an integrated workspace, in which multiple display devices are connected to the system through certain methods (e.g. wire or wireless networks, infrared connection, etc.). The workspace that we discussed here is more like a logical concept, i.e. the involved devices don't have to be co-located. They belong to the same workspace by supporting users in accomplishing a common task.

Introduction of MDE (Multiple Display Environment)

MDEs are commonly applied for meetings and collaborative group work. Data exchange is essential in those pro-

Object relocation task is defined as fundamental task

cesses. And extracting and transferring digital information from other computers or heterogeneous portable devices are ubiquitous and critical. Considering this requirement, in this work, we will discuss the characteristics of some selected interaction techniques when the fundamental task of MDEs is performed. Here, the fundamental task refers to relocating a digital object from one work surface to another, or within a large display. The surfaces could be operated by one system (one computer, multi-displays), or several systems (multi-computers, multi-displays). Cross-display object movement is one of the basic functions enabled in MDEs. There are many existing interaction techniques supporting this kind of task. To describe each technique we analyze them with a series of dimensions, and each interaction technique can be represented in a visualized design space.

Comparing with previous classifications, which focus on discussing properties of a technique from performance or other single aspect , we are interested in its overall capabilities.

Previous works of classifications have focused on the performance, such as the accuracy and task completion time[Nacenta et al., 2005], The description of gestural interaction technique, such as the context of using different type of gestures[Karam and m.c.schraefel, 2005], and collaboration issues such as group awareness. In this work we are interested in the overall capabilities of the selected interaction techniques. Precisely, what they can exactly do when performing the cross-display object movement task. However, some newly designed techniques are hard to fit the criteria defined in the existing taxonomies. Thus we reused several dimensions in the article of Nacenta's [2009], some of which with a different view point, and explored more by reviewing the new techniques in recent years.

The concept of design space is inspired by Card's [1990]work. We further modified and rearranged the express method of the design space by using the newly explored dimensions. It will improve and expand the existing taxonomies.

Our approach of classification

Our classification is based on three groups of dimensions: 1) what the initiation methods are and how do they work (input method, positional mapping, replace-ability of input device, and power of working area); 2) how displays are referred to the systems and users, as well as the control paradigm (referential environment, input model type, feed- forward, and feedback); 3) how the identification is managed during the parallel operations (parallelism, and identification). The design space is sketched according to those dimensions, using notations of lines and nodes. The property of each selected interaction technique will be represented in 3 2-dimensional tables. This representation will visualize the characteristics of the selected interaction techniques, and can further assist the researchers and designers to create new techniques in MDEs.

Our work reviewed major interaction techniques up to now and complemented the existing criteria, thus newly designed interaction techniques can also be added to the multi dimensional design space. Besides, by transforming qualitative problems into quantitative measure, our evaluation provides a new way to gauge user preference of a technique influenced by various dimensions. Here is the overview of the thesis and brief summary of each chapter:

The contributions of this work

Chapter 2—"Related work" reviews previous work that related to the topic of design space, interaction techniques taxonomy, and theoretical fundamentals about cross-devices interaction.

Chapter 3—"Design space" introduces each dimension and its sub categories, presents the design space and the notations. In addition, a comprehensive look of the chosen interaction techniques is also summarized in this chapter.

Chapter 4—"Techniques supporting cross-device object movements" describes the capabilities of each technique in detail according to our dimensions. For each technique, alternative techniques that can be used for accomplishing the same task based on our design space are suggested.

Chapter 5—"Evaluation" presents the hypotheses and our approach to verify them. In this chapter we describe the design of our user test, the methodology of analyzing qualitative data, and the results of our experiments.

Chapter 6—"Summary and future work" concludes the work we have done and the contributions of this thesis, and discusses the possible research directions according to our findings and work can be done in the future.

Chapter 2

Related work

The related work can be subdivided into the following parts: 2.1 classifications, 2.2 design space, 2.3 cross-device interaction techniques.

2.1 Classifications

There are only a few papers about taxonomy or classifications of interaction techniques. The paper of Karam [2005] provides taxonomy of gesture-based interactions over the last 40 years. The techniques are categorized into four groups: gesture styles(the forms of human gesturing as interaction technique), the application domains (the use context), enabling technology (the forms of input) and system response (the forms of output). The authors provide us an overview of the field and their work inspires us much about the way of exploring new categories. We also consider the forms of input/output as very important dimensions. However, this paper is focused on the gestured based concept. We are interested in interaction techniques that are initiated by all kinds of input methods, and focus our attention on cross-device interactions, especially in MDEs.

Taxonomy of gesture-based interactions has been presented by Maria Karam et al.

In the report of Nacenta's [2005] a comparison of techniques for remote display accessing is presented. The involved devices are still co-located, i.e. the furthest destina-

Nacenta et al. evaluate and compare six techniques for co-located multi-displays reaching.

tion is within the visual horizon. In the research users are required to put digital objects in remote destinations with different techniques. Six interaction techniques are evaluated according to nine dimensions: topology of the underlying interaction space (the level of utilizing virtual conception), reaching range (the furthest accessing range), nature of the destination display (the importance of the specific location for the destination), input device, feedback, display and input area requirement (the required area for input space), privacy rules (the level of information sharing), one-sided vs. two-sided techniques (one-sided: being sender or receiver only; two-sided: being sender and receiver), and symmetry (the reversibility of pointing direction).

Using Nacenta's [2005] dimensions

Those dimensions are mostly derived from the task requirement, and what interests us the most is the concept of utilizing virtual conception (corresponding dimension: topology of the underlying interaction space). It indicates the relation between physical space and its digital representation. This dimension is sub-categorized into 4 types: virtual space, coupled virtual and physical space, physical space, and discrete. We consider these sub-dimensions as different levels of utilizing spatial information. In our work, colocated workspace is an important use context, and the idea of utilizing spatial information brings users efficiency of aiming the destination device in such context. We use this dimension as well as its sub-dimensions, and classify them into two groups. We rename the dimension as 'referential environment' and its definition will be introduced in 3.1.5. In addition, we decide to combine the concept of 'reaching range' and 'the display and input area requirement', thus the new dimension 'power of working area' (in 3.1.4)is created.

Nacenta's taxonomy [2009] provides us fundamental knowledge in this filed and several important dimensions that we can use in our classification.

The most important work related to this paper is the taxonomy of Nacenta et al.'s in [2009]. This work provides us the conceptual vocabulary and a comprehensive classification of the field, and surveys many kinds of interaction techniques supporting cross-display object movement task. We use the modified dimension concept of input model type (mapping between physical arrangement and digital representations), control configuration (the capability of controlling the final object position), and referential domain (the level of utilizing spatial information), to fit in our situation. Besides, by reviewing the recent developed techniques, we explore more dimensions that help to better describe their characteristics and nature. Control configuration is renamed as feed-forward to echo the feedback, because the feed-forward is determined by feedback. The definitions of our dimensions will be introduced in detail in 3.1.

2.2 Design space

The design space analysis aims at helping developers and designers reason about design, systematizing the properties of interaction techniques and visualize the interrelationship between them, thus helping others to understand the design aspect of each interaction technique in nature.

Introduction of Card's design space

The fundamental knowledge and representation method of the design space is derived from [Card et al., 1990]. In their design space, nodes are used to represent physical properties of the input devices; three kinds of lines represent the corresponding composition types between properties: the black line, dashed line, and double line with arrow stand for merge, layout and connection composition respectively. Taking the mouse as an example in Figure 2.1, the two nodes and the black line connecting them represent that the mouse can be moved along the x- and y-axis; the node labelled with 3 and the dashed line connecting the node with others, stand for the composition of three buttons.

Comparison between Card's design space and ours

The abstraction of Card's [1990] design space is at low levels, which tend to describe the properties of an entity from hardware aspect. In contrast, our design space is at high levels, with which relatively more attention has been paid on how to accomplish tasks. Card's dimensions are not complicated, thus the design space can be represented in a single 2-dimensional table. Our design space is more complex. It describes the properties of an interaction technique from 3 main aspects: the choice of input method, the control mechanism as well as the referential method, and the parallelism. We decide to use three 2d tables to represent

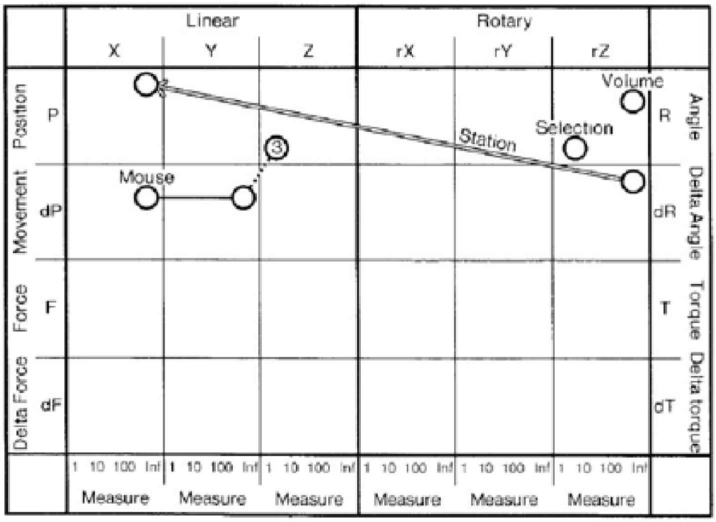


Figure 2.1: Card's design space of input device. [Card et al., 1990]

the full properties of an interaction technique.

2.3 Cross-device interaction techniques

Clarifying the definition of 'cross-device' and 'medium device'.

Strictly speaking, the term 'cross-display' is not equal to 'cross-device' interaction, although in many technical reports they have no obvious difference because of the equipped screens of most devices. The term 'device' here refers to the storage location of the source or the destination of the digital object to be moved. The tools used for transmission as intermediate is called transmission medium device, or medium device. Consider of involvement of no screen devices, such as the movable storage or other physical medium devices, we use the term cross-device instead. And MDE is a subset of cross-device environments.

Communicating with a collection of heterogeneous computing devices is very different from interacting with a single computer. Early in time the researchers dedicated in the study of exploring ways of connecting and integrating multiple devices. With the development, the concept of human-centered design gains more popularity. How to use the knowledge about the physical environment that the user already has to enable intuitive human-computer communications and how to reuse the physical skill to manipulate a digital object, have attracted attentions of interaction techniques designers and developers. In this section we will discuss how those challenges are met by the previous work.

Challenges of the cross-device interaction techniques: devices integration and use of the real world knowledge.

The connection between two devices can be established in several ways. For single computer operating system, monitors and other displays can be connected by cables or wireless network. However, more complex system is commonly used, which involves several devices and can scale more easily. The so-called meta-operating system [Smarr and Catlett, 1992] utilizes network connection and middleware to realize cross-device interactions. With these connecting method, two problems arise. The first problem is the dependency of the physical proximity. Connecting methods that utilize cables, wireless LAN, infrared beaming restrict users within a certain range. These methods are applicable to meetings, in-door collaborative work, where the involved portable devices are usually co-located and the participants gather around. The second problem is the so-called referential problem. For example, a user can choose a target device by selecting its name from a list. In the list, all the connected and identified devices within the network are represented by names (e.g. Linda's computer, the iPad of lab room 2U02). However, in spite of indicating belonging relation, those names usually can't depict any other characteristic of the represented devices, or provide useful information to help users better distinguishing one device from the other. The referential problem confuses users and requires extra effort for remembering the names and corresponding devices.

Different connecting methods and the problems along with them

To solve the dependency problem, the online storage service, or instant messaging service could be the solutions. Recently cloud storage techniques are increasingly in popularity. They utilize the accessibility of internet and the cloud server, with the installed client software in supported devices, a user can access the same files among different devices, examples of this kind of service include i-cloud, dropbox, etc. Instant messaging based techniques also can

Solve the dependency problem

be categorized as service based techniques because of the dependence of a central server. However, such techniques realize the independence of devices and user location.

Solve the referential problem

To solve the referential problem, more and more interaction techniques with intuitive connecting and identifying methods are developed. As a result, the concept of utilizing spatial information and physical skills is proposed. In the following parts of the work, the term 'spatial information' stands for the absolute location of a device or its relative position to the user or to other devices.

Pick-and-Drop attempts to improve human-computer interaction method by using spatial information and physical skills.

The interaction technique *Pick-and-Drop*, developed by [Rekimoto, 1997], is one of the earliest approaches that made attempts in communicating with the digital system by using the spatial information and physical skills of the real world. *Pick-and-Drop* allows the user to 'pick up' a digital document by tapping it on the source surface with a special pen, and drop it on a different device surface by tapping it again with the same pen. By mimicking the action of picking up things in the real world, the user can get the impression of picking up a digital object through the pen instead of the virtual network. This physical interfaces concept draws attentions and is expected to be a good solution of the referential problem. The visibility of the interface and the physical manipulation method allow users to access and transfer the digital object in a quick and easy manner.

In fact, the idea of using physical skills or knowledge of the real world is not limited to the direct manipulation method. It's easier to handle a new technique based on the knowledge of real world and skills of manipulating real object.

Different ways of utilizing spatial information

For cross-device interactions, there are plenty of ways of using spatial information. Some techniques make use of it by have a digital map, which represents the involved devices in the real world according to their relative positions. For example, interaction techniques that utilize a miniature map, such as *Radar View* and *ARIS*, offer users an overview of devices' deployment. From the map users are kept informed of the spatial relation between two devices, thus are able to pick up the target among a mass of devices easily. Other techniques use physical proximity to establish connections. Those connected devices usually equipped with sensors to detect the distance to other devices. If they are close enough the binding of two devices becomes possible. Blue-tooth technique, *SyncTab*, and *Synchronous Gestures* techniques [Hinckley, 2003] are of this kind. The third kind of using the spatial information is easier to understand, i.e. virtually or physically access the target devices as where it is. For example, some techniques extend the user's control range, such as making the cursor move across different displays, e.g. *Hyperdragging*, *PointRight*. Here we take the classic *Pick-and-Drop* as an example again, which fully makes use of the original spatial information of the real world. We consider the problem of how to use the spatial information as a very important dimension in our design space. More definitions will be introduced in details in Chapter 3

Chapter 3

Design space

In this paper we focus on studying and analyzing those interaction techniques that support the task of cross-device objects movements. The term 'object' refers to any digital information form that can be relocated to a new destination. It could be an icon of a digital file or application, or a running application window. Throughout this paper, unless otherwise specified, the term 'object' and 'digital object' stand for the same meaning. The aim of the object relocation task is to transfer a digital object or to migrate a running application window to a virtual port or a certain position on the specific screen. The movement can take place between different displays operated by one system, between different operating systems with multi-screens, or on a large screen.

Clarifying the definitions of 'object' and 'relocation task'.

3.1 Dimensions of interaction techniques

Dimensions are terms we used to describe the properties of an interaction technique. By reviewing literatures concerning taxonomy of interaction techniques and exploring from the technical reports, we defined the following dimensions and standardize some sub-dimensions.

3.1.1 Input method

Mouse-cursor/Pen- based/Finger and touch screen

Button-based

Token-based

vision-based

Tracking system/Motion sensing/Sensor based techniques

This dimension refers to the method of initiating an interaction technique for the kind of task we defined above. The choice of input method is influenced by many factors, such as the nature of involved input devices, the way of using them, the modality of input, and the properties of the object to be moved.

Mouse-cursor/Pen-based/Finger and touch screen

The first category includes commonly used pointing devices, which provides positional information to the system. The most widely used traditional input method *Mouse-cursor*, which can be directly used or as an alternative in many kinds of interaction techniques. Strictly speaking, for *pen-based/Finger and touch screen*, the surface that can sense the position of objects above it is also a part of the tools of enabling techniques. Thus the touch screen and finger/stylus above it can all be considered as input devices. The difference between using a stylus or finger lies in two aspects: first, the pen or stylus enables more precise taps on the touch screen, especially when the input area is small or the digital objects are densely distributed on the screen. Second, some gesture commands require multi-fingers action, which is hard to simulate by a stylus. The mapping property of pointing will be further introduced in the next dimension.

Button-based input method includes the traditional computer keyboard, the keypad on mobile phone, or other devices equipped with special functional keys. In our context, this kind of input devices enables textual command input, direction key control, or transferring command directly to the system after pushing the functional button.

Token-based access means to use objects in the real world as physical representatives of digital information, i.e. use physical object (token) to access the digital information that stored outside the object [Holmquist et al., 1999], e.g. the technique *passage* used in *i-Land*, allows user to assign digital information to arbitrary physical object via a special device *bridge*. By using the physical selection [Valkkynen, 2008], the web link can be represented by a tagged physical object and recognized by digital terminal, such as a RFid (Radio-Frequency identification). The token can be further classified into 3 types: container (the properties of the physical object don't affect the activity), token (the physical shape of the token indicates some properties of the represented data), and tool (the properties of tool affects the operation).

Token-based

Vision-based refers to the input method that enables image as input captured by camera. For example, the interaction technique deep shot [Chang and Li, 2011], uses the screenshot of the source device captured by a mobile phone camera as input, and utilizes the visualized information to realize status migration.

The last group of input methods includes techniques that apply various sensors in several ways. *Tracking system* refers to those interaction techniques that utilize sensors to detect the presence or movement change of other devices or objects. The interaction technique *TractorBeam*, uses the Polhemus Fastrack together with a corded stylus to realize a direct remote pointing [Parker et al., 2005]. Such system is a six degrees of freedom tracking system and can continuously track the position of the optical sensor that emits a beam. *Motion sensing* techniques use webcams together with sensors to detect user's whole body movement in 3d. Kinect sensing technology enables the tracking of user's motion without any help of sensor embedded device (such as game controllers) in user's hand or attached to the user's body. *Kinect* sensor computes a depth map through the infrared projector and CMOS (Complementary Metal Oxide Semiconductor) sensor. The user's position can be inferred with the help of machine learning. Comparing with the *vision-based* input method, this kind of method utilizes mounted cameras for environmental detection, and is able to track 3d motions and orientations with full artic-

Different use of sensors

ulation. In addition, the techniques can be classified into *sensor-based*, if sensors are involved during the relocation task and are served as initiation tools in some other ways. For example, with *Synchronous Gesture* the tablets to be connected are equipped with touch sensors on the edges to detect physical proximity.

3.1.2 Positional mapping

Absolute input

Relative input

Rate-controlled input

Many interaction techniques are initiated by pointing devices, which provides positional information to the system. The pointing devices can provide positional data by itself, such as a sensor or a computer mouse, or its position can be sensed directly by the surface. E.g., when using *finger on a touch screen*, the absolute position of contact point is sensed by the surface. The *positional mapping* refers to the mapping between the user's action and positional information of the device, which can be categorized into *absolute input*, *relative input*, and *rate-controlled* input.

Absolute input

Absolute input means that the position of the contact point of input device, either physical or virtual, is directly sensed by the system. For example, when using a stylus on an interactive surface, the contact point of the stylus is exactly mapped into the positional information of the input. The laser pointer is another example, which indicates that the absolute input device does not always have physical contact with the surface. This kind of input is usually used when the users want to make use of their hand to have full control the object, such as the especially when the form of the task is to mimic a real world activity, for example, to *Drag-and-Drop* an digital object using finger on a touch screen is similar to drag an object along the physical surface using hand in real world. Because of the property of direct manipulation, the interaction techniques using absolute input require a certain range of working area, which becomes another dimension and will be introduced in 3.1.4.

On the other hand, *relative input* refers to the different mapping principle. The positional information of the input is determined by the offset of the input device's position. The mouse is a typical relative input device. The small movement of the mouse can be mapped into a larger movement of the pointer. Thus the techniques using this kind of input may benefit from the small working area requirement comparing with the one above.

Relative input

Rate-controlled input device such as a wheel mouse, maps the user motion to cursor velocity [Zhai et al., 1997]. Usually this kind of mapping is compatible with elastic or isometric devices, e.g. the elastic joystick. However, other input device such as standard mouse or pen can also apply the rate-controlled mapping. For example, with the gesture *Flick*, the velocity of the pen becomes the control of strolling rate.

Rate-controlled input

3.1.3 Replace-ability of input device

Dedicated device

Alternative devices

Compatible devices

Use anything as input

Sometimes different input devices can simulate parts of or whole functionality of others, thus the input device sometimes can be replaced by alternative ones under some conditions. Then we have the four levels of the replace-ability as follow:

Dedicated device means only specific input device can be used. At this level the type of the input device can't be directly replaced by other devices. Usually this kind of input device is responsible for some special tasks. E.g. some dedicated sensors are hard to replace because of their specific purpose of use. E.g. the thermometer is this kind of dedicated sensor.

Alternative device means under some conditions, such as by installing the same software, several kinds of input devices can become the alternative choice to initiate an interaction technique. Taking the example of *Pick-and-Drop*, the interaction is initiated by a special interactive pen, and each pen has a pen id as the unique identifier. The normal pen can replace the interactive pen by making some design modifications, such as assigning each pen an id and installing the recognition software on the system.

Compatible devices means that for some interaction techniques, diverse input devices can be used, as long as they have the same functional for what this technique requires, e.g. for some throwing-based techniques [Holmquist et al., 1999] (*Superflick, Push-and-Throw, Drag-and-Throw*, etc.), the input device could be a mouse, pen/stylus, or other pointing device.

The category *use anything as input* is derived from the concept of *Passage* used in i-Land, where any physical object that has certain weight can become the input device. The only identifier is the unique weight of the object and can be distinguished by the device bridge. The transmitted data is assigned by the bridge connected to the source computer, and can be read by the bridge that has connection with the destination device. This category is closely related with the *token-based* access input method in our first dimension. The container (one of the subclasses of token) is a typical example of this category.

3.1.4 Power of working area

Within hand's reach

Within arm's reach

Beyond arm's reach

The dimension 'power' used in Nacenta's taxonomy [2009]

The meaning of 'power'

refers to the distance between the user's hand and the furthest possible destination he/she can virtually reach using a specific interaction technique. The further the destination is, the bigger the power this interactive technique has. E.g., the technique using planar or perspective input model has high power, while the literal technique has limited remote power. We reuse the concept but with a different viewpoint: instead of describing the distance we are interested in the range of user's motion.

Within hand's reach refers to the working area where user's hands have full control of. *Within arm's reach* means the user need to physically or virtually (e.g. *TractorBeam*) reach the position by stretching arms or waist, but his/her feet still don't move. *Beyond arm's reach* refers to the remote position where the user can't reach by hands until he/she change his/her physical location. *Pick-and-Drop* is the technique of this type. The user needs to physically move to the destination position if it is far away.

It's to be noted here that it makes more sense to use *within hand/arm's reach* to distinguish the operation range of those techniques with absolute input. E.g. when interacting with a large touch screen, the operation may require physically reaching a position which is relative further than the one of *within hands' reach*, i.e. the user must stand up (assume that the users are sitting around the table) and stretch arms or the waist, which consume much more effort than just sitting there and pointing.

3.1.5 Referential environment

When an user wants to move an digital object to another position, he/she needs to express this intend by referring the destination position. The referring could use the information of access paths, or spatial information of the screens. This dimension refers to the different referring methods concerning spatial information and indicates to which extent we need to relate the physical environment to its virtual representation. There are two groups of the referential environment, non-spatial and spatial, each of which can be further subdivided into two categories.

Non-spatial:

Discrete/no topology

Virtual space

Non-spatial topology means that the representation of environment dose not need to be mapped to the physical world . *Discrete or no topology* means different displays need to be virtually distinguished regardless of their specific physical locations. For examples, the instant messaging based techniques distinguish different destinations through uniform resource identifier. A user only needs to know the receiver's virtual address. On the other hand, in order to sense the presence of the target device, some techniques utilize physical proximity as auxiliary to realize channel establishment, such as SyncTab, Bluetooth technique, etc. The topology between devices defined by this kind of techniques does not rely on the spatial information so should be categorized here.

The *virtual space* could be the virtual reality that simulates the environment from real world, or complete imaginary objects or interfaces. The representations of the source or destination devices could be a viewport and are not associated with their position in real world . Examples of such technique include the *virtual meeting room*, etc.

Spatial:

Coupled virtual and physical space

Physical space

Spatial topology is widely used in the MDEs when the involved devices are co-located. *Coupled virtual and physical space* refers to the interaction techniques that keep track of the involved devices' positions or keep a topology map on the server which records the physical location of each screen. The maintenance of such system needs dynamic hybrid combination. The *Sketch Radar* [Aliakseyeu and Martens, 2006], for example, is characterized as spatial topology when performing the object relocation task. However, when it adds new devices to the map and establishes connections by scanning bar-codes, is non-spatial. When the map is initiated at the beginning, it is based strictly on the physical location of the screens or other devices. But the user can change the layout on the map, which changes the physical correspondence .

Physical space is easy to understand because it is our surrounding environment in the real world. When interacting with physical screens or other physical objects, no virtual representation of them is needed. Such techniques include *Pick-and-Drop*, tangible interaction techniques, etc.

We will discuss the types of representation of the spatial referential environment in next dimension.

3.1.6 Input model types

Planar

Perspective

Literal

This dimension refers to the mapping between the physical arrangement and the virtual space, which is derived from Nacenta, M.A's taxonomy [2009]. When the environment is spatially referential, the MEDs are defined by their display configurations, which are influenced by the input model type to great extent.

Planar

An input model is *planar* when multiple displays are represented within a two-dimensional plane. It is easy to implement and scale. However, the representations of devices and displays are usually oversimplified, and users may have perspective problems for realizing which representation stands for which device. When more non-aligned devices are added to the workspace, the problem becomes more serious. Examples of this category include *Slingshot*, *PointRight*, *Radar View*, etc.

Perspective input model type makes use of the position knowledge of the user and the spatial representation of

Perspective

the whole workspace to provide intuitive layout from the user's point of view. This kind of model type provides the user a view of the environment as a first-person camera. Just like the experience in the 1st person shooter game. The view changes when the player moves according to the perspective law. Examples include *Perspective Cursors*.

Literal

The input model is *literal* when the interactive technique relies on the physical environment, and does not need any virtual representation of the displays. Typical examples are *Pick-and-Drop*, *Synchronized Gestures*, and *Tangible Bits* [Ishii and Ullmer, 1997].

3.1.7 Feed-forward

Open-loop

Closed-loop

Open-loop

The term 'feed-forward' refers to the control ability of the object's destination. It works together with the next dimension: feedback.*Open-loop* refers to the interaction techniques that are lack of feedbacks, which can indicate the final position of the object before it reaches there. And no further adjustment can be performed while the object is still moving. Usually a single gesture is used, e.g., *Flick* is such kind of technique.

Closed-loop

On the other hand, *Closed-loop* control enables the user to adjust the execution of his/her action according to the feedback until the action is finished. E.g., the input device of *SyncTab* are the connection buttons, before the file transfer start,a message will be shown on the screens to indicate that the connection is successful established. This feedback makes sure that the file being moved will be sent to the right destination. For mouse-cursor enabled techniques, the moving cursor acts as the continuous feedback, so that the user can adjust the mouse movement in real time. The interaction technique *Pick-and-Drop* is also *closed-looped*. The user has full control of the object's position because of the *absolute* input. More examples include *Superflick*, *TractorBeam*, etc.

3.1.8 Feedback

This dimension distinguishes the types of feedback provided by the system according to the modality of the output.

Visual feedback

Audio feedback

Haptic feedback

The devices that provide *visual feedback* in 2-dimension include different kinds of screens, projected displays and portable mobile devices. The modality of *visual feedback* could be a cursor, a dialog box, digital lines, etc. In Maria Karam's [2005] taxonomy, *audio feedback* is recommended as an assisted method to avoid distracting the user's visual focus. *Haptic feedback* such as vibration is widely used on mobile phones.

3.1.9 Parallelism

Multi-user parallel operation

Multi-user serial operation

Single user parallel operation

Single user serial operation

Multi-user parallel operation and *single user parallel operation*

This dimension refers to the possibility of the parallel operation of multiple interactions, which include multiple users' multiple interactions and single user's multiple interactions. *Multi-user parallel operation* means more than one user's operation can be performed in one interaction. E.g. the interaction point used in *Dynamo*, pointers from different users can interact with the system simultaneously. *Single user parallel operation* means to one user, he/she can initiate more than one file transfer task at the same time. Strictly speaking, the initiation actions should happen one by one because of the limitation of human's focus. However, if one user can initiate another transfer task before the transmission process of the previous one finishing, we consider such interactions as parallel.

Multi-user serial operation and *single user serial operation*

Some techniques only allow serial operation, otherwise collisions may happen. For multi-user case, *syncTab* for example, one interaction can be initiated at a time. Collisions happen when more than two connection requests are sent at the same time. More complicated examples include the clipboard systems, where different user can transfer files to different destinations at the same time, but from the point of view of a single user, the operation is serial.

3.1.10 Identification

The identification problem comes with multi-user parallel operation, especially in a shared work space. It is necessary to distinguish the ownership of cursors or any other input devices when multiple operation from different users happens simultaneously. In this dimension we discuss about the identification methods of different users and devices from two aspects: how the system distinguishes the ownership of devices, and how users differentiate their own input signals from others. Therefore we further divide this dimension into the following two groups.

Distributed user ID

Centralized user ID

Device id (no user ID)

The sub-dimensions within the first group present three solutions of the system identifying problem. In [Pato and Rouault, 2003] the *distributed user ID* is defined as follow: for multiple users the information of their identities can be assigned on different systems or server respectively, and the identity information can be exchanged across more than one trusted domains. For such identification method, no central user ID management mechanism is required and the information of the users' IDs are stored and maintained at its original source. For example, with the *Radar View*, each user can assign or modify his/her user name and the color of corresponding telepointer from his/her local device and the id information will be updated through the awareness mechanism. Although the group awareness server within a dynamic workspace might be centralized, there is no unified restriction of user ID setting.

Centralized user ID is widely applied in instant messaging based techniques, in contrast with *distributed user ID* , each user is required to register and logon on a unified server in order to apply the feature of the interaction technique. Examples include *Dynamo* [Izadi et al., 2003], where each log-in user has a unique color labeled on the control bar, and each interaction point is labeled with the same color according to the user's account.

Facial recognition supported by *vision-based* techniques provides a direct and perceptual method of user identification. The system can recognize a user via 2D image analysis, i.e. to compare characteristics of the facial features with the information stored in pre-defined database. In *tracking system* or *motion sensing* environment, the sensors can capture 3d face image, thus improve the accuracy.

For some interaction techniques *no user ID* is required. The system can only recognize the *device id*. For example, with *Pick-and-Drop*, each interactive pen has a pen-id and can be uniquely identified by the system. Besides, the identification method such as scanning 2D barcode can also be classified into this sub-dimension.

Visual labeled

System identified

The second group of sub-dimensions indicates whether the users can distinguish their own input signals from others with eyes.*Visual labeled* means users can distinguish his/her input from others by different appearance, such as colored labels, different shapes. E.g. by using *Dynamo*, each interaction point is represented by a color-coded telepointer. For some interaction techniques no *visual feedback* of the input ownership are provided, although the system can distinguish the ownership of input signals. *System identified* techniques includes *TractorBeam*, *SyncTab*,etc.

It's to be noted that this sub-dimension is used to identify users or devices when the referential environment requires a digital representation, such as non-spatial or *coupled virtual and physical space* referential. When the user directly utilizes the physical environment, i.e. the environment is referential as *physical space*, the identification method is considered to be *visual labeled* as default, since the user directly manipulates the input or medium devices during the task, and it's easy and clear to distinguish the ownership of devices among multiple different users.

3.2 Summary of interaction techniques

According to the 10 dimensions we have, properties of the selected 13 major interaction techniques will be summarized and listed in 3 tables. The dimensions within the same table describe one aspect of an interaction technique. For example, from the Figure 3.1 we can have an overview of the selected techniques from the aspect of initiation method. in Figure 3.2, the properties that are relevant to the referential environment and the control mechanism are summarized. Besides, the parallelism in multiuser or single user cases, and the corresponding identification methods, are depicted in Figure 3.3. The proposed design space is based on these summaries.

Our approach of classification

The term 'interaction style' refers to the ways users interact with the computer system. As we narrow it down to our particular topic, we intend to describe how the users are enabled to finish a common task with different interaction techniques. However, those interaction techniques are hard to be categorized into over-simplified groups, since each technique we choose has subtle distinctions comparing with the others. Therefore, the interaction style is implied under the comprehensive dimension analysis. In next Chapter, we will introduce those techniques in detail and put them into the proposed design space, followed by the design space analysis for each of them.

	Input Method					Positional Mapping			Replace-ability of input device				Power of Working Area		
	Mouse/ Pen/Finger	Token-based	Trackings./Motion sensing/ Sensor-based	Button-based	Vision - based	Relative	Absolute	Rate-based	Dedicated	Alternative	Compatible	Use anything as Input	Within Hand's Reach	Within Arm's Reach	Beyond Arm's reach
Radar View	✓						✓				✓			✓	
Hyperdragging	✓					✓					✓		✓		
Passage		✓										✓			✓
Perspective Cursor	✓					✓					✓		✓		
Drag-and-Pop/Pick	✓					✓					✓		✓		
Superflick	✓							✓			✓		✓		
Pantograph and Slingshot	✓						✓				✓		✓		
Deepshot					✓						✓		✓		
IM-based Techniques	✓					✓					✓		✓		
TractorBeam	✓		✓	✓			✓		✓					✓	
Pick-and-Drop	✓	✓					✓			✓					✓
SyncTab				✓						✓			✓		
Synchronous Gesture			✓				✓		✓				✓		

Figure 3.1: Summary of interaction techniques: Initiation method

3.3 Design rational

With the dimensions presented in the previous section, the design space of interaction techniques is illustrated. We use three tables to represent the corresponding aspects of the properties: the initiation method, the referential environment and control mechanism, and the parallelism. Each table together with the dimensions inside forms a 'design subspace'(or subspace). Although every dimension in the whole design space may have potential impact on the others, the dimensions within the same subspace are strongly coupled. In this section we are going to discuss about the design rational for each dimension in the corresponding subspace, especially the interrelation between dimensions.

subspace

	Referential Environment				Input Model Types			Feed-forward		Feedback		
	Spatial		Non-spatial		Planar	Perspective	Literal	Open-loop	Closed-loop	Visual	Audio	Haptic
	Coupled virtual and physical space	Physical space	Discrete/ No topology	Virtual space								
Radar View	✓				✓				✓	✓		
Hyperdragging	✓				✓				✓	✓		
Passage		✓					✓		✓	✓		
Perspective Cursor	✓					✓			✓	✓		
Drag-and-Pop/Pick	✓				✓				✓	✓		
Superflick	✓				✓				✓	✓		
Pantograph and Slingshot	✓				✓				✓	✓		
Deepshot			✓						✓	✓		
IM-based Techniques			✓						✓	✓		
TractorBeam	✓					✓			✓	✓		
Pick-and-Drop		✓					✓		✓	✓		
SyncTab		✓					✓		✓	✓		
Synchronous Gesture		✓					✓		✓	✓		

Figure 3.2: Summary of interaction techniques: Referential environment and control mechanism

3.3.1 Initiation method

The division of subspace

As shown in Figure 3.4, in the initiation method subspace, the input method indicates how the input data can be sent from the user to the system, which is the choice needs to be made at the first place. If the input method involves pointing, the dimension *positional mapping* describes the physical properties of the corresponding input controls. Thus for the *button-based* and *vision-based* input methods, there is no alternative attribute of *positional mapping*, only the *power of working area* needs to be subdivided since the mapping method directly determines the space required by the task. For example, one of the design purposes of relative input device is to realize remote control via the manipulation of within hand's device. Thus the *power of working area* of a relative input device is *within hand's reach*. From this aspect the rate-based control is similar. On the other hand, for the

	Multiple Users		Single User		Identification				
	Parallel Operation	Serial Operation	Parallel Operation	Serial Operation	Distributed User ID	Centralized User ID	No User ID/ Device ID	Visual labeled	System identified
Radar View	✓			✓	✓			✓	
Hyperdragging	✓			✓			✓	✓	
Passage	✓		✓				✓	✓	
Perspective Cursor	✓			✓			✓	✓	
Drag-and-Pop/Pick				✓					
Superflick	✓			✓	✓			✓	
Pantograph and Slingshot	✓				✓			✓	
Deepshot				✓					
IM-based Techniques	✓		✓			✓		✓	
TractorBeam	✓			✓	✓			✓	
Pick-and-Drop	✓			✓			✓	✓	
SyncTab	✓			✓	✓				✓
Synchronous Gesture		✓		✓					

Figure 3.3: Summary of interaction techniques: Parallelism

direct input device, the prerequisite of the control is to contact the physical surface of the computing device directly. Thus the user may vary his/her position to approach destinations in various distances. Another physical property of input devices is the replace-ability, which indicates the potential choice among similar or simulative devices to all input methods.

3.3.2 Referential method and control mechanism

In this subspace (Figure3.5), the *referential environment* and *input model type* are presented in the columns. It should be noticed here this subspace describes two aspects of a technique. As discussed before, the *input model type* describes the mapping between the physical arrangement and the virtual space, when the environment is spatially referential. Thus only the columns *coupled virtual and physical space*s and

Positional mapping \ Power of working area \ Input methods		Mouse-Cursor/Pen-based/Finger and touch screen				Token-based				Tracking system/ Motion sensing/ Sensor-based techniques			
Relative	Within hand's reach												
Absolute	Within hand's reach												
	Within arm's reach												
	Beyond arm's reach												
Rate-based	Within hand's reach												
P. m.	Power. / Replace-ability	De.	Alt.	Co.	Any.	De.	Alt.	Co.	Any.	De.	Alt.	Co.	Any.

Button-based				Vision-based				I. m. / Power.
								W. h.
								W. a.
								B. a.
De.	Alt.	Co.	Any.	De.	Alt.	Co.	Any.	Power. / Rep.

Abbreviations:

I.m. : Input methods
Rep. or Replace-ability : Replace-ability of input device
P. m. : Positional mapping
Power. : Power of working area

De. : Dedicated
Alt. : Alternative
Co. : compatible
Any. : Use anything as input

W.h. : Within hand's reach
W.a. : Within arm's reach
B.a. : Beyond arm's reach

Figure 3.4: Design subspace: Initiation Method

physical space are subdivided into three parts, which represent the three kinds of input model types.

The headers of the rows present the control mechanism of an interaction technique. The feed-forward indicates the possibilities of control the moving object while executing an interaction. And each kind of feed-forward is divided into three sub-rows according to different modalities of feedback. The dimension feed-forward is strongly related to the feedback. Whether the feedback can indicate the final position of the moving object determines the type of the feed-forward.

At the beginning, when we started to choose dimensions, the continuous/discrete was also considered to be a sub-dimension. However, from the literature that we surveyed, we came to the conclusion that the feedback of interaction techniques with closed-loop control was always continuous, i.e. the possibility of adjusting the moving object was realized only when the system can provide user the immediate perceptible response in real-time. The discrete feedback means that user can get system response after his/her action is done. The typical example is the *button-based* input method. The continuity of feedback pervades many other dimensions, thus we decide not to list it independently to avoid redundancy.

The continuity of feedback pervades many other dimensions.

3.3.3 Parallelism

The last subspace describes the parallelism of the technique and the three kinds of solutions of identification problem during the multi-user parallel operation. From the Figure 3.6 we can see that the sub-dimensions in upper column header and the left row header indicate the use cases of single user or multi-user. The intersections of those rows and columns make four combination types: *Multi-user parallel operation, multi-user serial operation, single user parallel operation, and single user serial operation*. Each technique that supports multiple users' parallel operation should have two attributes among the four.

The parallelism plays an important role in the collaborative

Feed-forward / Feedback \ Referential environment		Spatial						Non-spatial	
		Coupled virtual and physical space			Physical space			Discrete/No topology	Virtual space
		Planar	Perspective	Literal	Planar	Perspective	Literal		
Open-loop	Visual feedback								
	Audio feedback								
	Haptic feedback								
Closed-loop	Visual feedback								
	Audio feedback								
	Haptic feedback								

Figure 3.5: Design subspace: Referential Environment and Control Mechanism

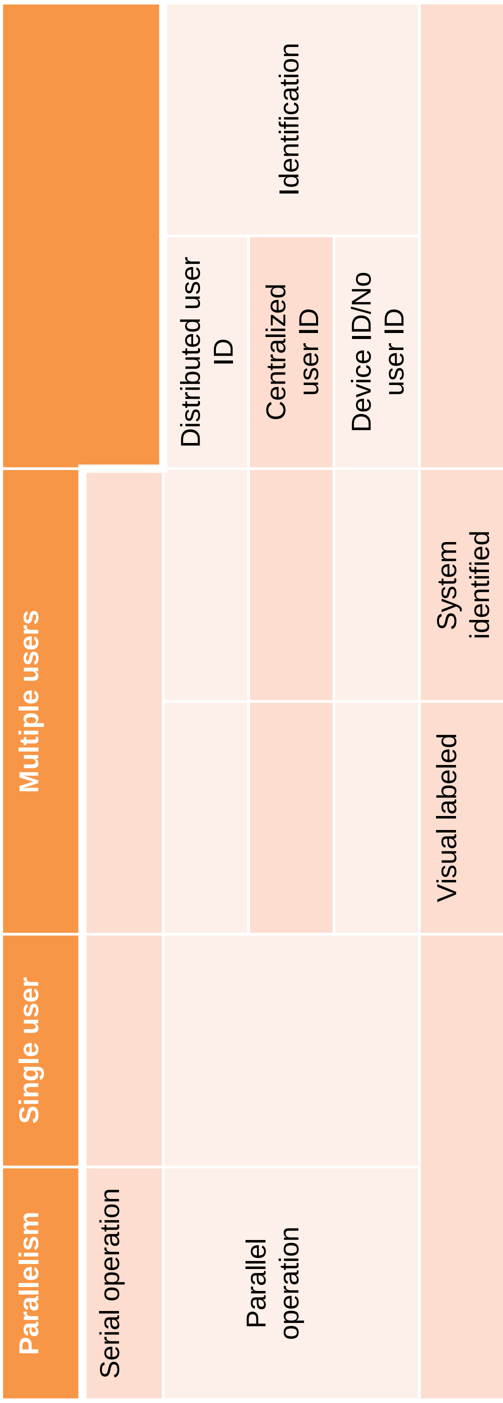


Figure 3.6: Design subspace: Parallelism

work space. For example, when using IM-based technique as file transfer tools, one user can send many files to different users at the same time. In contrast, when using the technique *Pick-and-Drop* or *token-based* techniques such as passage, the transmission can be initiated once a time. The first method may be effective in transferring several files, and protect the privacy of each user, but may cause the problem of losing focus of the transferred files. The second method will cost more energy and time when multiple objects need to be transfer one by one. However, group awareness is emphasized during the transmission. Thus the choice of the technique is depended on the scale and number of the objects to be transferred, and the priorities of group needs in a collaborative situation.

The type of identification can be categorized from two aspects. The sub-dimensions in right rows header indicate the ways of recognizing the identity of different users. The visualization types of different identification method are presented in the bottom column header.

3.4 Notations and representations

3.4.1 Notations

Merge composition

Dimensions are co-related with each other. We use normal/dashed lines to represent strong/weak merge relation.

In this section we will introduce how the properties of an interaction technique are presented with the design space. In our design space, the properties of the technique to be described are represented by nodes. The design space is comprised of three subspaces, within which at least one node should appear. And each node represents a synthesized attribute of the technique, i.e. it indicates the intersection of properties (sub-dimensions) from at least two dimensions. When two or more nodes appear in one subspace, they will be connected by one or more black lines. Similar to Card's design space, the black line stands for the merge composition. However, we specify the different kinds of merge composition. The nodes that appear within one subspace are strongly correlated, and they are describing the same aspect of the interaction technique. Thus we call this kind of connection *strong merge composition*. On the other hand, the nodes in different subspaces are connected by dashed lines, which stand for the *weak merge composition*. There is no sequential restriction among nodes, thus if there are more than one node in one subspace, the dashed line can connect any one of them to the node in other subspace.

3.4.2 Example of design space

Design space analysis of *Pick-and-Drop*

Taking the representation of *Pick-and-Drop* as an example, see Figure3.7, the node in the initiation method space stands for the physical properties of the input device and working area, i.e. the input device of *Pick-and-Drop* is a special digital pen, which is also an token [Holmquist et al., 1999]. This pen is an *absolute input* device and can be replaced by others under some special condition. When the user manipulates the involved devices and digital surfaces, the working area may extend to the position that is far away from where he/she is original located.

In the referential method space the node expresses the following properties: the involved devices and the working environment are spatially referential. The position information of the destination device is derived from the physical space directly. And no virtual representation of the work space is required.

The nodes and black line in the parallelism space mean that when a single user uses this technique, he/she can move an object once a time, but multiple users can initiate operation in parallel. With different device ids, the system recognizes which file is 'taken' by which pen. And the ownership of different input or mediate devices can be distinguished by the user's eyes.

The dashed lines connect the composite attributes from each subspace. The entire path together with all the nodes within it forms the comprehensive view of the described technique .

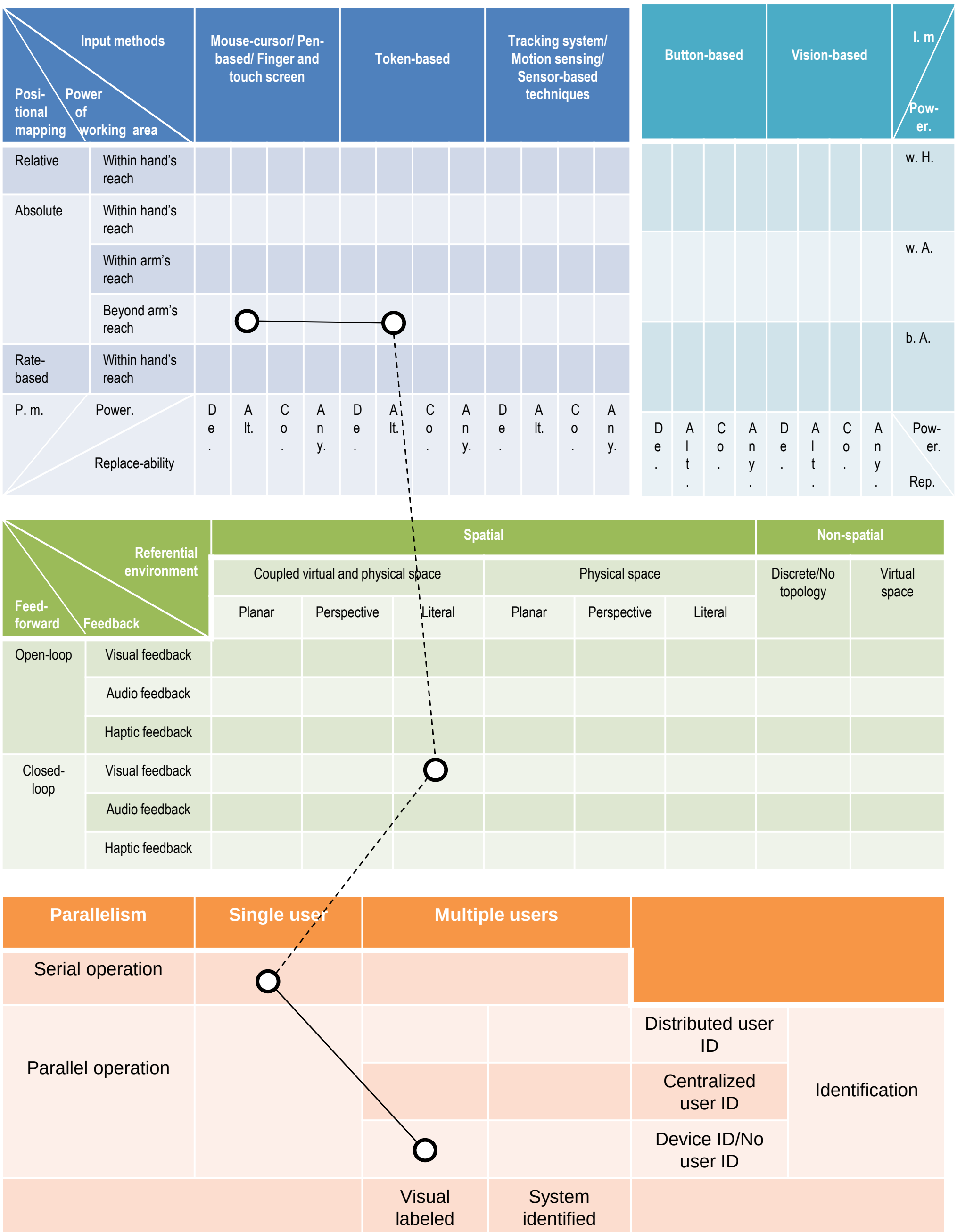


Figure 3.7: The Design Space of *Pick-and-Drop*

Chapter 4

Techniques supporting cross-device object movements

In this chapter, the selected 13 interaction techniques will be introduced from the aspects of design concept and technique capability. For each technique, first we describe how it enables users to perform the object relocation task, then a figure illustrating its synthesized properties via our design space representations will be presented, followed by the analysis according to the dimensions we defined in Chapter 3.

4.1 Radar View

The workspace is represented via a map

By duplicating the remote displays and keep their arrangement on the local screen, the technique *Radar View* [Gutwin et al., 1996] applies a miniature map to represent the entire workspace. Users are able to drag digital object from the thumbnail (or icon) of one device onto the other on the map. To trigger the appearance of the map, the user uses the pen to touch the object. As the map shows up, the pen should not be lifted until it drags the object to the target. The telepointer on the remote display will move and perform the same action according to the control on local screen. Devices could be added and removed according to actual needs of specific tasks. To keep the context information up-to-date, the miniature map changes dynamically according to the alteration happened in the real world.

The early version of *Radar View* is a groupware device of SharedARK system [Smith, 1992]. This awareness-enhanced miniature is initially applied to represent other participants' activities and the object motions within a workspace in real-time. In this thesis we discuss the main design concept of *Radar View*, rather than describing details of a specific version of *Radar View* widget.

Design space analysis

The design space of *Radar View* is shown in Figure 4.1. As a *pen-based* technique, *Radar View* enables object repositioning among remote displays by having direct contact with the local display only. To initiate this technique, any direct pointing device can become the input tool, such as *stylus*, *Finger and touch screen*. Therefore, the working area is small and under the control of the user's hand. However, if the map is displayed on a very large wall or big interactive surface, the user needs to manipulate with bigger movement range, thus the power of working area may be *within arm's reach*.

Input method of *Radar View*.

The dynamic context information of the entire workspace is gathered and represented in the form of a digital planar map, which is realized via replicated architecture. It's a typical way that makes use of spatial information to present the work context. Moreover, the arrangement of the involved devices on the map is consistent with their layout in the real world. On the map the user performs *Drag-and-Drop* to relocate the object. With continuous feedback of the cursor, the final position of the object is determined directly by the user's motion.

The control mechanism and the use of spatial information

Radar View supports multi-user parallel operation with the system. However, if a user wants to move several objects within a map, he/she has to do it one by one. Since dragging is required by each operation and no more operation is possible until the previous one is finished, to a single user the operation is serial.

multi-user parallel operation; single user serial operation

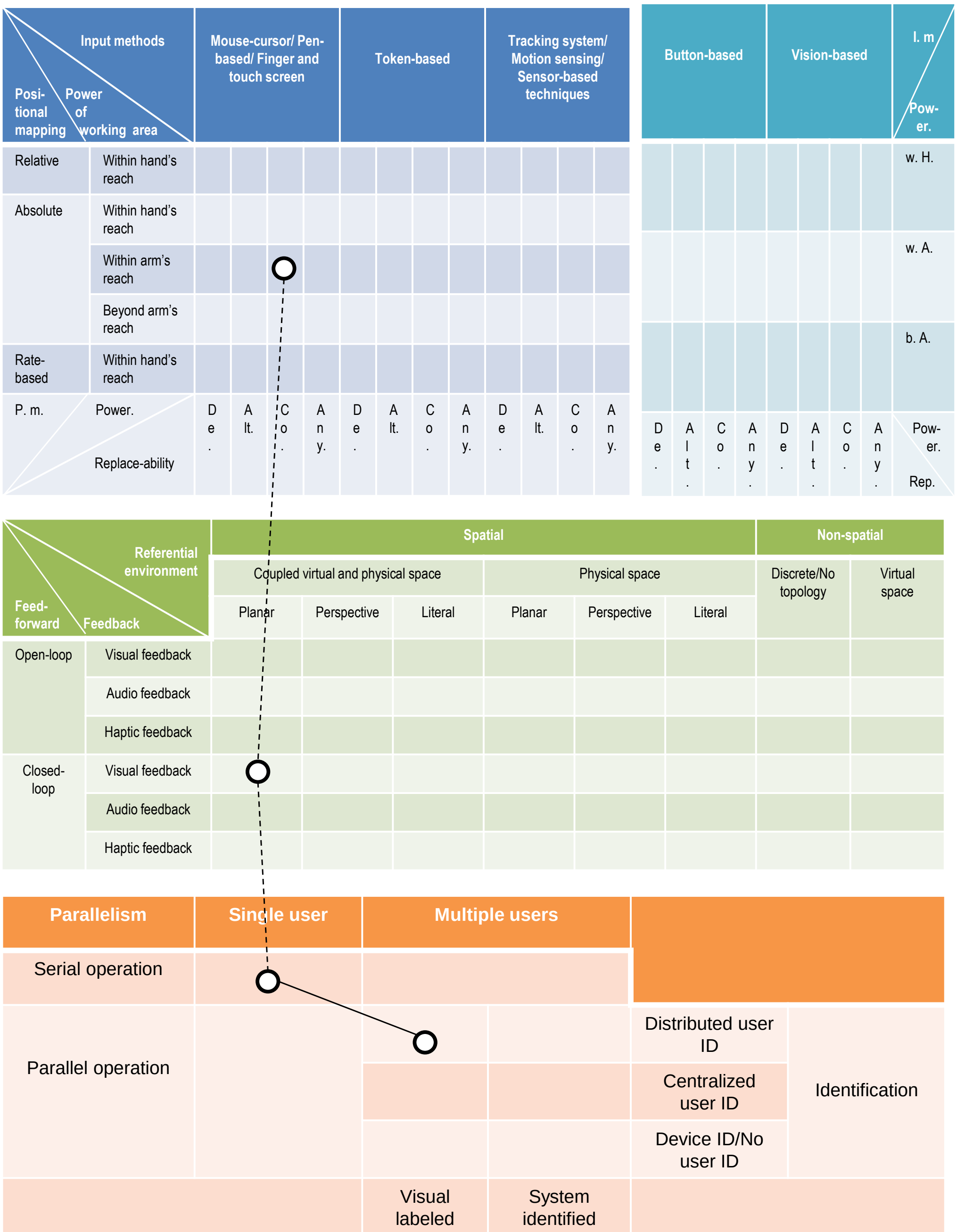


Figure 4.1: The Design Space of *Radar View*

Identification method

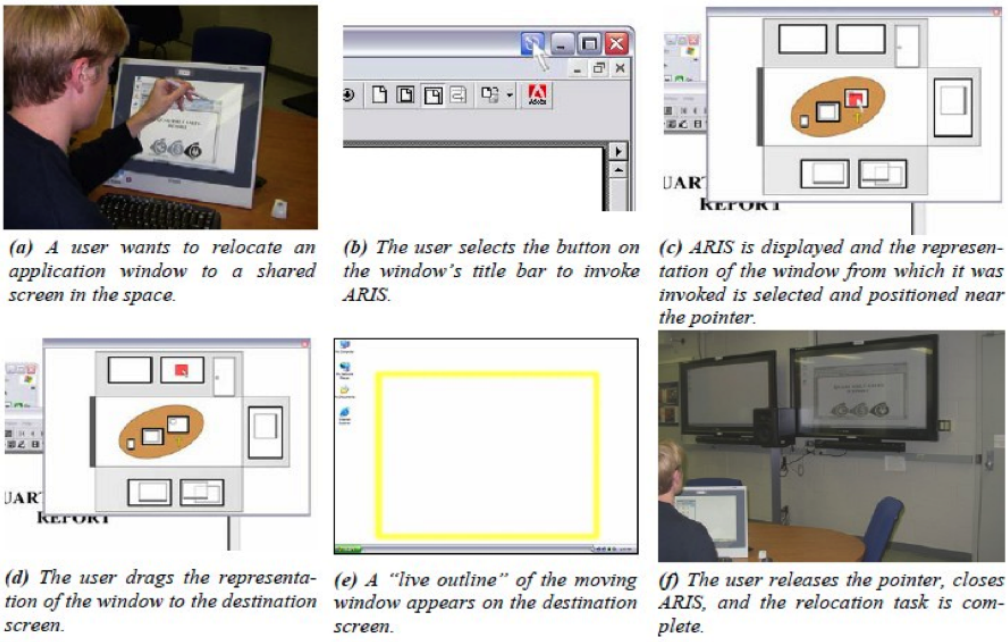
The view rectangles of participating users and the corresponding telepointers indicate the operating area of different users and their current actions [Gutwin, 1998]. Early versions of *Radar View* have low fidelity, and the representations of different displays and the view rectangles from different users are hard to identify. With developments and improvements, colors, names, pictures of users, icons, and telepointers are applied for identification. With *Portrait Radar* [Gutwin et al., 1996], user name or portrait attached to a view rectangle can serve as the distinguishing mark.

ARIS utilizes similar design concept of *Radar View*, which is also map-based.

Many techniques utilize similar design concept as *Radar View*. For example, *ARIS* [Biehl and Bailey, 2004] is an ego-centric and iconic interface which uses the same world-in-miniature concept. The involved devices and room components (e.g. wall, doors and tables, etc.) are represented by icons, each of which has the similar outline to the described entity. The map is composed by those icons and is arranged in a flattened form, with spatial relation corresponding to the layout in the real world. Figure 4.2 shows the interaction sequence of *ARIS* when a user performs an application window relocation task. We can conclude that it is of the same interaction style of *Radar View*.

with *Sketch Radar*, the physical correspondence can alter.

In the technical report of *Sketch Radar* [Aliakseyeu and Martens, 2006], the miniature map concept is used as well. However, rather than strictly keeping the layout consistency between digital representation and physical location, users are allowed to rearrange the map when the environment becomes familiar to them. As *Radar View*, the map is initiated with correspondence to the physical world. The difference is that the referential method of *Sketch Radar* is not always spatial. When a new device is added to the workspace, it can be recognized and connected by scanning bar code, which is discrete and non-spatial. Moreover, the representations of the devices on the map can be re-adjusted (e.g. size, positions, etc.), which brings operation flexibility and offers the possibility of application personalization.



(a) A user wants to relocate an application window to a shared screen in the space.

(b) The user selects the button on the window's title bar to invoke ARIS.

(c) ARIS is displayed and the representation of the window from which it was invoked is selected and positioned near the pointer.

(d) The user drags the representation of the window to the destination screen.

(e) A "live outline" of the moving window appears on the destination screen.

(f) The user releases the pointer, closes ARIS, and the relocation task is complete.

Figure 4.2: A user relocates an application window from his tablet to a shared display using *ARIS*. [Biehl and Bailey, 2004]

4.2 Hyperdragging

Hyperdragging [Rekimoto and Saitoh, 1999] allows users to move a digital document across physically separated displays. Users can select an object from one display and drop it on another by using a standard mouse in the smart environments, where the infoTable and infoWall (display digital data through LCD projectors) are applied as the extended desktop of the portable notebook. The use of dedicated applications brought constrains and complexity. The control of the cursor is from laptops and only specialized java applications can be run on the involved displays.

As a relative remote pointing technique, *Hyperdragging* extends the reaching range of traditional *Drag-and-Drop*. Interactions between devices are initiated by sliding the mouse along the local surface. With the commonly used mouse and dragging gesture, users can utilize the manipulation knowledge they already have.

Design space analysis

The referential method of environment depends on context

The input device of *Hyperdragging* is the standard mouse, with *relative* positional mapping and *within hand's reach* working area. Apart from the pre-established wired connections between displays, new devices can be added to the workspace in a *discrete* manner. For example, by reading the attached visual marker—printed 2d barcode, laptops and other physical objects such as tapes can be recognized when they are placed on the infoTable. After the recognition, an object aura around the physical object appears and represents its data space. Users can attach digital data to the object by dragging files from other screens into the object aura. We can conclude that after establishing connection, the involved devices in the workspace are referenced in a spatial manner. As input device, the mouse has virtual contact to other displays. Thus the movement of the cursor needs to be tracked by the system because of the relative mapping type. Thus the referential method of environment is *coupled virtual and physical space*.

Hyperdragging is a *closed-loop* control technique, because with continuous manipulation (dragging) users have full control of the object movement and its final position. As normal *Drag-and-Drop*, the moving object follows the trace of the cursor, which serves as the *visual feedback*.

Figure 4.3: When the user interacts with the extended surface, such as the table surface, the moving cursor and the corresponding notebook it belongs to are connected through a digital line, we call this visual label of identification *anchored cursor*. [Rekimoto and Saitoh, 1999]

Multiple users can perform *Hyperdragging* in parallel. Since the control can be from laptops only, identifying the ownership of users' cursors can be regarded as identifying the different laptops. A digital line is projected to connect the cursor and controlling computer. This line is called *anchored cursor* and becomes the visual label to identify device ownership. Besides, the identification can also be realized through *device id* because of the unique marker attached to each device, with which the device can be recognized by the system and added to the workspace.

Distinguishing different users can be considered as identifying different devices

PointRight also enables usess to control the pointer across displays among different devices. Comparing with the *Hyperdragging*, those screens driven by the same Operating System (OS) are physically aligned [Johanson et al., 2002]. It pairs the edges of adjacent displays and virtually connects them, so that the user can move a cursor across boundaries of the displays as if they are an extension of the local control surface. A map of the topology of the screens is maintained and updated dynamically by the system. Although *PointRight* can realize cross-displays pointing via mouse as well, it doesn't rely on the dedicated applications. However, when parallel operation is performed, no *visual feedback* of the cursor ownership identification is provided. The input streams from different users can be distinguished by a specific application, thus the identification method is *system identified* with *distributed user ID*. With Figure4.4 the design space of *Hyperdragging* is visualized.

Similar technique: *PointRight*

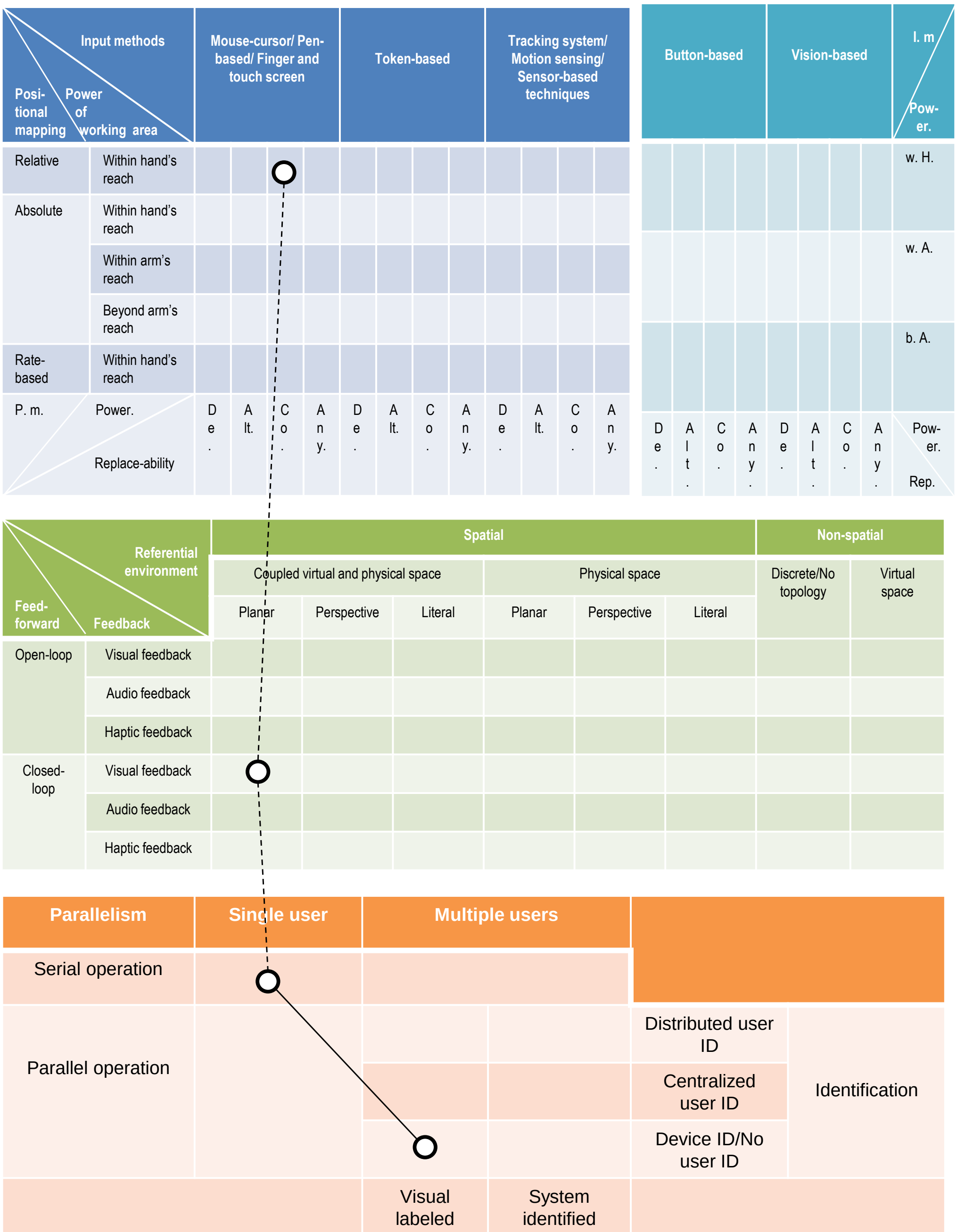


Figure 4.4: The Design Space of *Hyperdragging*

4.3 Passage

Passage [Konomi et al., 1999] is the technique applied in i-Land [Streitz et al., 1999], which enables user to use arbitrary physical objects as the medium device(the so-called *passenger*), as long as it has the certain weight. The design concept is giving digital information a physical form and providing the flexibility of choosing transmission medium.

Introducing the 'bridge' and 'passenger'

When a user wants to transfer files to another machine, he/she chooses a physical object, which can be anything he/she has at hand, such as a bunch of keys, a watch or a lighter. This physical object is put on the device *bridge*, whose physical part is like an electronic scale, and the digital part is the pop-up window on the screen when this medium device is recognized by the mechanism. The digital part of *bridge* indicates the data space of the corresponding *passenger*. To assign information to the medium, the user can simply drag-and-drop the file icon to the data space. After that the user can carry the information to be transferred like carrying a physical object to the destination device. To recognize the data assigned to the medium device, the user puts it on the *bridge* that connected to the destination computer, the information carried by the medium can be recognized by the weight of this *passenger*.

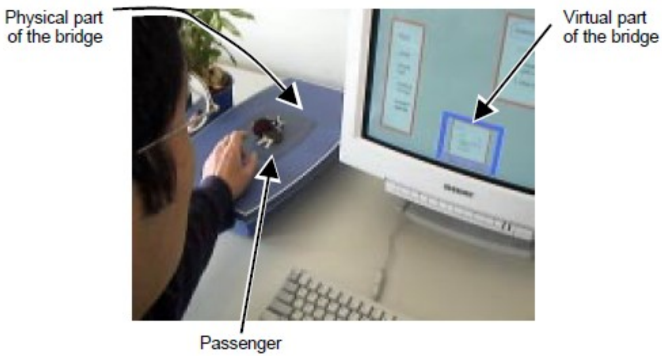


Figure 4.5: A user assigns digital information to the key chain using *bridge*. The physical part of the *bridge* is an electronic scale and the virtual part is a window, which shows the data space of the chosen physical object, the so-called *passenger*. [Konomi et al., 1999]

Design space analysis

Token-based technique

As shown in Figure 4.6, *Passage* technique utilizes the *token-based* input method and the chosen physical object(medium device) is not a storage. The digital object to be transferred is only assigned to this medium device, and other mechanisms are required to recognize this assignment. To be more precise, the *passenger* is a container, which means the physical properties of the medium device doesn't affect the performance at all. We consider this level of input device replace-ability as *use anything as input*, since the only identifier is the weight of the physical object. After assigning digital information to the *passenger*, the user needs to physically take it to the destination device and thus the working area may *beyond arm's reach*.

The *Passage* technique enables users to fully utilize the spatial information, so that they can manipulate the medium device in the *physical space* directly. The design concept is similar as *Pick-and-Drop* in the aspect of mimicking the relocation action of a physical object in the real world. Thus the input model type is literal, and feed-forward is *closed-loop*. In this design the virtual world and the digital world are connected using the device *bridge*, which is able to assign and read the data only when the object is put on it. After assigning information to the medium, during taking the medium device towards the destination, no feedback is provided to indicate whether this object still has the data. When the *passenger* arrives at the destination and is put on the Bridge accordingly, the virtual part of the Bridge becomes the *visual feedback* which indicates what information has been brought here.

Using weight as the identifier may cause recognition problems

Since the only restriction of the medium device is its weight, the problems come when more users use physical objects that have similar weight. Although they may have big difference in appearance, there will still be an identification problem to the system. As an alternative feature, the Bridge can identify electronic tags, thus provides a more reliable method for device recognition. The electronic tags can be served as device id, and the appearance of the chosen physical object serves as the visual label, which is obviously identifiable. Multiple users can take different *passengers* to different *bridges* without conflicts. To a single user, several medium devices can be taken with him or her simultaneously. Although during the recognition process, the *bridge* needs to identify each medium device one by one, the relocation task can be seen as parallel in general.

Passage is different from removable storage device in nature.

From the ways of system recognition and the characteristics of token [Holmquist et al., 1999], the *passenger* and *bridge* seem to be similar comparing with our commonly used USB drive and USB slot. However, despite the replaceability of input device, they have great difference. The essential distinction is that the digital object or information is stored on the USB drive.

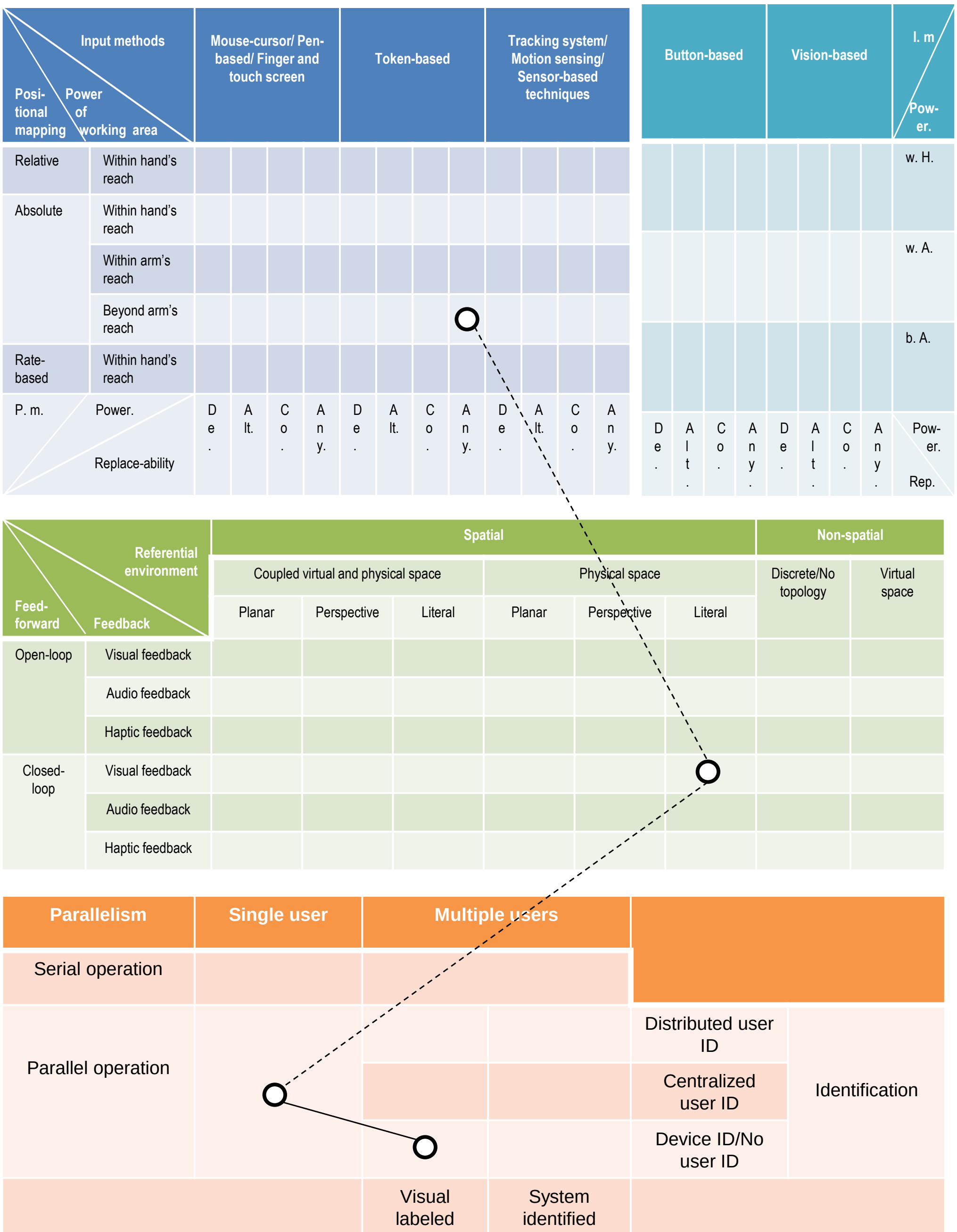


Figure 4.6: The Design Space of *Passage*

4.4 Perspective Cursor

The *Perspective Cursor* [Nacenta et al., 2006] is a perspective-based interaction technique that provides natural mapping between the pointer and display space. The design of perspective techniques aims at improving the interaction efficiency in MDEs by using the user's own point of view. To be more precise, the design concept is to improve the accuracy of direct remote pointing by improving visibility of control.

Perspective Cursor helps users get better views of the cursor, thus improve the control accuracy.

When applying the *Perspective Cursor*, the user can move around the cursor across displays, but the cursor appeared on each display seems to be contiguous and its size keeps unchanged to the user, i.e. the actual size of the cursor may vary, but from the user's location its size keeps constant. To enable this kind of intuitive pointing and determine which displays are adjacent in the user's point of view, a 3D model that represents the whole environment in real time together with a 3D position coordinated of the user's head is used [Nacenta et al., 2006]. Besides, *Perspective Cursor* provides control to the user only over the displays that are visible to him/her.

Design space analysis

The design space representation is depicted in Figure 4.7. As a mouse initiated technique, *relative* positional mapping device together with tracking sensors are used to determine the relative distance between the destination display and the user. The sensor that detects the position of user's head provides the coordinate of the user's point of view.

The use of spatial information

One highlight of this technique is to use the position knowledge of the user and the original spatial information of the whole workspace to provide intuitive layout [Nacenta et al., 2006] of control space for users. Comparing with the traditional stitching technique, multiple displays are virtually connected and represented in a single plane regardless of the actual situation of devices' alignment in the real world, the perspective technique enables users to directly use the real environment, which avoids the spatial inconsistency, and provides intuitive *visual feedback* of the mov-

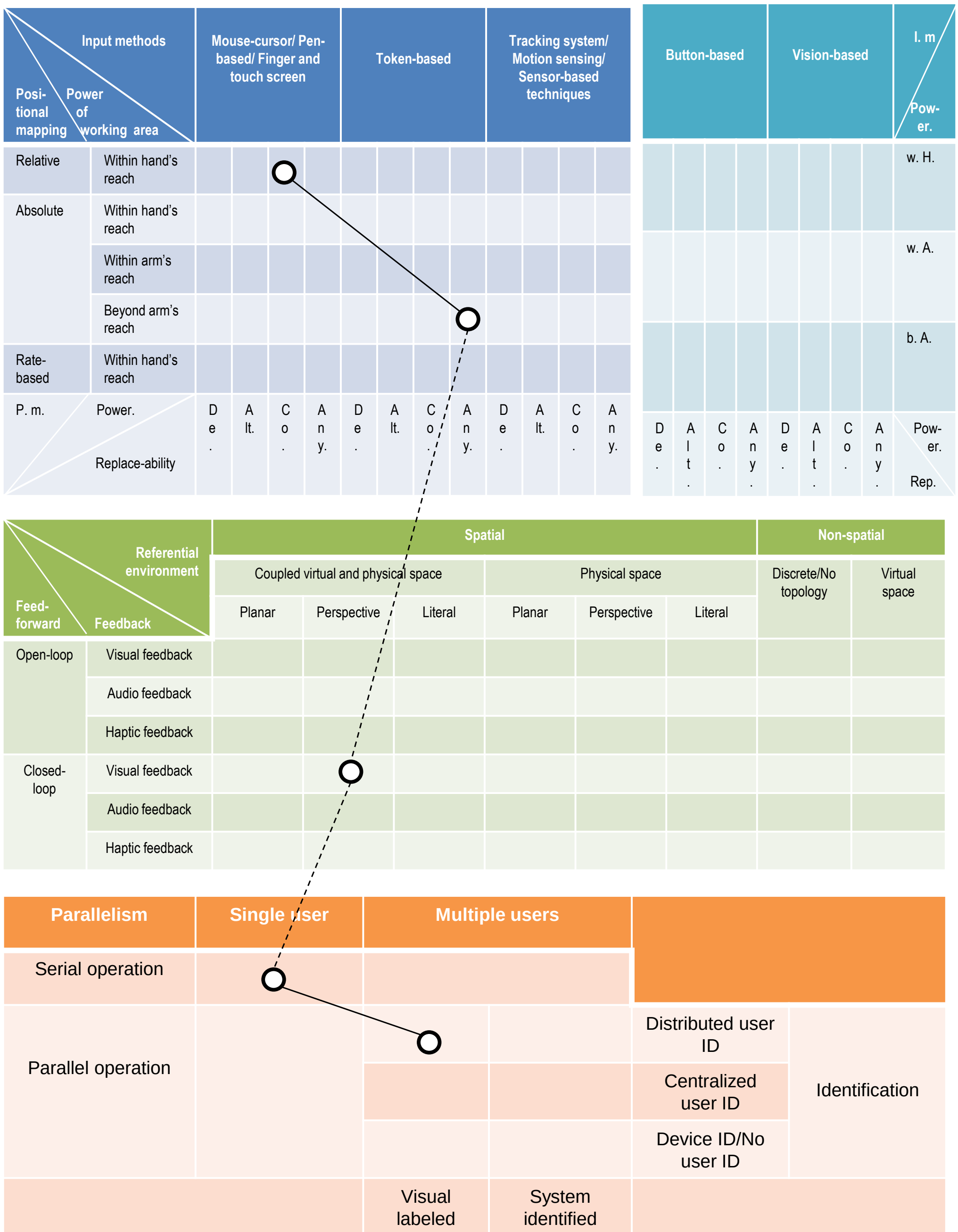


Figure 4.7: The Design Space of *Perspective Cursor*

ing cursor.

To be able to show the continuous cursor movement to the user, the virtual location of the cursor and its movement are calculated according to the user's location coordinate, thus the environment is referential as *coupled virtual and physical space*.

Although continuous feedback is provided when the cursor is moving within a specific screen, the feed-forward is intermittent for the existence of gaps between displays. When the cursor moves into non-display area, the widget *halo* provides the approaching awareness. *Halos* are parts or the entire circle which are centered at the cursor and appear at least in one of the screens. the closer the cursor is to one display, the smaller the bending radius of the halo is shown on this display. The example can be seen in Figure 4.8.

The widget *halo* solves the blind-zone problem caused by gap between displays.

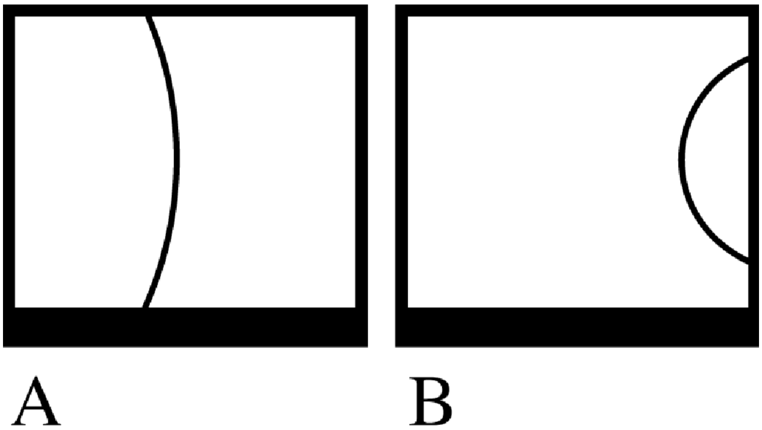


Figure 4.8: The halos served as the *visual feedback* when the *Perspective Cursor* is moving into no display area. Here are the examples of halos which indicate the distance of the nearby *Perspective Cursor* : A) the cursor is far from the left of the screen. B) the cursor is close to the right of the screen.[Nacenta et al., 2006]

The parallelism is not mentioned in the literature. The design concept of perspective cursor emphasizes the importance of accurate pointing and using user's point of view. When the object relocation task is performed, the user selects a target and performs traditional *Drag-and-*

Drop gesture, with the extended control range of the whole workspace. We can infer that to a single user, more operation will happen only in serial because the nature of dragging. And it is possible for multiple users to operate in parallel with the help of identifiable sensors on each user's head.

4.5 Drag-and-Pop and Drag-and-Pick

Drag-and-Pop [Baudisch et al., 2003] enables the user to relocate an object by moving its icon towards a target at first. As a result followed by this action, the proxies of the potential targets that are located in this direction are brought closed to the current position of the pointer. To relocate an object the user only needs to drag and drop it to the proxy of the target. After the release of the object onto the target proxy, all other proxy icons disappear.

Difference between *Drag-and-Pop* and *Drag-and-Pick*

Being different from *Drag-and-Pop*, the technique *Drag-and-Pick* is initiated on empty screen space. When the user performs a drag action on the screen, proxies of all the icons located in the direction of the drag motion pop up. The user drags one of those icons as selection, then only the proxies of the compatible icons left on the screen and become the potential targets. Afterwards it works like *Drag-and-Pop*.

Drag-and-Pop and *Drag-and-Pick* are proxy-based interaction techniques that bring remote target close to the user, thus only small drag motion is required from the user comparing with the traditional *Drag-and-Drop* [Baudisch et al., 2003].

Design space analysis

Both techniques work for all pointing devices, but are particularly useful for direct pointing devices such as *pen* or *finger and touch screen*. Thus it is flexible in choosing input devices. With these two techniques, the opposite approach comparing with the throwing techniques is used. The design concept of throwing techniques such as *Pantograph* or *Slingshot* is to magnify a user's motion. In contrast,

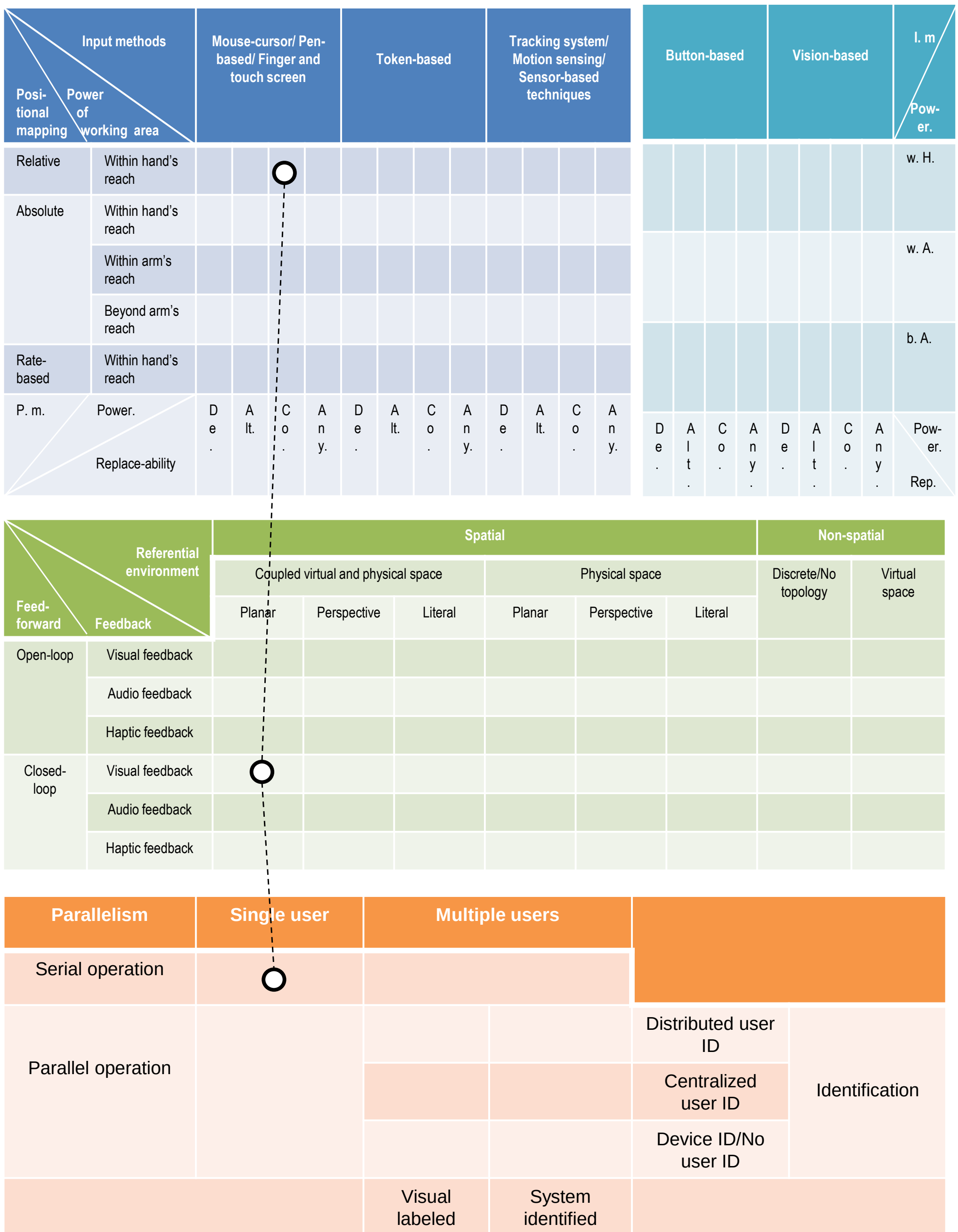


Figure 4.9: The Design Space of *Drag-and-Pop* and *Drag-and-Pick*

Drag-and-Pop and *Drag-and-Pick* bring a set of selection candidates close to the user, so that he/she can make use of the full resolution of hand. Therefore, no matter which mapping method of input device absolute or relative is, only small working area is required, i.e. *within hand's reach*.

The layout of the proxies keeps the spatial arrangement of their originals but with higher density. The referential method is *coupled virtual and physical space*, because the position information of the (virtual or physical) contacting point to the surface need to be tracked, which will affect how close the popped-up proxies of potential targets should be brought to the object. To filter the potential targets, only the proxies of compatible types and those located roughly in the direction of the user's dragging motion will pop up [Baudisch et al., 2003].Besides the continuous *visual feedback* as the traditional *Drag-and-Drop* gesture, the proxy of the target icon and its original are connected with a virtual rubber band.

According to the knowledge deriving from literature review, these two techniques are designed for single user use case. The design space is presented in Figure 4.9.

4.6 Superflick

Superflick [Reetz et al., 2006] is the improved version of the *Flick* technique. *Flick* is a kind of technique that uses a quick stroke gesture to make the object to be moved slide along the large interactive table for a relative longer distance. *Superflick* adds the optional adjustments functionality, which enables the user to correct the execution of his/her action. To move an object, the user applies *Flick* by quickly sliding the object in a certain direction for a short distance. The object is thrown out as soon as the user releases the pen. If the object is offset from the target and is still moving, the user can put the pen back to the table surface and initiate remote *Drag-and-Drop* to adjust the movement of the object.

The design purpose of flick and *Superflick* is to relocate object on a large digital table. By imitating the sliding action of passing a real object along the physical table, users can utilize their physical skill that they already have. After the flick or *Superflick* action is finished, the pointer remains close to the user, which prevents the problem of losing focus. It keeps the quickness and simplicity of flick technique, and adds the correction step to improve accuracy.

Design space analysis

Flick and *Superflick* are ate-based controlled techniques

To initiate the *Superflick*, the user can choose pen and touch screen, stylus, or any other direct pointing devices. It's to be noted here, the *Flick* and *Superflick* utilize the *rate-controlled* mapping. Although many so-called throwing-based techniques can amplify the user's motion, they don't actually utilize the velocity of user's movement. This important property is presented in Figure 4.10 and makes *Superflick* distinct from other techniques that use throwing concept. Theoretically, the indirect pointing device can also be used, but Reetz et al. [2006] focused on the direct input device because they intend to emphasize the importance of using physical knowledge and mimicking real world action. During the flick step, the motion of the object is determined by size, duration and direction of the flick gesture [Moyle and Cockburn, 2001].Since it is throwing-based, the short movement of input device can be mapped to a long distance of the object, the required manipulation space on the local screen is small.

The adjustment of the moving object is optional

The concept of taking advantage of user's physical experience makes the referential environment spatial. When the object is sliding along the digital surface, its position is tracked by the system. Thus it can be further classified as *coupled virtual and physical space* according to our definition. The final position of the object is displayed as soon as the flick gesture is complete, so that the user can decide whether an adjustment action is needed immediately. Continuous *visual feedback* that indicates the final position of the object is provided during the correction process. The control mechanism of *Superflick* is *closed-loop* due to the optional adjustment for the final position of the moving object.

Within a large interactive table, users can perform *Flick* or *Superflick* gesture in parallel. To a single user, more operation must be taken in serial because of the continuous

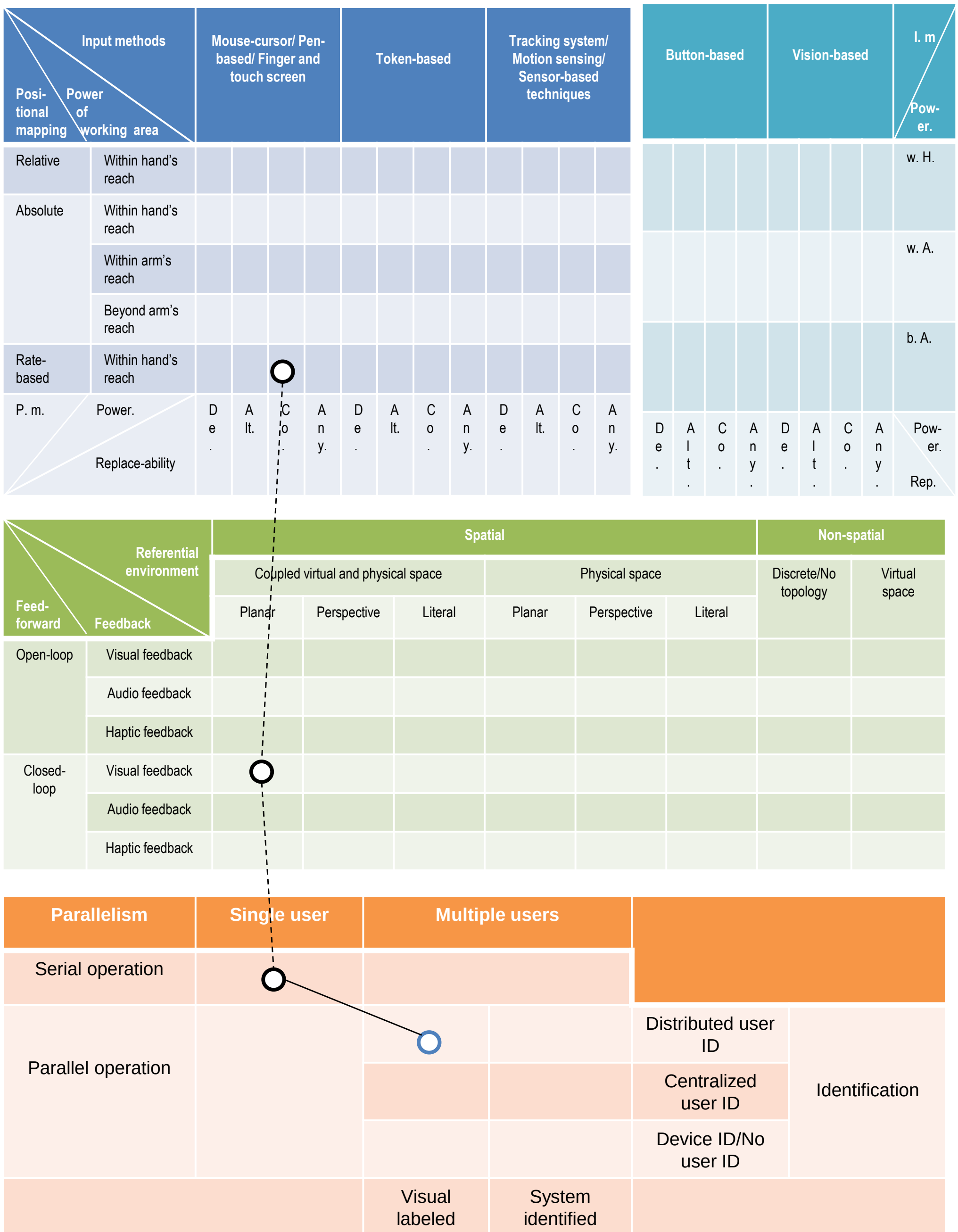


Figure 4.10: The Design Space of *Superflick*

contact between input devices and surface during the flick step. There is no further detail about how to differentiate users when they work in parallel. But we can infer from the design description that possible visually labeled mark can be served as identification method. For example, a colored digital line which connects the position where the user release the pen and the indicated final position of the moving object, can become the *visual feedback* and the color of telepointer can serve as the *visual labeled* identifier. In the design space, we use the blue node to represent that this choice of identification is from our assumption rather than directly derived from the literature.

multi-user parallel operation; single user serial operation

Other throwing-based techniques include *flicking and catching* developed by Aliakseyeu et al. [2008], which utilizes the naive flick concept to pass object to the user sitting around the table; the throwing-based techniques *Pantograph* and *Slingshot* are similar to *Superflick* except the way of positional mapping. They will be described in detail in the next section.

Similar techniques

4.7 Pantograph and Slingshot

Pantograph (*Push-and-Throw*) and *Slingshot* (*Drag-and-Throw*) are cursor extension techniques that allow relocating object from the local screen to other displays. The user touches the object with a pen and slides it forward or backward to define the movement distance (using *Pantograph*, the user slides to the same direction to the destination, in contrast, for *Slingshot* the pen is moved backward), moves left or right to adjust directions and the line becomes continuous feedback which indicates the reachable direction and range. Finally the user releases the pen to complete the task and the object is 'thrown' to the target.

Design space analysis

Pantograph and *Slingshot* are techniques which both use throwing models [Collomb and Hascoët, 2004]. This kind of technique is characterized by the mapping type, i.e. the short movement of the input device is mapped into the long

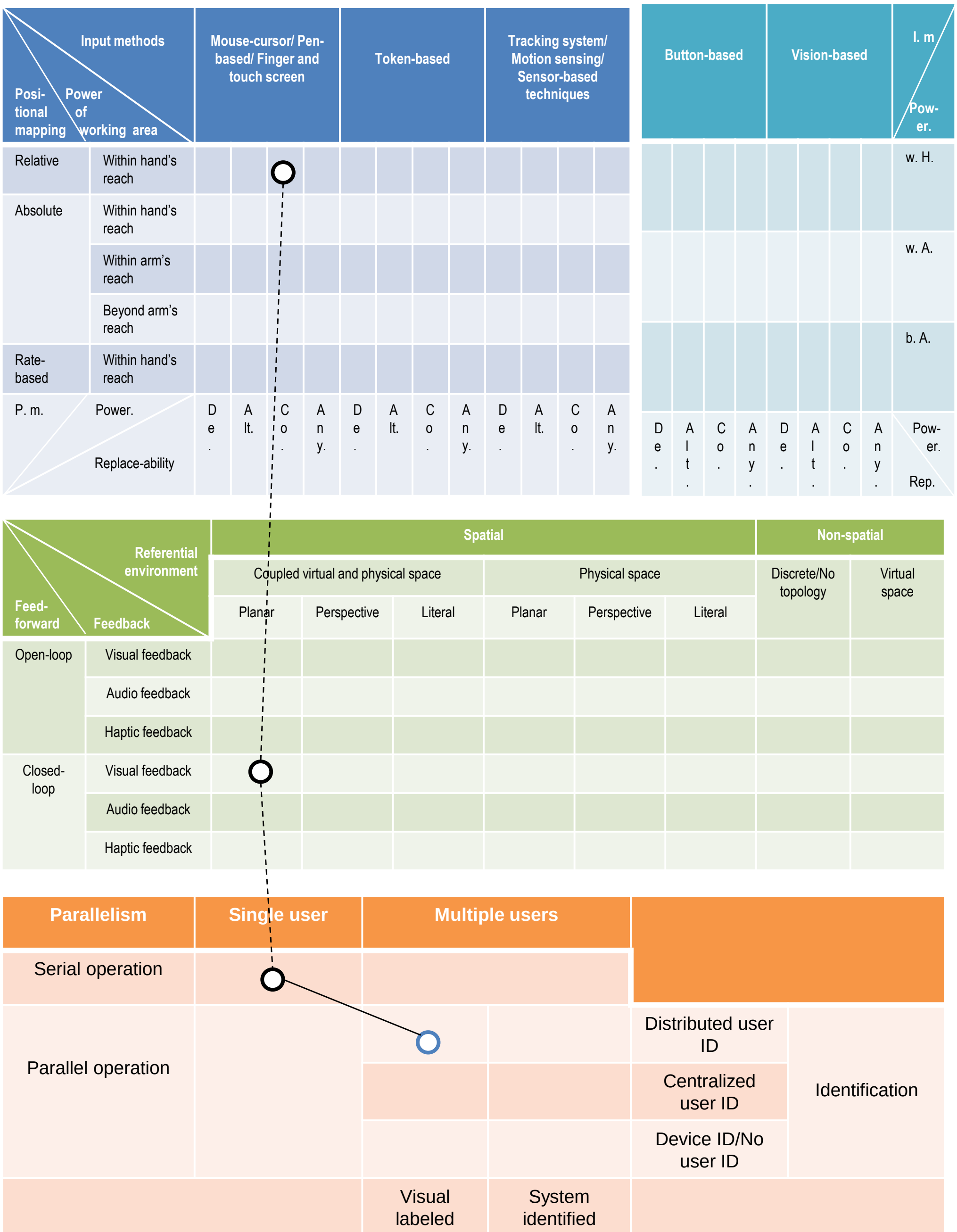


Figure 4.11: The Design Space of *Pantograph* and *Slingshot*

movement of the object. According to this characteristic, the positional mapping is relative when the 'throwing' is executed, even when the user is using a pen. To initiate these techniques the user can chose any pointing device, regardless of the mapping type [Hascoët, 2003].

Limitations of *Slingshot* caused by the take-off area

Since the difference between these two techniques lies in the slide directions, the requirement of working area range and the resulted performance differs as well. Although their working areas are both *within hand's reach*, for *Slingshot* the possible movement of the pointer is limited in the take-off area. The so called take-off area is in semi-circular shape and the center of the circle is the starting point of the drag action. The amplification of the pointer movement is not linear because the surface of the take-off area is not equally distributed [Collomb and Hascoët, 2004] to the surface of the target field. The lever effect [Hascoët, 2003] resulting from this design may cause low precision problem and difficulty in aiming at the target. For *Pantograph* there is no such problem because of the linear amplification of the pointer movement.

The spatial referential method used for both techniques is similar to *Superflick*, which is *coupled virtual and physical space*. To indicate the path of the object to be shown along the surface, digital lines serve as continuous *visual feedback*. The analysis of parallelism and identification is similar to what we described in the previous Section 4.6—"Superflick". The design space of *Pantograph* and *Slingshot* is presented in Figure 4.11.

4.8 Deepshot

Deepshot [Chang and Li, 2011] is a framework that enables users to migrate information, tasks or running applications between mobile phone and other devices. It supports two types of techniques, *deep shooting* and *deep posting*, to realize information pulling and pushing.

Deep shooting for information pulling

For *deep shooting* technique, information from computer can be migrated to the mobile phone by taking pictures of the computer screen using the camera of the phone. The region of interest is identified through the analysis of the captured photo automatically. Moreover, the information of the running application or tasks, including the displaying content as well as the application states, is extracted and encoded as a Uniform Resource Identifier (URI). This encoded information is sent to the mobile phone and can be recognized through the mechanism of *Deepshot*. In this way the suspended task can be resumed and the application can be launched on the operating system used for mobile phone.

Deep posting for information pushing

In contrast, *deep posting* is used for pushing information to other computing devices. When using an application on the mobile phone, the user can keep the current process on the screen of it, and make the mobile phone camera aim at the desired computer screen. The *deep posting* feature is triggered by pressing the hot-button on the mobile phone.Then the reviewed application on the phone will be shown on the computer screen with semi-transparent form. The application can be continued on the computer after the user's confirmation.

Design space analysis

Vision-based technique

As depicted in Figure 4.13, the *Deepshot* technique is *vision-based*, and utilizes the expressiveness of picture to extract deep information of the running task. With the help of hot-button the proper technique can be triggered without ambiguity. As the major input device of the *Deepshot*, the user can choose any ordinary camera equipped on the mobile phone. Therefor the input method is composite and the replace-ability of input device is *compatible*. Since all of the migration process is initiated by the mobile phone, the

Figure 4.12: Interaction sequence of the *deep shooting*:top-left, a user takes a picture of the screen shot using the mobile phone camera; top-right, Deepshot mechanism recognizes the running application in the captured picture; bottom, the suspended task can be resumed on the mobile phone. [Chang and Li, 2011]

user's hand has full control of the input device, the working area is small and within the user's hand's reach.

With this technique, the referential method of target device is non-spatial and *discrete*. Because shooting and accepting pictures of the screen-shot can be seen as producing and scanning a bar code, and the spatial information such as location of the target device or its relative distance to the source device is not important. The only constraint of the environment deployment is that the target device should be close enough to the camera, so that the picture taken by mobile phone could have adequate resolution in order to be identified.

The user has full control of the mobile phone and the camera. The mobile phone is the destination when using *deep*

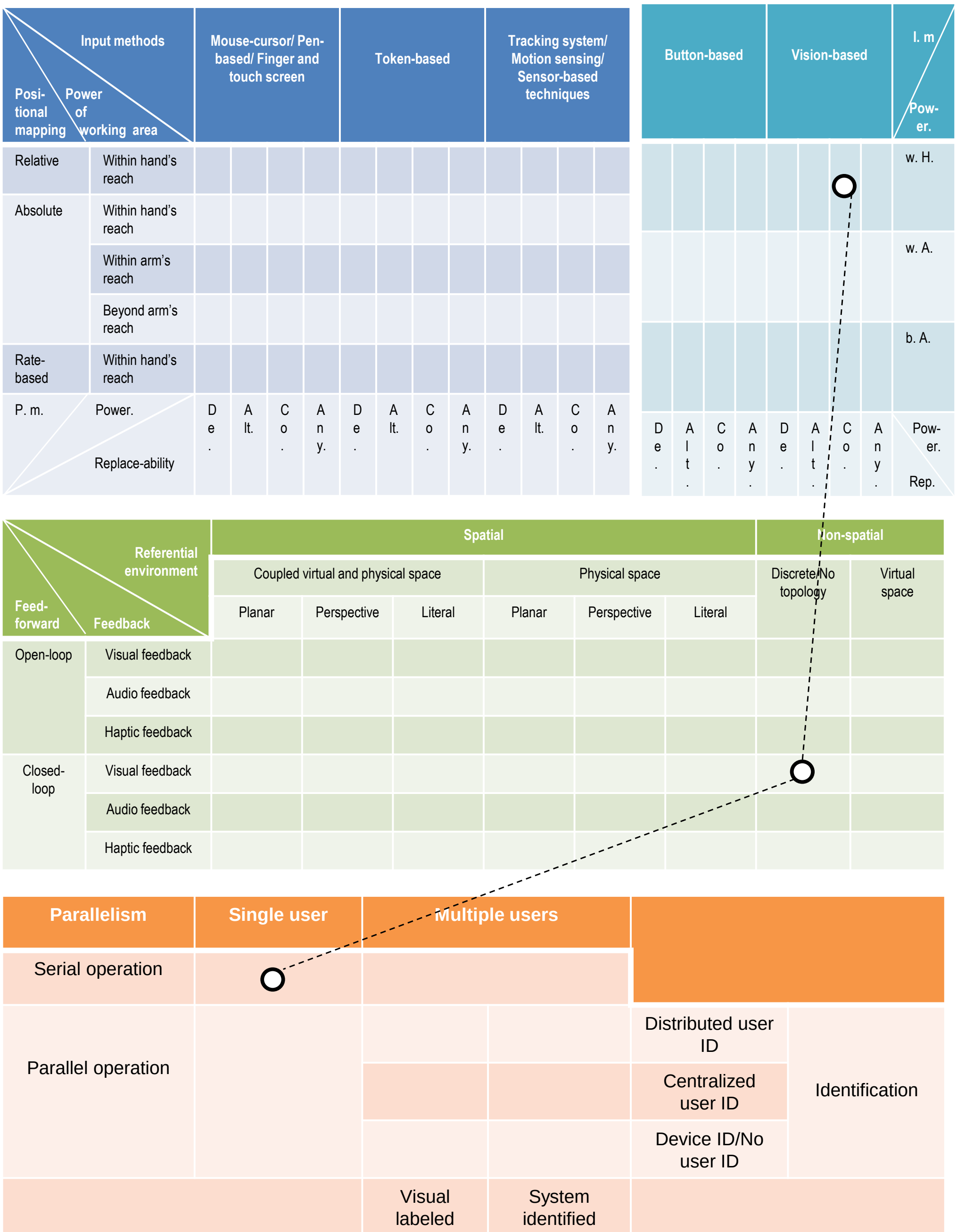


Figure 4.13: The Design Space of *Deepshot*

shooting. On the other hand, its camera is used for targeting the desired screen when using *deep posting*. Two kinds of *visual feedback* are provided to indicate the final position of the migrated task: before pressing the capture button, the previewed image displayed on the view finder is the continuous *visual feedback* and indicates the destination location. After triggering *deep posting*, the semi-transparent application window shown on the destination screen is another *visual feedback* which indicates the concrete parts of the screen where are occupied. Therefore, the feed-forward of both techniques are *closed-loop*.

Closed-loop control with *visual feedback*

Currently *Deepshot* is designed for single user to migrate tasks across personal devices. Authentication is required on each personal device. Multiple operation of a single user can be handled in serial.

4.9 Instant messaging as cross device interaction technique

Instant messaging (IM) is a text-based real-time interaction technique in widespread use. It enables the on-line communication between specified users. File transferring is one of the many features that IM offers, and mobile instant messaging allows IM services to be accessed from many kinds of portable devices. Interactions between devices are initiated by establishing a transmission channel, which is initiated by the sender via sending a transfer request and getting the permission of channel connection from the receiver.

Design space analysis

In this thesis we only focus on the file transmission task and the involved events, such as connecting channels or forming a network connections. Instant messaging is a service that requires internet connection and central user account management. The file sharing or transfer only can be processed between users who are on each other's 'buddy list'. Apart from that, there is not much constraint of the environment. The software can be installed on the computer or mobile devices from end users, and when considering file

Independency of user location and application transfer, only the pointing device is involved for selecting the sharing feature from the toolbar. Any kind of pointing device can be compatible with this technique.

Non-spatial referential

Taking the MSN Messenger as example, data exchange between end users is realized through MSN peer to peer protocol. There is no requirement of any spatial information of the environment. Thus the way of establishing transfer channel between end users is non-spatial and discrete.

The feed-forward is *closed-loop*, since once the connection with the desired user is established, no error of choosing the destination will happen. During the file transferring process, continuous *visual feedback* of the transmission state will be provided via a progress bar, which indicates the file size, transmission speed, and how much is still left.

Multi-user and single user parallel operation

To use the features of the IM, the user needs to log-in via user name and password. Then he/she can be authenticated by the central server of IM. Multiple transmissions for multiple users can be initiated simultaneously. A single user can send transfer request to several users and establish different channels in parallel. Because of the identification has been made during the course of logging onto the service, the multi-user interactions wouldn't cause conflicts. The design space is presented in Figure 4.14

This interaction style is characterized by the independency of devices and user location. Through a central user account management, multi-user' identification problem is solved. However, the concept of peer to peer data exchange can be adapted to other interaction techniques. It provides users an alternative of private interaction in the collaboration environments.

4.10 TractorBeam

TractorBeam [Parker et al., 2005] enables the user to access remote or close targets using a stylus on a large tabletop surface. The user can reach the remote target using the invisible beam casted from the stylus and select it by clicking

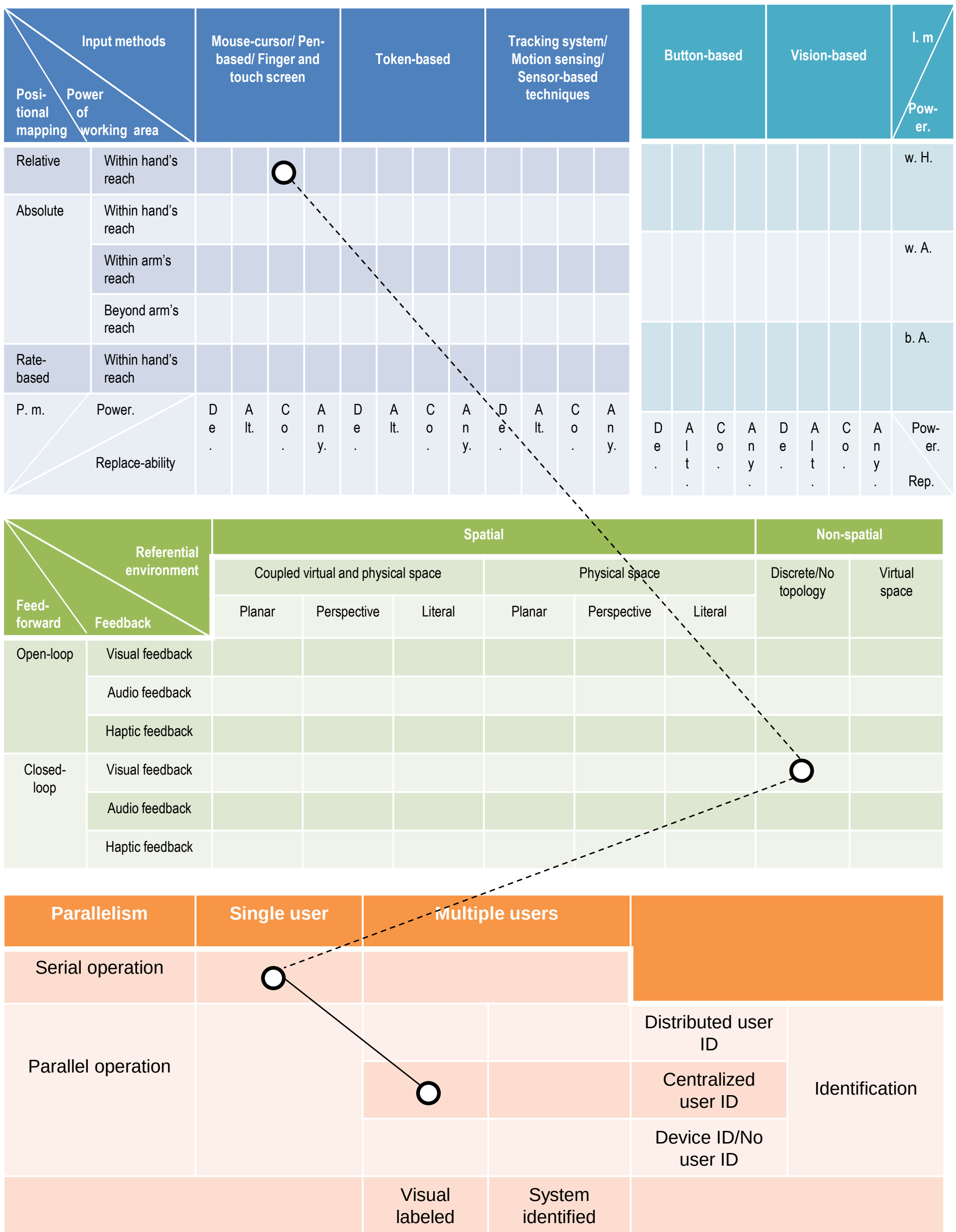


Figure 4.14: The Design Space of *Instant messages as cross device interaction technique*

a button on the stylus. Then the user can manipulate a remote dragging in order to bring distant object closer and do further interaction with it.

TractorBeam is a hybrid touch-point technique. The user can switch between a direct and indirect absolute device through lifting from or holding against the surface. And *TractorBeam* enables the user to access both close and remote targets without switching modes.

Design space analysis

Input method

The input device of *TractorBeam* is composite, which combines *pen*, *button-based* and *tracking system* input methods. The replace-ability of input device is *dedicated* because of the special sensors and hardware settings. When the user points at the table from above without physical contact with the surface, a cursor appears to indicate its trajectory [2005]. The task for selection can be performed by clicking the button. When the user manipulates the nearby object on the digital table, the stylus can be used as normal one and the relocation can be performed via *Drag-and-Drop*. For both operation the positional mapping is *absolute*.

Hybrid input model type

During the action of direct remote pointing, the spatial information of involved displays in the physical environment is fully utilized. The referential method is *physical space* since no digital representation of the environment is needed. Moreover, for absolute pointing, it's not necessary to store any position information of involved elements. The input model type is hybrid as well. It is *perspective* when remote pointing is performed for choosing *beyond arm's reach* targets, and *literal* when *Drag-and-Drop* is performed for choosing close targets.

Three selection aids for *TractorBeam*

TractorBeam is *closed-loop* technique, since when dragging the object continuous *visual feedback* is provided as the traditional *Drag-and-Drop*. To improve the accuracy of selection especially for the remote small targets, Parker [2005] developed and tested three selection aids for *TractorBeam*. By using those methods, if the cursor passes 90% of the distance between its starting point and the object location, there are three kinds of feedbacks that indicate whether the object is selected: expanding the cursor (use a 30mm halo under the

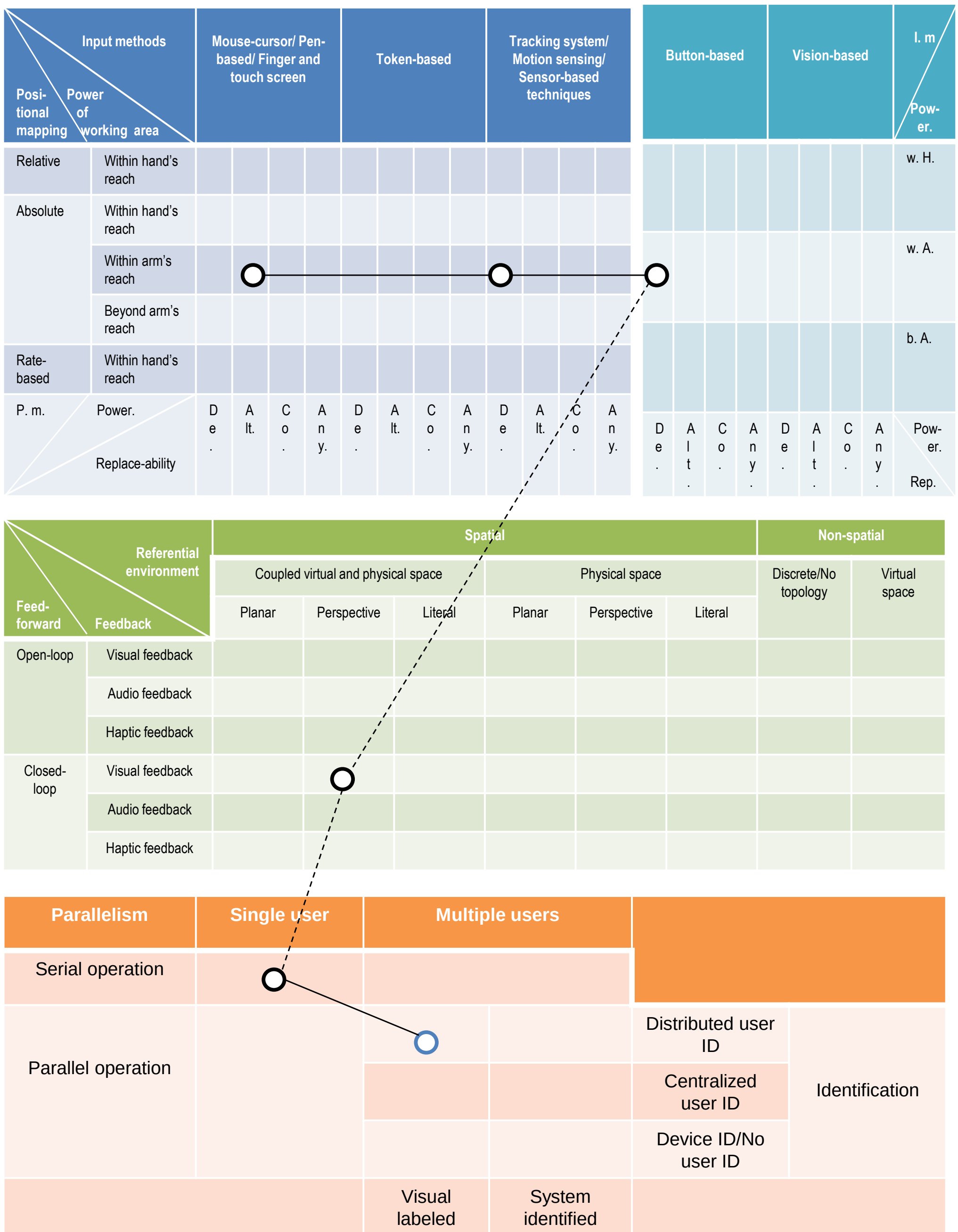


Figure 4.15: The Design Space of *TractorBeam*

cursor), expanding the target (the target expands to a predefined size), or snap selection (the cursor snap to the object center until it moves away outside the 90% range).

Parallelism and identification

Multiple interactions can be performed at the same time by using several *TractorBeam*s. There is no detailed description about the method of multi-user identification. We assume two possibilities. First possibility is that multiple cursors can be colored according to the different devices. Second one is using *distributed user ID*. Before operation each participant can choose a representative color, and when multiple operation happens in parallel, the system and other users will recognize which cursor belongs to whom. Thus it's possible to identify users by device id or *distributed user ID* in a *visual labeled* manner.

4.11 Pick-and-Drop

How does *Pick-and-Drop* work.

Pick-and-Drop, developed by Jun Rekimoto [1997], is a direct manipulation technique enabled by using a special pen. Users can pick up digital object from one display by tapping the file icon, virtually hold the information via the pen, and drop it on another display within the same network. Every pen is assigned with a unique ID, which can be recognized by the server on the network. The server is called pen manager. All computers or ported devices are connected to the network. When the user taps an object on the screen with the pen, the pen ID is bound with the object ID through the pen manager. Through this unique pen ID, the system is informed that the pen is virtually holding the object. Then, when this pen taps on the target display, data transfer from the first computer to the second one is executed under the control of the pen manager.

The use of pen manager.

As one of the earliest HCI (Human Computer Interaction) approach, the design purpose of *Pick-and-Drop* is to solve the inconvenience problem brought by the symbolic referential method of the work space. The design concept takes advantage of the tangible (physical) user interface, which is the better utilization of physical affordance [Ishii and Ullmer, 1997] than the traditional GUIs.

Design space analysis

The input device of *Pick-and-Drop* is special because of its functionality of the unique device id. The purpose of using such direct input device is to utilize the spatial information of physical locations. Although the digital object can be 'physically taken' by the pen, it is stored outside this input device. Thus the pen can be considered as a token, more precise, a container [Holmquist et al., 1999]. However, the power of working area may be large because of the need of direct contact with the destination surface.

As input device, the special pen can be replaced by other direct pointing device, to which a unique id can be assigned. In the original design of Jun Rekimoto's [1997], the input device is electromagnetic pen. In 2003, Henrik Gelius evaluated a prototype of *Pick-and-Drop* with some design modifications [Gelius, 2003]. The purpose of this study is to find out whether this technique can promote collaboration among children. In Henrik's prototype, the electromagnetic pen is replaced by pens assigned with RFID tags. Similar to the pen manager, a RFID reader can recognize the unique id information from each RFID tag.

The input device of *Pick-and-Drop* can be alternative.

For the design purpose we discussed earlier, *Pick-and-Drop* utilizes the physical space referential method and the input model type of it is literal. With these characteristics, the user only needs to tap the involved one or more surfaces twice, at the beginning and in the end. No continuous operation such as holding the left button on the mouse is required.

The *feed-forward* of *Pick-and-Drop* is *closed-loop* since the direct contact with the destination is required, so the user has full control of the final position of the object. Because no continuous manipulation is required, a special feedback is needed to indicate whether the pen is holding an object when it is getting close to the target surface. In the original design of Rekimoto's [1997], the object shadow is the *visual feedback*. It will first appear after the user tapping the file icon with the pen and lifting it as pick action. The shadow won't disappear until the pen has moved away and exceeds the predefined distance threshold. When the pen gets close enough to the destination surface the shadow ap-

Feed-forward and feedback of *Pick-and-Drop*.

object shadow

pears again. However, when the user picks the object and moves the pen away, the object shadow disappears. In such cases, although the shadow mimics the situation when people move a physical object in the real world, the feedback that can answer the following questions is lacking: whether an object is 'held' by the pen? Or what exact object it is when the user is on his/her way to the destination surface? To address this question, in Henrik's prototype [2003], the icon of the object to be moved is kept in the original place with faded colors.

Parallelism and identification method of *Pick-and-Drop*.

Parallel operation from different users is supported when using *Pick-and-Drop*. However, to each single user, the operation can only happen in serial because after picking an object, the pen can't pick another object until it drops the first one to the destination. To identify different users during the parallel operation, Jun uses identifier buttons on the pen while Henrik uses attached RFID tags. The identification aims at helping the system to differentiate users. There is no need to be visually labeled in the digital surface because of the literal input model type. The representation of design space is already presented in Figure 3.7.

4.12 SyncTab

Techniques for establishing connections between devices

SyncTab [Rekimoto et al., 2003] [Rekimoto, 2004]is a synchronized interaction technique for establishing wireless connection between devices. Interactions between devices are initiated by pairing physical buttons on two devices, i.e. the user presses and releases the connection buttons (*SyncTab* buttons) on both devices at the same time. To connect a remote device with a local one the user could make use of the *SyncTab* button on the controller.

Utilizing physical proximity

This interaction technique provides a solution to the problem of connecting heterogeneous nearby device. The need of this kind of connections is usually spontaneous and temporary. The main design concept of *SyncTab* is to utilize the physical proximity to specify target devices. Instead of choosing symbolic identifier from a list, with *SyncTab* the user can establish such connection in a more direct way.

Design space analysis

As depicted in Figure 4.17, the *SyncTab* is *button-based* technique, and users' hands have full control of the input devices. Thus the power of working area is *within hand's reach*. The *SyncTab* button could be specially installed, or could be any existing button on each device, i.e. the function of identifying and connecting other device could be added to other buttons, such as the escape button on the keyboard, etc. Because when *SyncTab* works, the synchronization of the press and release interval of both buttons needs to be checked. This special requirement of *Synchronous Gesture* avoids conflict between the additional connection function and its original ones. Therefore the input device replaceability of *SyncTab* is *alternative*.

button-based,within hand's reach,alternative input device

SyncTab is non-spatial referential technique. Since comparing with previously developed techniques, which promotes connection between nearby devices, it doesn't rely on any physical proximity and spatial information. Moreover, there is no need for virtual or symbolic representation of the devices to be connected. And the user has direct and full control of the devices. Thus the feed-forward is *closed-loop* since no possible mistaken destination will be chosen under our assumption. It relies on the physical environment because of the requirement of direct contact with the *SyncTab* button. It also holds for the *SyncTab* button on the remote controller. Thus the input model type is literal according to the definition. After the *SyncTab* buttons on both devices are pressed at the same time, the *visual feedback* will appear immediately to confirm the establishment of connection, which could be a message or blinking LEDs.

As a kind of *Synchronous Gesture*, the term 'at the same time' means 'within a certain time interval'. That is because of the consideration of the delay of network or human performance error. This reduces the user's manipulation difficulty, but brings some identification problems. The synchronization of *SyncTab* is checked by comparing the arriving time of the UDP packets from other device with the button release time of the local one. In addition, the time interval between the press and release time of the local button is also a parameter to identify the sender and receiver pairing. As shown in Figure 4.16, device A and B intend

How the synchronization of SyncTab is checked.

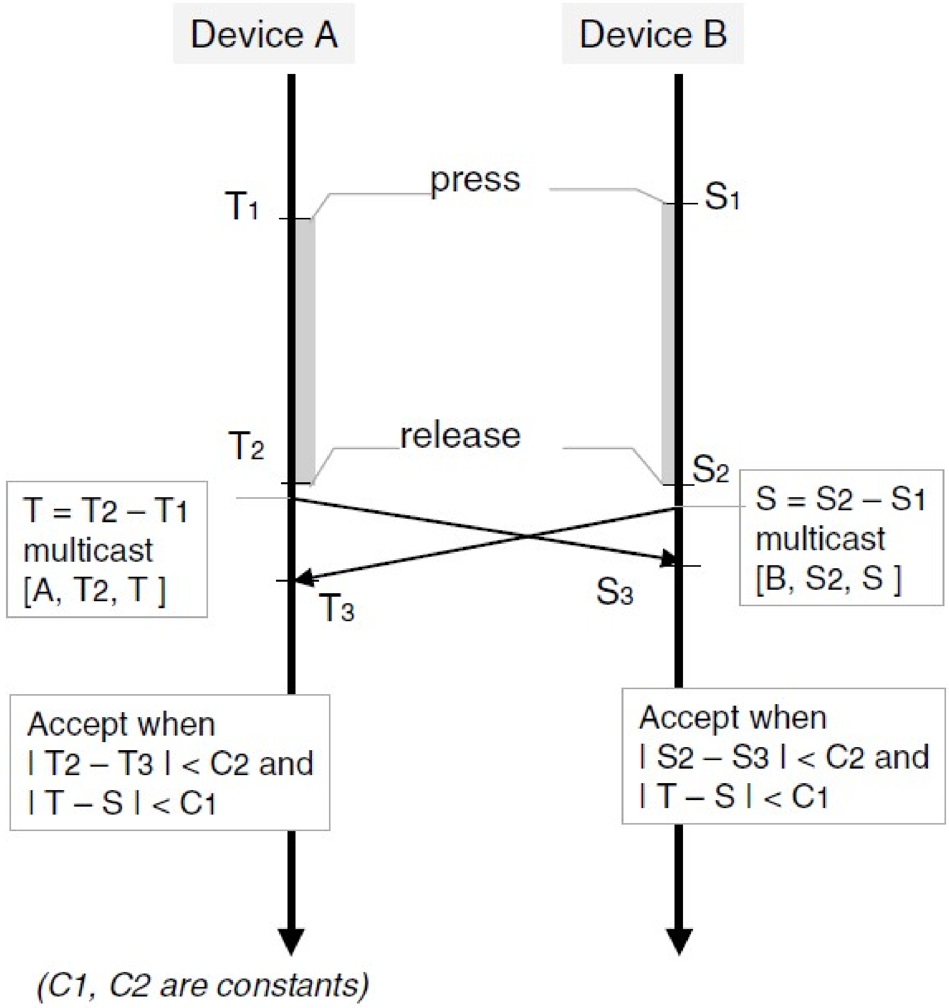


Figure 4.16: The packet exchange protocol of *SyncTab*. [Rekimoto, 2004]

to establish a connection, the time interval between button press time (T1) and release time (T2) of A is T, the corresponding parameter for device B is S, with press time (S1) and release time (S2). The arriving time of B's packet to A is T3 and the one of A's packet arrives at A at S3. When the interval between T2 and T3 and the interval between S2 and S3 are both smaller than a predefined constant C2, and the time difference of T and S is smaller than a constant C1, the connection request from both sides are accepted.

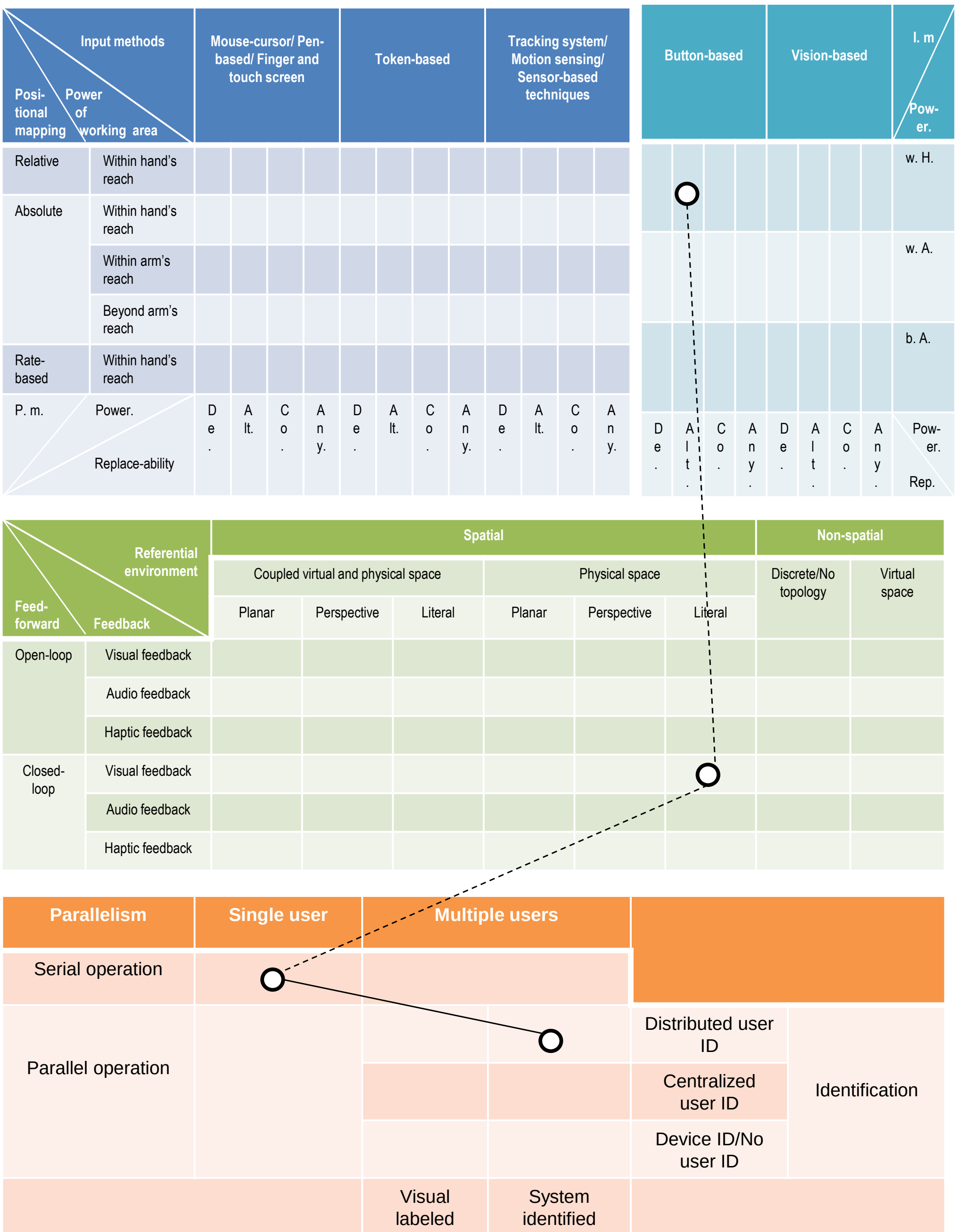


Figure 4.17: The Design Space of *SyncTab*

4.13 Synchronous Gesture

Information exchange via bumping the two devices

In 2003, Ken Hinckley developed the *Synchronous Gesture*, which enables users to stitch screens of two computing devices or exchange information between them [Hinckley, 2003]. Both functions are realized through bumping the edges of two devices, which are equipped with touch sensors and accelerometers. With bumping one device to another static one, the dynamic display tiling is realized, with which the image displayed on the dynamic tablet expands to the stationary one. As a result a temporary bigger display is created. On the other hand, two tablets can be bumped towards each other at the same time to realize mutual sharing of information. With mutual sharing, the content that each user is browsing respectively, such as a web page, will be sent to the other's screen.

How to identify different kinds of bumping.

The different kinds of bumping result in different accelerometer signatures, thus the system can identify the signal wave accordingly and distinguish those two situations. For the first feature, the left side edge of one device is hit by the right side edge (or in reverse), the two accelerometers of both devices sense the equal power but with opposite directions. The accelerometers signatures see the Figure 4.18, where the term 'local device' refers to the tablet the user holds at hand, and 'remote device' refers to the stationary tablet that is lying on the table. The accelerometers can sense movements from two axes: forward/backward, and left/right. In Figure 4.18, signal waves that colored green or blue stand for forward/backward, and waves that represent left/right movements are in black or red. A clear spike shows in the wave when the one side bumping happens, and there is not much influence to the forward/backward axis. By signal wave recognition the system can determine which device is the dynamic one and edge on which side of the device is the contacting edge, so that its display content will span both displays across the contacting edge.

On the other hand, we can see the difference in the 4.19 when mutual sharing of information is initiated. The wave signal represents the bumping action when two users move the devices simultaneously. Signals from all axes are affected and have fluctuation, and the force seems to be sim-

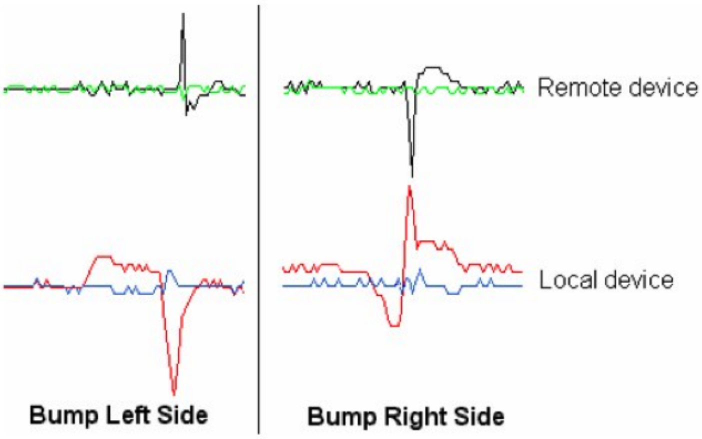


Figure 4.18: Accelerometer signatures for bumping one tablet into a stationary one, with forward/backward and left/right axes for the local(initiative) and remote(stationary) devices [Hinckley, 2003]

ilar and no spike appears.

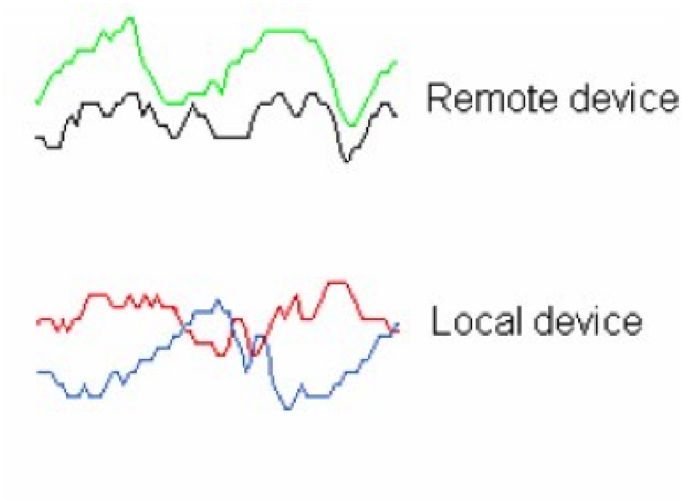


Figure 4.19: Accelerometer signatures for bumping two tablets into each other at the same time. [Hinckley, 2003]

Design space analysis

The *Synchronous Gesture* is a subdivision of synchronized technique. By detecting the synchronization of signal from specific sensors, a certain virtual connection can be established. As we narrow it down to the techniques we discussed in this section, unlike the *SyncTab* technique we introduced in the previous section, *Synchronous Gesture* is *sensor-based* technique. Thus the performance could be influenced by the wireless network connection, physical proximity and signal strength. The edges of tablets are equipped with two-axis linear sensors, which are of dedicated use as the proximity detection tools and hard to be replaced by other kinds. The bumping action is controlled by hands, thus the *power of working area* is *within hand's reach*.

With the help of *dedicated* sensors, it seems the virtual connection is established through physical display tiling. The involved devices are spatial referential as *physical space*. The corresponding input model type is *literal*, because the trigger event is a physical action, and no virtual representation of environment is required. The feed-forward is *closed-loop*, since the operation such as aiming at or directly manipulating the destination device is easy to finish within hands. To distinguish the two main features of *Synchronous Gesture*, different *visual feedbacks* are provided. When initiating the dynamic display tiling, as shown in Figure 4.20, two arrows with different sizes and pointing directions indicate the force and the moving direction from each device. For mutual sharing of information, the similar arrows appear on screens of both devices, but with the same size.

As described in the technical report, more than two tablets can be tiled one by one. For example, after the first two devices are connected via bumping, the third one can be tiled with them by only striking one of them. No user identification method is mentioned in the reviewed literature.

Similar techniques: *Stitching*, *Cooperative Gestures*

Similar techniques include pen stitching [Hinckley et al., 2004], which establishes connection by drawing a line along the digital surfaces of two adjacent devices; *Cooperative Gestures* [Gonzalo Ramos, 2009], with which the sender drags a file icon to the top edge of his/her screen and keeps holding it there, while the receiver can accept the file by dragging the same icon from his/her screen. The design space is visualized in Figure 4.21.

Figure 4.20: *visual feedback*s for the dynamic display tiling, which indicate the connection and disconnection of the displays. Top left and right: the device on the left side bumps into the one on the right side, the arrow with lager size indicates the connecting display. Down left and right, these two splitting arrows indicate that one tablet is moved away from the other one and the connection is broken.

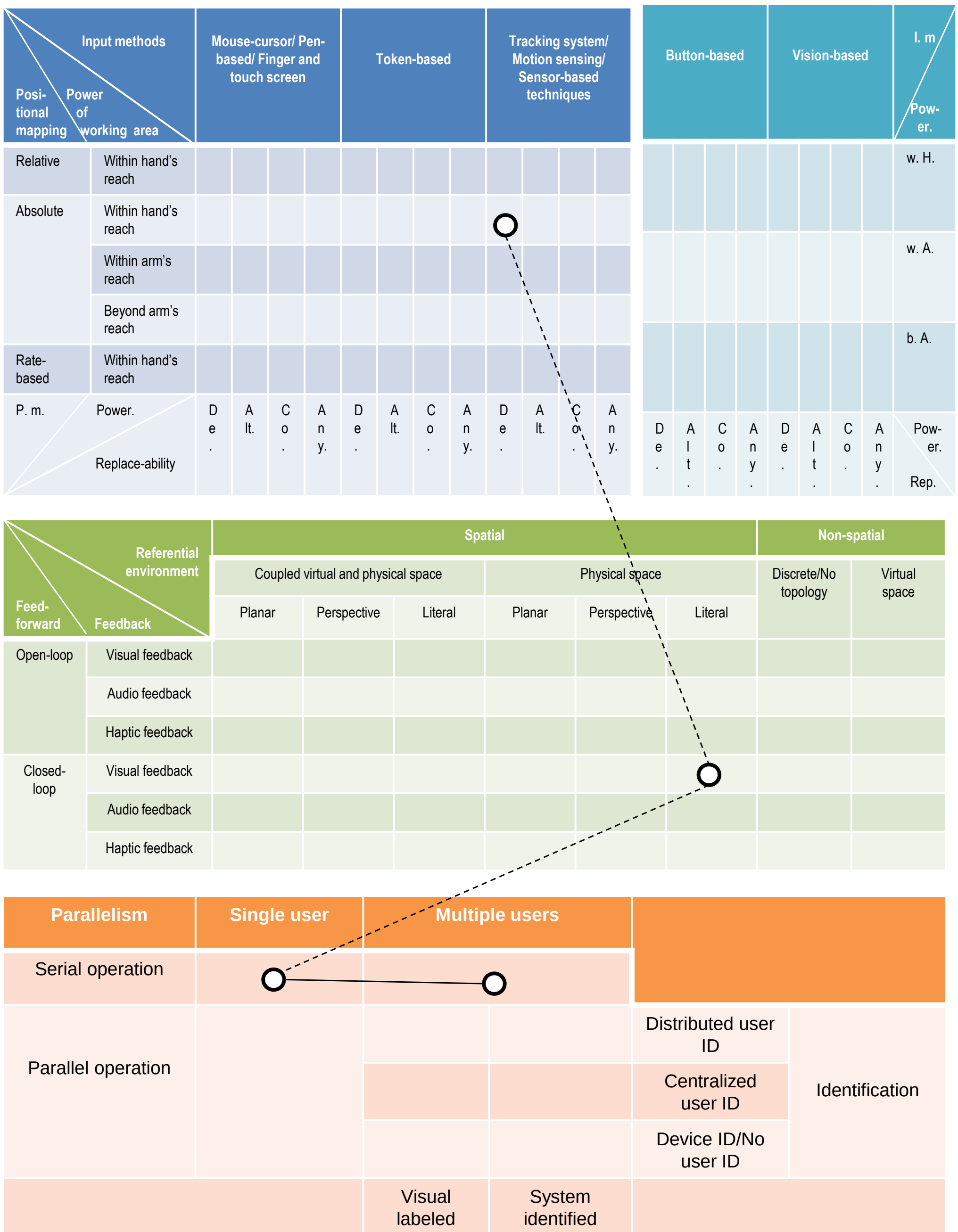


Figure 4.21: The Design Space of *Synchronous Gesture*

Chapter 5

Evaluation

In this thesis, we have illustrated the design space for depicting properties of interaction techniques, and described 13 major techniques particularly concerning the cross-device relocation task. To evaluate our work, we have two hypotheses:

Our two hypotheses for evaluation

H1: The dimensions of the design space (defined in Chapter 3) have influences on user preference of the technique.

H2: The difference on user preference for technique properties reflected by dimensions depends on whether they are from different subspaces.(The definition of subspace is introduced in Section3.3—“Design rational”)

To verify the above hypotheses, we designed a user test and develop the corresponding evaluation method. Basically our approach was to transform the qualitative problem into quantitative data measurement and analyze them statistically. To verify H1, we calculated the importance value of dimensions by analyzing the mean values of user survey outcomes. To verify H2, we applied the paired-samples t test to analyze the differences of user preference between dimensions. In addition we checked the reliability of all the results.

The aim of the user test

The aim of the user test is to provide users an experience of different interaction styles from the perspective of dimensions that we identified, and measure the user preference influenced by each dimension by means of user surveys.

5.1 Evaluation methodology

General methods of evaluation: observations, interest views, and questionnaire

The user test is a qualitative experiment that is run by performing user tasks using physical- or paper prototypes. However, to collect quantitative data we transform qualitative problems into scale questions in the questionnaire. Thus the user test can be considered as a hybrid form. User experience will be evaluated by observations, interviews, and questionnaire. During the user test, each user is asked to finish a series of tasks and his/her expressions and behaviors will be observed. Before performing the tasks and after filling out the questionnaires, interviews will be conducted to collect users' opinions. The questions about the following aspects will be asked: 1) user's prior experiences of the involved devices; 2) the emotions and special behavior that was shown by the user during the test; 3) explanations of some answers in the questionnaires.

The relocation task is divided into 3 subtasks.

To observe users more precisely, we subdivide the cross-device object relocation task into three subtasks: picking up an object on the screen, moving the object towards destination, placing the object into the target position. For each subtask, we chose one dimension to test at a time. By specifying instance of the corresponding sub-dimension, the user can obtain a perceptual knowledge of the involved properties of the mentioned interaction techniques.

After accomplishing the required tasks, the user needs to fill out a questionnaire, which will be introduced in the next section.

5.2 Design of questionnaire and data analysis methods

Our questionnaire can be found in Appendix B—"Questionnaires and list of tasks". The questions are concerned with user background information, the preference of different input devices or control mechanisms, the importance level of some feature settings, and the other factors that influence the user's choice of interaction techniques. The user is encouraged to make comments about usability aspect and the reasonableness of dimension settings. However, in order to get the user's real opinions, most of the questions are implicit.

Forms of questions

There are 3 different forms of questions in our questionnaires: choose one item from a list, text box, and scale. The first two kinds of forms are used to get information about users' backgrounds, get some dedicative answers about their choices or the problems they have during manipulations.

The results are mainly obtained via analysis of the scale questions.

Our experiment results are obtained mainly by the analysis of the scale questions, by which the importance of a certain property is measured on a standard scale of 1-5 (the preference level of 1 means the function of a technique in this aspect is essential to me, 5 means I don't care about this property at all).

The reason of using two question to represent one dimension

Every dimension in our design space corresponds to two scale questions (Q1 and Q2), and we call these two questions paired questions. The reason of using two questions is that each question describes an example in a use case, which reflects parts of the characteristics (sub-dimensions) of a dimension. In the questionnaire, if we only use one question to represent each dimension, users may have the understanding problems. Besides, more questions can depict different use contexts, which is also an influencing factor on user preference.

Verification of paired questions

To verify whether users are able to understand the correspondence between dimensions and paired questions correctly, we calculate the correlation between the paired questions. The correlation value between two items ranges from 0 $\tilde{1}$/0 ≈1. We only care about the absolute value of it. If the absolute value of the correlation is close to 1, it means that the two items are highly correlated, and the design of the paired questions is successful. Then we can sum the result of both questions and divide by two, and obtain the importance value of the corresponding dimension. If the importance value is smaller than 3, we can conclude that this dimension has influence on user preference of selecting a technique.

Calculation of importance value for dimension

5.3 Environment settings and prototypes

The user test is conducted in a lab room, where wireless network connection is provided. The following computing devices are involved: desktop computers, iPad 2, standard mouse, iPad capacity pen, and mobile phones. Besides, the physical prototypes of *Pick-and-Drop* and Synchronous Gestures, the digital or hand-draw paper prototypes of *Radar View*, *Passage*, *TractorBeam* are produced and applied during the test.

The use of physical prototypes

The use scenario is dependent on the dimension we choose. For example, when we test the dimension 'power of working area', the use scenario can be set as a co-located workspace, with different range of operation area. Generally speaking, the involved use scenarios include all the cases that we mentioned in Chapter 3, the introduction of relocation task.

For time and source limitations, we use low-fidelity physical prototypes, which are enough for supplying a rough overall understanding to the user. Besides, the hypotheses are more from the non-functional aspects, thus for the physical prototype no performance and other technical issues need to be tested. Considering that most users have no knowledge related to this field, we decide to let them experience the involved techniques by manipulating the Wizard-of-OZ prototypes.

Wizard-of-OZ prototypes

In the following parts of this section, the simulation methods of selected techniques and the reason of choosing those techniques are introduced.

Passage can be served as a very good example of the dimension replace-ability of input device. The concept of use anything as input is novel to users. This deep impression will help user better understand and remember the definition of replace-ability. To simulate the appearance of using this technique, we 'connect' two scales to the corresponding source and destination computer using wires. A USB flash drive is plugged in the source computers beforehand. The folder of disk space represents the data space of the passenger. When a user puts one of his/her belongings (this belonging serves as the medium device, the so-called passenger) on the scale, we drag the passage data space folder manually. Then the user can drag the target file into the folder and take this passenger to the destination. Similarly, when the user puts the passenger onto the scale at destination, we plug the USB flash drive in the destination computer and try not to let the user notice this action. The data space folder pops up and the user will get an illusion that the file is transferred via the passenger, which he/she can choose arbitrarily. The action sequence of applying the Passage prototype is shown in Figure 5.1. As we can see in this figure, the user chooses a bunch of keys as the medium device. During the user test, we have interesting findings about the selection of medium device, which will be introduced in 5.5.3.

Prototype of *Passage*

Prototype of *Pick-and-drop*

Pick-and-drop is the technique that fully utilizes physical environment and can be served as the example of power of working area (beyond arm's reach), referential environment (spatial, physical space) and input model type (literal).

To mimic the application method of this technique, we make two fake tablets by using materials of stiff paper, plastic film, and magnets. The input device is a pen stuck with a small magnet on the cap. The target object is represented by a paper icon sticks to a small magnet.

Here are two virtual tablets (source tablet and destination tablet), both of which have 3 layers: plastic film, layer1 and layer2. As we can see in Figure 5.2, layer1 is printed

Figure 5.1: Action sequence when applying the *Passage* prototype, a white scale is connected to the source computer, and a pink one is connected to the destination computer: a) A user puts a bunch of keys on the scale; b) the passage folder appears and she drags the target object into the folder; c) the screenshot of assigning information to the passage data-space; d) the user moves to the destination computer, and puts the same bunch of keys on the destination scale; e) the passage folder appears on the screen of destination computer, as well as the digital object the user assigns to this data-space.

with the same desktop wallpaper as layer2, only with a hole in the central position. The magnet attached to the pen (pen magnet) and the magnet behind the paper icon (icon magnet) are with the same polarity in source tablet, and opposite in destination tablet. Before experiments, to keep the file icon visible to the user, it is placed between the layer2 and layer1, exactly where the hole is. By touching the icon magnet on the source tablet with the pen, the icon is pushed away because of the 'Congeniality afoul'; the icon also could be pulled out by the pen on destination tablet due to 'isomerism attracted'.

The structure of the *Pick-and-Drop* physical prototype

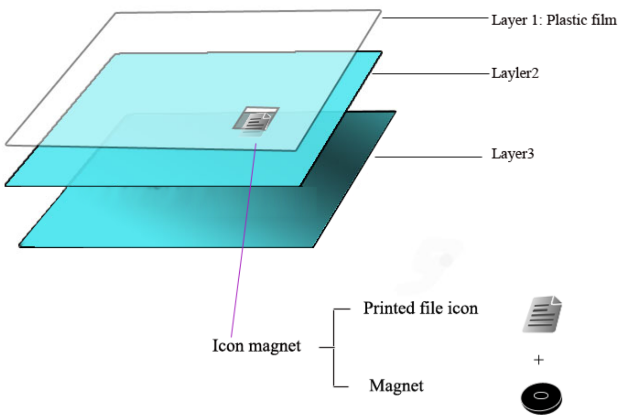


Figure 5.2: The design of *Pick-and-Drop* prototype

TractorBeam is a good instance of remote direct pointing technique. One of its main characteristics is the power of working area (within arm's reach). We use laser pointer to mimic the beaming device. The selecting feature is simulated by paper prototype: we print file icon on paper and attach it to a small wooden chip, which is tied with a string. When a user points at this paper icon for a few second, we consider it as an intension of selection. Then the user moves the laser pointer around and we drag the other side of the string to keep the paper icon following the laser spot.

Prototype of *TractorBeam*

Prototype of *Flick*

Flick is one of the rate-based controlled techniques. Although it is commonly used in browsing options on touch screen, most users are not familiar with this concept. As shown in Figure 5.3, a user is required to flick a real object (paper icon attached to a magnet) along the physical surface. The higher the flick speed is, the further the object will be thrown. This action also helps users to understand the definition of open-loop control, because it's hard to determine where the moving object stops.

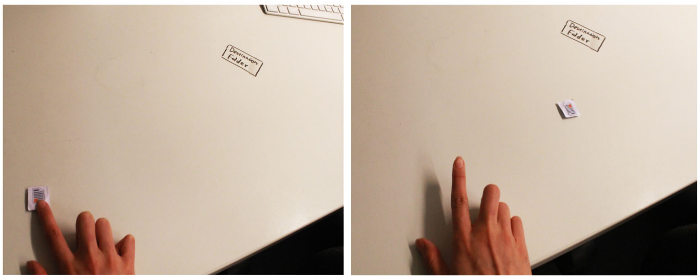


Figure 5.3: The use of *Flick* prototype: A user is flicking a real object to the 'destination folder'.

Prototype of *Synchronous Gestures*

Synchronous Gestures is selected due to its input method. It is an instance of sensor-based input method and bumping two tablets into each other seems to be a fancy data exchange approach. We use thick Plexiglas to make two tablets (with thickness it's easier to bump tablets into each other). For every tablet, beneath the Plexiglas there are two layers. The first layer in each tablet is a printed webpage. The image printed on the second layer is the same webpage of the first layer in the other tablet. After bumping the two tablets, we pull out the first layers of both tablets, and the webpage on each tablet is 'exchanged'.

Prototype of *Radar view*

Radar view is an example concerning spatial referential method. It provides users an intuitive understanding of utilizing spatial information, with the help of which users are able to aim at the destination device in a quick and easy manner. In addition, it also helps users to get the knowledge of input model type (planar) and identification method (*visual labeled* and *distributed user ID*).

Figure 5.4: Applying the prototype of *Synchronous Gestures*

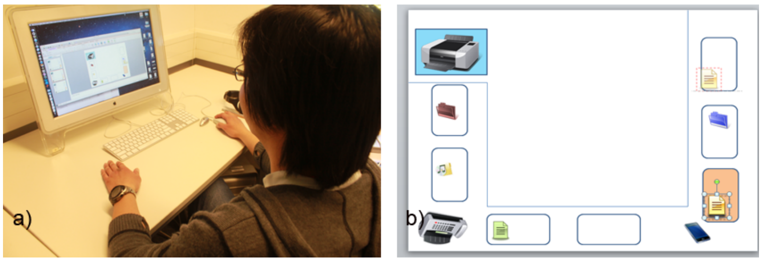


Figure 5.5: Applying the prototype of *Radar View*: a) a user is dragging a file icon to another computer on the map; b) the screenshot of the map.

To mimic the miniature concept, we illustrate an iconic map to depict the layout of our lab room via power point. The computers and involved peripheral devices are represented by thumbnails. The position correspondence between the real device and its digital representation is strictly kept. In the course of performing tasks, users are required to drag file icons across the representations of the devices on the map.

5.4 Procedures

The user test is composed of the following parts: 1) Introducing definitions; 2) demonstrating and experiencing the prototypes; 3) performing task on the list; 4) filling out the questionnaire; 5) interviewing.

1) Introducing definitions

At the beginning of the user test, the definition of cross-device operation and work principle of the involved techniques were introduced. To assist the user to get the picture quickly, hand-draw storyboards (in Appendix A—"Storyboard of using interaction techniques") are shown together with the verbal explanations.

2) Demonstrating and experiencing the prototypes

To help users to understand the questions and involved dimensions, we transformed the abstractive definitions into concrete instances. As described in Section 5.3—"Environment settings and prototypes", each prototype we used was an tool to get general knowledge about some dimensions. In this procedure, we demonstrated to the user how the prototypes worked, and let them play around for a little while. This experience helped the user to better understand the task on the list and the questions in the questionnaire.

3) Performing task on the list

After a short practice of using the prototypes, a list of tasks was given to the user, and he/she performed the required tasks in sequence. When prototypes were applied, some system responses were visualized and represented by paper drawing. The user was encouraged to 'think aloud' and the thoughts and behaviors were recorded by Note-Taking. During this procedure only the questions concerning how to use the prototypes could be answered. This procedure was the main part of user test and lasted for about 30 minutes. Each task was a very concrete example of a certain sub-dimension.

The list of tasks is presented in Appendix B—"Questionnaires and list of tasks". Taking the tasks from1.2.1 to 1.2.3 (group 1 'picking up an object', test dimension 2: positional mapping) as the examples.

Examples of evaluation tasks

For this task, the user was about to experience three kinds of pointing devices with different positional mapping types. As we defined in Chapter 3.1, the three kinds of mapping type are: absolute, relative and rate-controlled. Thus we chose three input methods in order to let the user experience the perceptual differences of them respectively. With absolute mapping, the user was asked to tap a file icon on the iPad touch screen (task 1.2.2). On the other hand, dragging a file icon to a file folder on the desktop via the standard mouse was an instance of applying the relative mapping device (task 1.2.1). By utilizing the flick gesture on the iPad screen, the user could make a quick stroke on the desktop to browse the icons and find the desired one (task 1.2.3). (Although the action 'picking up the desired object' was realized by tapping on the icon, browsing the relevant candidates was also an important part of the selection.) In this way, the input method with rate-controlled mapping was instantiated.

4) Filling out the questionnaire

After performing all the tasks, the user was asked to answer the questionnaire. As we introduced in Section 5.2—"Design of questionnaire and data analysis methods", the answers of most questions are measured with standard scales. From this procedure we obtained the quantitative data and were about to further analyze them statistically.

5) Interviewing

In the end we had a brief interview with the user according to the results of the observation and findings in the questionnaire. Users were encouraged to make comments about their experiences and the involved techniques. The results were noted down and served as our qualitative data source.

5.5 Result and findings

5.5.1 Participants

Considering the limitation of hardware and software, we only tested parts of the sub-dimensions. 24 participants took part in the user test. They are all students in university, studying computer science, mechanical engineering, metallurgy, and music, etc. 5 of them are female and 19 are male. They are from Asia and Europe, aged between 19 and 34. Despite most commonly used devices such as standard mouse and desktop computers, all of them have experience to use touch screen. On the other hand, before this user test none of them has any knowledge of the technique that we prototyped in the experiment.

5.5.2 Data analysis

Dimension 1 (input method) is evaluated by other methods

The dimension 1 (input method) is hard to be evaluated by scale questions. Considering the outcome of interviews and observations, many users showed interests about the novelty of input method. The question 1.1.1 was asked to choose users' preferred input methods. In Figure 5.6 we can see that *finger and touch screen* gained popularities. One of the main reason is that 'what you see is what you manipulated', i.e. the direct manipulation and physical contact with the object on the screen reduce the operating difficulties. In collected answers for the question 1.1.4 (What other aspects will influence your choice of input method?), the requirements such as 'the ease of use', 'direct manipulation', 'multi-functions' are frequently mentioned. Besides, the results of question 3.3.2 (When transferring a file to other user's computer, which method do you prefer?) showed that, most users chose the option 'cloud technique', because it realized the independency of users' locations. In a word, rather than considering how this input method works, users cared more about what this input method was capable of, which could be depicted by other dimensions.

As introduced in Section 5.2, we applied scale questions

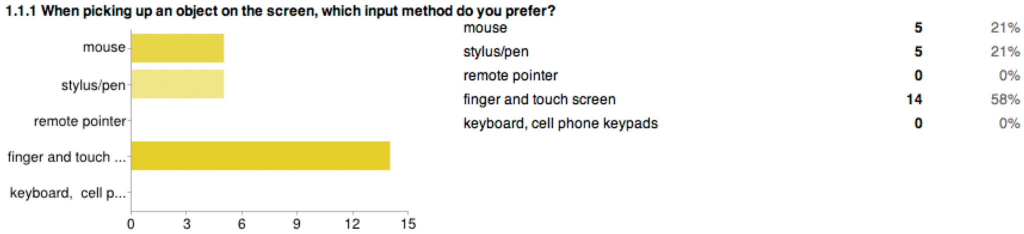


Figure 5.6: Result of choosing the user's favorite input method

to measure the user preference influenced by dimensions. To verify whether users were able to understand the mapping between paired questions and the depicted dimension, we applied Pearson's correlation to test the association between Q1 and Q2. If a user gets the idea correctly, there should be some linear dependency between the answers, i.e., for every participant, considering one dimension Di (i= 2 to 10), if the answer of Q1 increases, the answer of Q2 increases or deceases by a proportionate amount. Pearson's correlation coefficient reflects the degree of the linear relationship between two variables. The correlation coefficient ranges from +1 to -1, and here we only care about the absolute value of it. If we find out that for one dimension, the absolute value of the correlation coefficient is close to 1, it means that Q1 and Q2 are highly correlated and the design of questions is successful. Here we apply 0.5 as the lowest acceptable level of the correlation coefficient.

We applied Pearson's correlation coefficient to test whether users can understand the mapping between paired questions and the depicted dimension.

To make sure that our results are not occurred by chance, we tested the 2-tailed probability p (Sig. 2-tailed) of every result, because the hypothesis that whether Q1 and Q2 for Di are correlated is non-directional. Here we take the 0.05 as the criterion for significance since it is commonly adopted, i.e., if the p of an obtained result is smaller than 0.05 in a 2-tailed test, we can conclude that this result is significant and is not a chance finding.

We applied two-tailed test to test the significance of correlation.

After the user test we calculated the correlation coefficient for every dimension and the corresponding p (shown in Table 5.1). The results showed that most of the obtained correlations of paired questions could be accepted except for D3 and D9. Therefore we calculated the importance value (introduced in Section 5.2) of each dimension. From the re-

Our first hypothesis is verified.

sults, the obtained importance values of all dimensions are smaller than 3, which means that users do care about those properties reflected by most of our dimensions (D2, D4, D5, D6, D7, D8, D10), and our first hypothesis is verified.

	Mean of Q1	Mean of Q2	Std.Dev. of Q1	Std.Dev. of Q2	Correlation	Sig. (2-tailed)	Importance value
D2:positional mapping	2.208	2.083	.977	.974	.620	.001	2.146
D3:power of working area	2.083	1.917	.880	.776	.202	.345	2.000
D4:replaceability of input device	2.292	1.958	.999	.908	.5417	.006	2.125
D5:referential environment	2.625	2.875	1.135	1.262	.512	.010	2.750
D6:input model type	2.917	2.792	1.442	1.215	.536	.007	2.850
D7:feedforward	2.167	1.750	.868	.847	.532	.007	1.958
D8:feedback	1.250	1.208	.532	.415	.542	.006	1.229
D9:parallelism	2.375	1.917	1.469	.929	.279	.187	2.145
D10:identification	1.375	1.750	.711	1.260	.546	.006	1.563

Table 5.1: Correlation between paired questions, mean and standard deviation of Q1,Q2. Sample size: N=24

As shown in Table 5.1, the different importance value indicates that the user preference influenced by dimensions is different. To verify our second hypothesis (which is also non-directional) and the reliability of our results, we applied the paired-samples t-test to test the difference between every two dimensions. In our user test, each user was asked to answer two questions to express their preference level towards one dimension. Thus comparing the difference of user preference between dimensions is equivalent to comparing the mean differences of 48 answers for both dimensions. Again, we take 0.05 as the criterion for significance in the 2-tailed test. The most important information of the test result is shown in Table 5.2, which tells us how large the mean differences between the two dimensions are and whether the results are obtained by chance.

Calculating difference on user preference between dimensions

	D2	D3	D4	D5	D6	D7	D8	D9	D10
D2	/	-.146 /.368	.02/ /.888	-.604 /.001	.708 /.001	.188 /.361	.917 /p<0.001	0 /1	.583 /.004
D3	/	/	-.125 /.452	-.75 .001	-.854 /p<0.001	.0417 /.772	.77 /p<0.001	-.146 /.488	.438 /.018
D4	/	/	/	-.625 /.009	-.729 /.002	.167 /.425	.896 /p<0.001	-.02 /.916	.563 /.001
D5	/	/	/	/	-.104 /.627	.792 /.001	1.52 /p<0.001	.604 / .009	1.188 /p<0.001
D6	/	/	/	/	/	.896 /p<0.001	1.625 /p<0.001	.708 /.001	1.292 /p<0.001
D7	/	/	/	/	/	/	.729 /p<0.001	-.188 /.394	.395 /.060
D8	/	/	/	/	/	/	/	-.917 /p<0.001	-.333 /0.19
D9	/	/	/	/	/	/	/	/	.583 /.015

Table 5.2: Mean differences/Sig(2-tailed) of all paired-samples

From the obtained data we can conclude that the differences within the following pairs are significant (only those results whose p of 2-tailed test are smaller than 0.05 are accepted, thus none of them is a chanced result):

D2 and D5, D2 and D6, D2 and D8, D2 and D10;

D3 and D5, D3 and D6, D3 and D8, D3 and D10;

D4 and D5, D4 and D6, D4 and D8, D4 and D10;

D5 and D7, D5 and D8, D5 and D9, D5 and D10;

D6 and D7, D6 and D8, D6 and D9, D6 and D10;

D7 and D8;

D8 and D9;

D9 and D10.

The results of most pairs verified our second hypothesis.

Among those results, we notice that the two dimensions that have significant difference are all from different subspaces except for those underlined pairs. For example, the user preference on D2 (positional mapping) is significantly different from those caused by D6, D8, and D10, because in users' opinions, the choice of mapping types didn't affect how the spatial information was applied (D5 and D6); which kind of feedback should be provided (D8); or how can simultaneous operation be performed and how users can identify themselves(D9 and D10). D2, D3, D4 are from the same subspace and are all designed to depict the properties concerning input methods. The test result is consistent with our subspace division.

Here we need to explain the results of dimensions from the second subspace (referential environment and control mechanism). This subspace is initially composed of two parts, D5 and D6 depict how the spatial information is utilized; D7 and D8 describe how the feedback is applied to provide efficient control. Those two aspects are different in nature. They are combined formally (in one table), but actually they can be considered as two subspaces.

Analysis of the unexpected results: the large differences between 'D7 and D8'

The differences between 'D7 and D8', and between 'D9 and D10' are unexpected large. The dependency between D7 (feed-forward) and D8 (feedback) is not apparent to the users, although it is obvious to the developers. In the interview records, most users expressed that they didn't understand the definition of feed-forward, although in the questionnaire we gave concrete example to this term(E.g. I need the continuous feedback that can indicate the final position of the object when I am moving it.). The reason of this outcome is that most techniques we commonly used for relocation task are *closed-loop* controlled. The *open-loop* controlled technique such as *Flick* is more applied as a browsing gesture. Users couldn't see the relation between feed-forward and feedback directly. However, the importance value of these two dimensions are very small, which means that they all have large influence on user preference. It also indicates that they are still related to some extent.

Similar reason could explain the test result between D9 (parallelism) and D10 (identification). To developers D10 is apparently dependent on D9, i.e. the identification is required when multi-user operation can be performed. However, the experiment result showed that comparing with D9, users were more familiar with the instances of D10, therefore, they were able to get a clearer picture. The importance value of D10 is the second smallest, which means users cared about this property very much. According to interview record, most users knew that the identification is required under the situation of multi-user operation, but they couldn't associate D9 with D10 directly due to the lack of multi-user operation experience (such as working in a collaboration environment).

Analysis of the unexpected results: the large differences between 'D9 and D10'

5.5.3 Discussion and findings

Correlation of paired questions

It should be noticed that the correlation coefficient of the paired question corresponding to dimension 3 (power of working area) and dimension 9 (parallelism) are very low. From the collected answers of interview, we can attribute this result to the dependency of the context. In both questions that correspond to the dimension 'power of working area', the context description is not concrete enough. As a result, some users showed different attitude concerning the similar questions. They explained that if the use context was doing a representation, they didn't want to change their physical location to access a remote device. On the other hand, if they were in office and handled daily works, they didn't mind moving around in the room. Similar reasons also can explain the results of dimension 9 (par-

Dimension 3 (power of working area) and dimension 9 (parallelism) are context dependent allelism). In addition, most users are familiar with single user operated systems only, such as personal computing devices. And multi-users operation is not common in our daily lives (the most commonly used example is the instant-messaging based techniques, which support multiuser parallel operation).

Possible reasons for the moderate correlation

On the other hand, for those accepted paired questions, only moderate correlation (between 0.5 and 0.8)has been observed. According to the interview we can conclude that it's partly because of the context dependency as well. Besides, there are two other possible reasons:

Loss of information

1) The user preference towards a certain property is a subjective factor, which is unstable in nature. To provide an objective measure of the importance value of each dimension, we mainly applied the transformation approach (qualitative problem to quantitative measure). However, the transformation caused loss of information inevitably, which somehow could make users misinterpret the dimensions and therefore influenced the correlation coefficient.

Inconsistency of using preference levels

2) To some users, the preference level of 1 or 2 meant the same (also held for 4 or 5). However, they might give different preference levels to the properties that they thought were equally important. As a result, the inconsistency of using preference levels brought negative influence to our correlation analysis.

Feedback

The obtained importance values show that D8 feedback is the most important dimension to users (as shown in Table 5.1, the importance value of D8 is 1.229). During interviews this term is intensively discussed as well.

The form of feedback is discussed during the interview.

Visual feedback was preferred by most users due to its expressiveness and efficiency. On the other hand, *audio* and *haptic feedbacks* were considered to be less reliable and sometimes even annoying (especially when multiple users). Three users believed that a combination of visual and audio feedback could be a better choice, because it freed users' eyes to some degree and reserved the efficiency of obtaining visual information. One of them pointed out that although audio feedback can express nearly the same information content as visual one, it took longer time, which made it less popular comparing with the visual one.

The context is also an influencing factor on the necessity of feedback in users' opinions. When using techniques that utilize spatial information, if the moving distance is small, users only care about the feedback at destination. If the distance is large, the feedback is required throughout the procedure.

Use context also influences user requirement on feedback

Touch screen

Being selected as the most favorable input method, what attracted the users to *finger and touch screen* were: 1) users can have direct contact with the manipulated object; and 2) this input method required no additional tool (The use of capacity pen is optional. Usually it is applied to provide users the feeling of using a real writing tool.) However, one user didn't like it due to the lack of haptic feedback. Two users stressed the importance of smooth operation, i.e. if there was time lag between the movement of finger and the manipulated object, they would not choose this input method.

A flaw of the most popular input method–touchscreen: the lack of haptic feedback

Identification

Identification was considered as the second important dimension by users(as shown in Table 5.1, the importance value of D10 is 1.563). During the interview, three users said that they liked the idea of keeping an account setting when multiple users applying the same technique simultaneously. However, they also expressed the annoyance and impatience of frequent logon. Even though the logon was required only once for every first use device, it's still annoying to some users. Therefore, we suggested applying more simple and integrated method. For example, when a user logged on a device for the first time, he/she could specify other trusted devices in the workspace, such as the view rectangle applied in *Radar View*. Those selected devices were considered as the authorized ones, and the logon information was automatically transferred when cross-device operation was performed.

The trade-off between keeping account setting and simple logon

The selection of medium device

The stability and reliability of medium device are important to users.

Although most users considered *Passage* as an interesting technique due to its flexibility, the reliability of the medium device was in doubt. One user was intent to use one pack of gum as the medium device, but he realized that it was not proper, because the weight might not be a constant when someone drew a piece of gum from the pack. During the interview, when talking about *Passage*, other users had similar doubts and asked if food was selected as the medium device, its weight could reduce when someone ate it. Besides, if two or more people selected very common objects, such as iPhones as their medium devices, the weight of any of them was not unique and could not be an identifier anymore. These problems brought more restrictions to the selection of medium devices, which made the idea of 'use anything as input' less attractive.

Chapter 6

Summary and future work

6.1 Summary

Our work is basically a theoretical analysis of the technical reports concerning the design concept and usability of the selected techniques. After introducing the background of cross-device interactions in Chapter 1—"Introduction", we reviewed the literature of classification and involved techniques in this field in Chapter 2—"Related work". During the study of the early work done by other researchers, we found out that there is a lack of the exclusive expressions or representations for describing all the important properties of the existing interaction techniques, especially for those newly developed ones.

On the basis of Nacenta's work [2009] we explored more dimensions, reasoning the meaning and expressiveness of each sub-dimension, and selected the most important 10 dimensions in Chapter 3—"Design space", which were: Input method, positional mapping, replace-ability of input device, power of working area, referential environment, input model types, feed-forward, feedback, parallelism, and identification. With those dimensions the primary and special features of major interaction techniques were described with consistency and clarity. In addition, we divided the

10 dimensions into 3 groups. With the dimensions in the same group, a sub design space could be formed. All the elements of the illustrated design space, including line notations representing the interrelation between properties nodes, were depicted and explained in detail. An example Figure 3.7 of design space was presented as well.

In Chapter 4—"Techniques supporting cross-device object movements", we described the selected 13 interaction techniques that supported cross-device object relocation task. Apart from introducing the features and working principle, we put each technique into our design space, and fitted all the important properties into the corresponding dimensions in the unified design space. Rather than deriving from past literature directly, some results were inferred from our analysis according to the dimensions we defined before. We emphasized the interrelation between properties, i.e. the change of one property would cause other alterations concerning usability.

In the Chapter 5—"Evaluation", we proposed two hypotheses, and designed the user test for collecting evidence. During our user test, observation, interview and questionnaires were applied. The user test aimed at providing users an experience of Wizard-of-OZ prototypes, which helped them to better understand the meaning of every dimension by means of performing user tasks. By instantiating each dimension and offering standard interval as the measurement in the questionnaire, we transformed the qualitative problems into quantitative measure. The results were presented and our two hypotheses were verified.

Contributions of the work

The most important contributions of this thesis are to provide a classification for interaction techniques by means of creating a comprehensive design space and to provide a systematic way to analyze the overall properties of techniques supporting cross-device object relocation task. Considering the completeness and expressiveness of our design space, no such work has been done previously. Besides, this work proposed a useful evaluation method to measure the user preference on techniques. And so far, the user test verified the rationality of our design space.

The advantages of our classification and design space are reflected in two main points: first, the division of sub design space indicates the possibility of comparison between techniques, i.e. we can't simply compare any two interaction techniques, only the properties within one sub design space can be compared respectively. Second, the interrelation between dimensions and sub-dimensions is visualized. Within a two-dimensional tabular, the property node can be filled into the corresponding intersection of columns and rows, and are connected with different kind of lines according to their relation types. The notations can be easily understood because of the intuitive mapping between meanings and expressions. Moreover, the use of the notations and representations of our design space is not limited by a specific task. With the scalability of each sub design space, the scope of use can be extended on demand.

The advantages of our design approach

6.2 Future work

Since the techniques develop very fast, to keep our design space up-to-date, there is still much work could be done in the future.

First, we should redo the user test to get more convincing data. To obtain more reliable data from evaluation process, the questionnaire needs to be modified. To evaluate a certain dimension, the corresponding questions should be highly correlated. In addition, more users should be involved in order to get lager sample size. In this way we can get more accurate estimate of user preference and thus improve the evaluation.

Improve the user test

Second, our work only proves partly of the design space, and many conclusions are inferred from literature without convincible data and evidence. For example, from the literature review we found out that the parallelism was an important attribute in multi-user environment, but after visualizing the selected techniques via our design space, we could obviously see that this attribute was not mentioned or not detailed formulated in the technical report of the corresponding techniques.

Test the parallelism properties of interaction techniques to refine the design space analysis

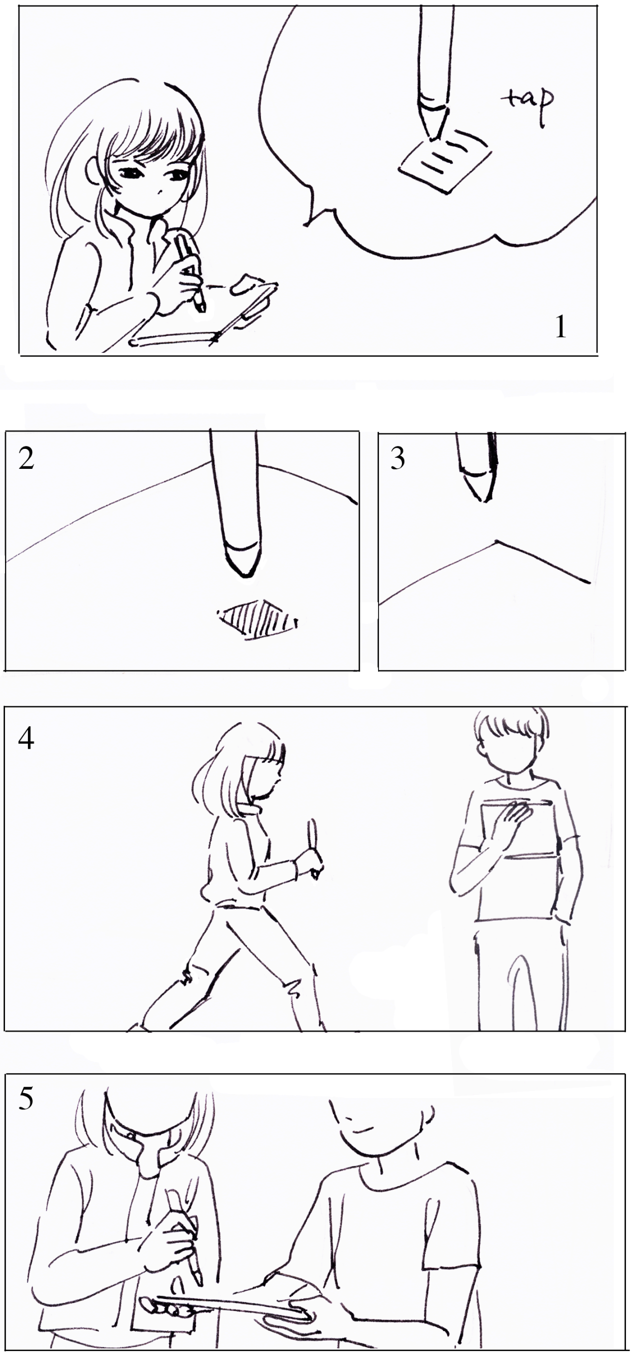
Therefore, one of our future directions is to test the parallelism properties of interaction techniques. To get more convincing evidence and data, comprehensive user test should be designed and validated. In this way the criteria of parallelism space could be refined, thus the design space can be improved.

Study the techniques that have application potential.

Third, during the study we found some gaps in the current interaction technique designs, those areas may become the possible research direction of our work. For example, as input device, pen/stylus is commonly applied by developers. Some techniques improve the usability of standard mouse and stylus, which allows the user to utilize the manipulation method they already have. However, the input device with *rate-based* mapping type is rarely used. We attribute this gap to the manipulative difficulties and the lack of accuracy, but in future work those problems should be overcome. We will concentrate on the research for fully utilizing the techniques that have application potential such as the *rate-based* controlled technique.

Appendix A

Storyboard of using interaction techniques



1. Select a file by tapping the pen on the file icon.
2. lift the pen (not so far away from the screen) and a shadow indicates that the file is bounded with the pen ID.
3. The pen moves away and the shadow disappears.
4. The sender moves towards the destination computing device.
5. Tap the destination screen and the file is pasted.

Figure A.1: Use *Pick-and-Drop* to place a file to another user's tablet

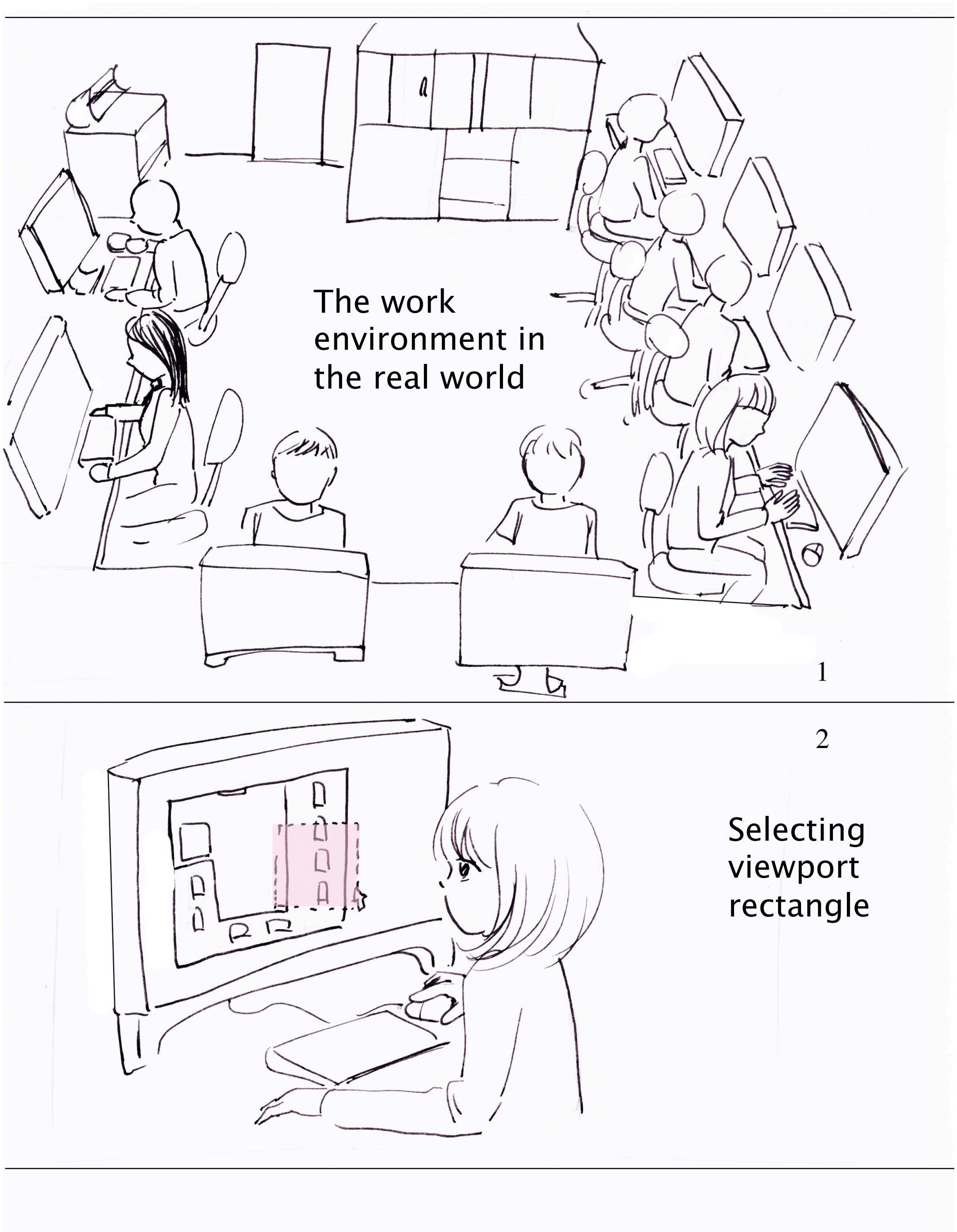
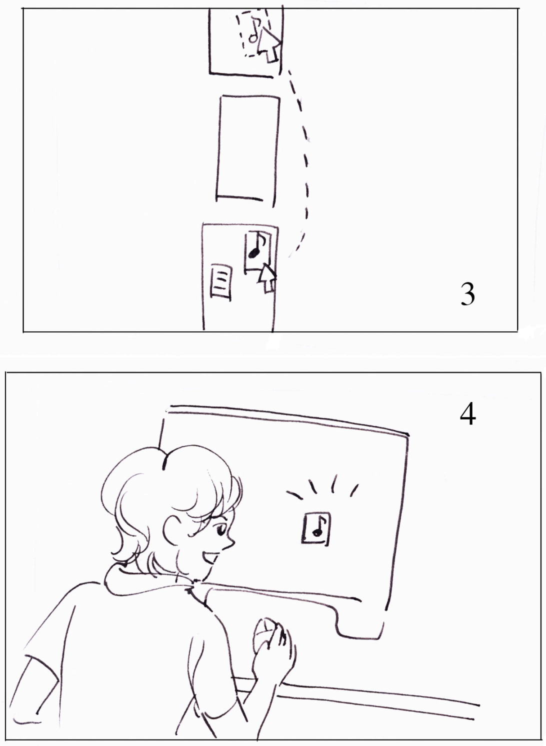


Figure A.2: Use *Radar View* to place a file to another user's computer



1. Our work environment

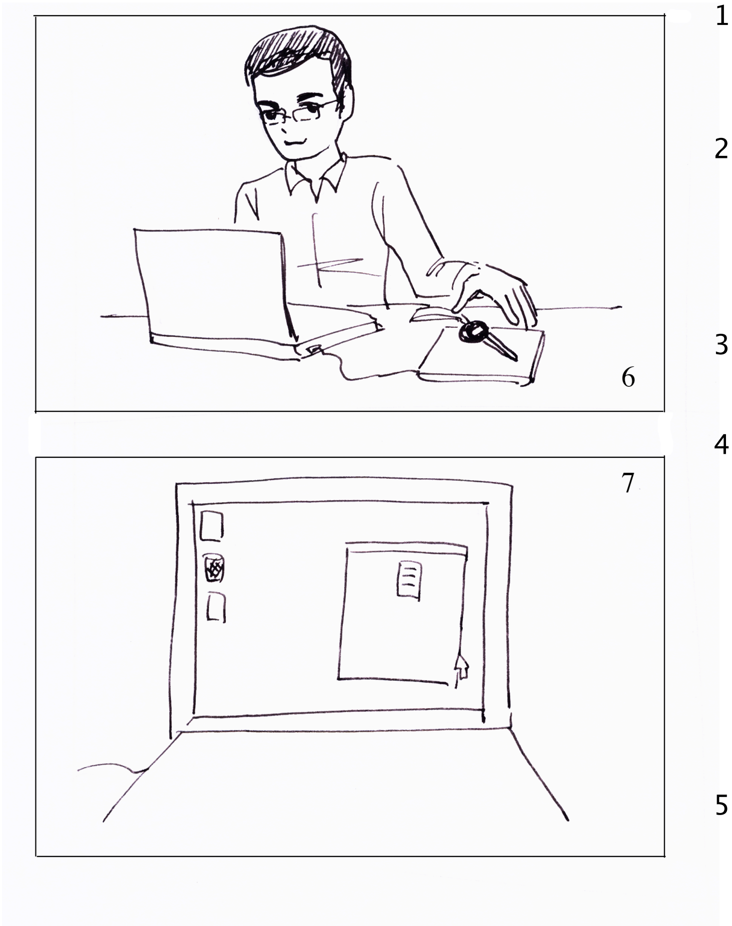
2. The workspace is represented via map, and the user selects her control area (viewport rectangles)

3. The user drags the file icon to the destination on the map.

4. The file is transferred to the destination computer.

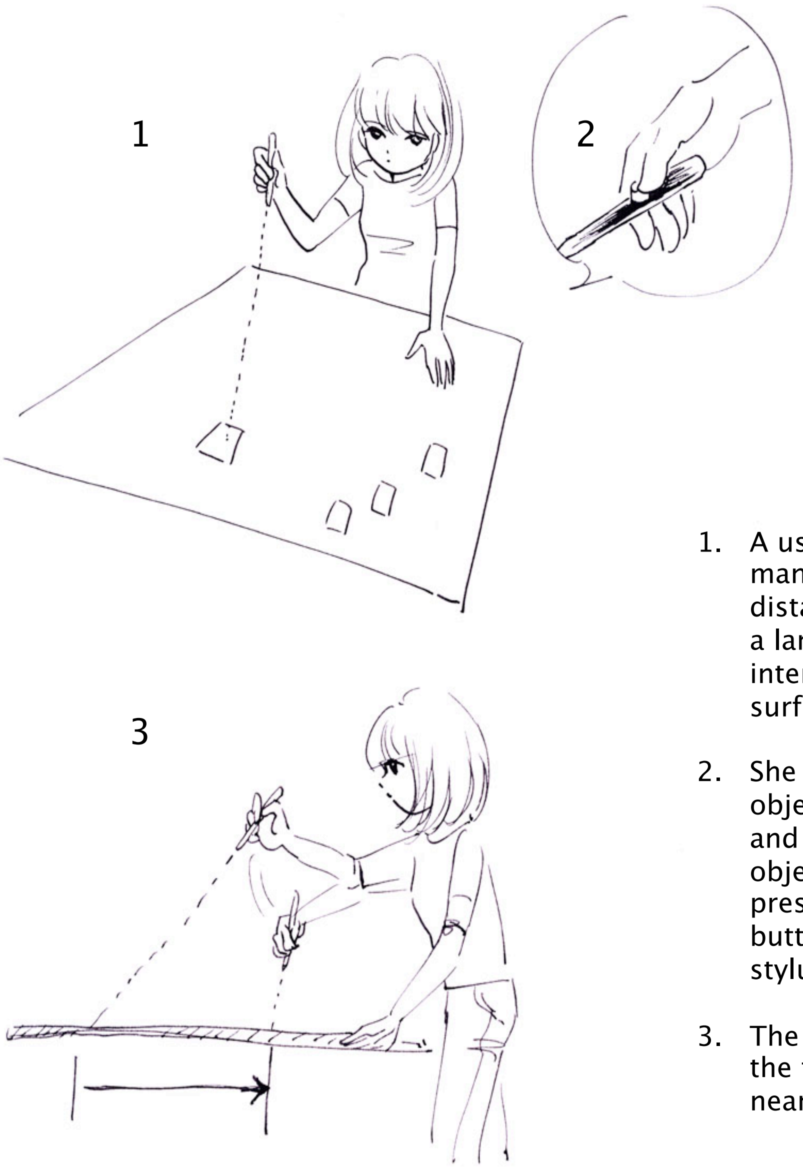
Figure A.3: Use *Radar View* to place a file to another user's computer

Figure A.4: Use *Passage*to place a file to another user's laptop



1. A user wants to transfer a file to her colleague.
2. She searches in her pocket and wants to find something she carries with,
3. And she finds a key.
4. She puts the key on the bridge (an electronic scale) and drags the desired file icon to the data space of the key. The weight of the key becomes the identifier.
5. Then she gives the key to the her colleague.
6. 7. The colleague puts the key on his bridge and gets the file.

Figure A.5: Use *Passage*to place a file to another user's laptop



1. A user wants to manipulate the distant object on a large interactive surface.

2. She points to the object via beam and selects the object by pressing the button on the stylus.

3. The user drags the file icon to a nearby position.

Figure A.6: Use *TractorBeam*to move distant object to nearby position

1. Two users are viewing different webpage on their own tablets.
2. They bump their tablets.
3. The two webpages are exchanged (to each user, the webpage viewed by the other user is added to the browser)

Figure A.7: Use *Synchronous Gesture*to exchange webpages between two users

Appendix B

Questionnaires and list of tasks

Questionnaires

User profile

Gender ☐ Male ☐ Female	**Age** ☐ Under 18 ☐ 18~23 ☐ 24~28 ☐ 29~33 ☐ More than 33
You are majored in? ☐ Computer science ☐ Non-computer specialty	

Task1: Picking up (selecting) a digital object (a file icon) on the screen.

1.1 Input method

1.1.1 When picking up an object on the screen, which input method do you prefer? (Choose one option from the list)

	Mouse
	Stylus/pen
	Remote pointer
	Finger and touch screen
	Button-based (keyboard, cell phone keypads)

1.1.2 Your prior experience of the device (your choice in the last question) is? (Choose one option from the list)

	I use this kind of device all the time and can use it with skill.
	I used this device several times.
	It's the first time I hear/use it.

1.1.3 How important do you think the following properties an input device should have?

Figure B.1: Questionnaires of user test, page 1

B.1 Questionnaires

1.1.3.1 With this input device I can access remote target without moving around.

	1	2	3	4	5	
Very important						I don't care

1.1.3.2 With small effort (short movement of my hands), I can have big control ability (the pointing device can reach far away position)

	1	2	3	4	5	
Very important						I don't care

1.1.3.3 It can be replaced by other commonly used devices, thus brings flexibility to my choice.

	1	2	3	4	5	
Very important						I don't care

1.1.3.4 With this device, I can directly utilize the physical environment. (having direct contact with the screen and the object to be moved)

	1	2	3	4	5	
Very important						I don't care

1.1.1 What other aspects will influence your choice of input method?

1.2 Referential environment and control mechanism

1.2.1 If the target object is on other user's computer, how would you like to access that device? (The following questions provide several possible solutions. Please choose the importance level of those properties.)

Figure B.2: Questionnaires of user test, page 2

1.2.1.1 Direct manipulation of the involved computers, and avoidance of remembering the symbolic name of the computing device where the target object is located.

	1	2	3	4	5	
Very important						I don't care

1.2.1.2 I can use the physical knowledge and skill that I already have. For example, to select an object can be like picking up a real object in the physical world.

	1	2	3	4	5	
Very important						I don't care

1.2.1.3 If a map is used, the physical correspondence of the represented devices should be strictly kept.

	1	2	3	4	5	
Very important						I don't care

1.2.2 When you pick an object successfully, feedback should be provided to indicate this state. How important does feedback mean to you?

	1	2	3	4	5	
Very important						I don't care

1.2.3 Which kind of feedback does you prefer? (Choose one option from the list)

	Visual feedback
	Audio feedback
	Haptic feedback

1.3 Parallelism

Please choose the importance level of those properties in your opinion.

Figure B.3: Questionnaires of user test, page 3

I can apply this technique while other users are using it at the same time.

	1	2	3	4	5	
Very important						I don't care

When multiple users operate in a shared workspace at the same time, the identification method should be provided.

	1	2	3	4	5	
Very important						I don't care

I can distinguish my pointer from others by my eyes.

	1	2	3	4	5	
Very important						I don't care

To distinguish my operation from others, it would be better for every participant to have an account name that attached to a portrait or icon.

	1	2	3	4	5	
Very important						I don't care

Task2: Moving a digital object (a file icon) on the screen.

2.1 Chose the importance level of the following properties.
When moving an object on the screen, what properties you think the involved devices should have?

With this input device I can move the object to remote place without change my own location.

	1	2	3	4	5	
Very important						I don't care

Figure B.4: Questionnaires of user test, page 4

1.3.1 I can apply this technique while other users are using it at the same time.

	1	2	3	4	5	
Very important						I don't care

1.3.2 When multiple users operate in a shared workspace simultaneously, the identification method should be provided.

	1	2	3	4	5	
Very important						I don't care

1.3.3 I can distinguish my pointer from others by my eyes.

	1	2	3	4	5	
Very important						I don't care

1.3.4 To distinguish my operation from others, it would be better for every participant to have an account name that attached to a portrait or icon.

	1	2	3	4	5	
Very important						I don't care

Task2: Moving a digital object (a file icon) on the screen.

2.1 Chose the importance level of the following properties.
When moving an object on the screen, what properties you think the involved devices should have?

2.1.1 With this input device I can move the object to remote place <u>without change my own location.</u>

	1	2	3	4	5	
Very important						I don't care

Figure B.5: Questionnaires of user test, page 5

	Via MSN file transfer
	Via USB flash drive
	Via cloud technique
	I can tap my screen to copy the object, and tap his/her screen again to paste this object.
	I can drag this file icon to his/her display via a map.
	I can use the cross-display cursor and directly drag this file onto his screen.

3.3.3 Please choose the importance level of the following properties.

3.3.3.1 I need the continuous feedback that can indicate the final position of the object when I am moving it.

	1	2	3	4	5	
Very important						I don't care

3.3.3.2 To initiate an interaction, I don't want to wait until other users finished theirs.

	1	2	3	4	5	
Very important						I don't care

3.3.3.3 When I successfully put the object to the destination, feedback should be provided to indicate this state.

	1	2	3	4	5	
Very important						I don't care

Figure B.6: Questionnaires of user test, page 6

B.2 List of tasks

User tasks:

Task group 1: picking up an object

Test dimension 1: input method

1.1.1: Select a file icon via tapping the capacity stylus on the screen of iPad.

1.1.2: Select a file icon via tapping the finger on the screen of iPad.

1.1.3: Select a file icon via clicking the mouse on the screen of a desktop computer.

1.1.4: Pick up an object by taking a picture, using camera on the mobile phone.

1.1.5: Point to the file icon and select it using the Tractor-Beam paper prototype.

1.1.6: select a picture in the gallery of a mobile phone using cell phone keypads.

Test dimension 2: positional mapping

1.2.1: relative mapping

Select a file icon via clicking the mouse on screen of a desktop computer.

1.2.2: absolute mapping

Select a file icon via tapping the finger on the screen of iPad.

1.2.3: rate-controlled mapping

Flick the screen of iPad to browse icons and select by tapping fingers on the desired icon.

Test dimension 3: Power of working area

1.3.1: within hand's reach

Select a file icon via clicking the mouse on screen of a desktop computer.

1.3.2: within arm's reach

Point to the file icon and select it using the TractorBeam paper prototype.

1.3.3: Beyond arm's reach

Select an object from the remote tablet by tapping the pen on that surface, using the physical prototype of Pick-and-Drop.

Test dimension 4: replace-ability of input device

1.4.1: compatible device

Select a file icon via clicking the mouse on screen of a desktop computer.

1.4.2: alternative device

Select an object from the local tablet by tapping the pen on the surface, using the prototype of Pick-and-Drop.

1.4.3: use anything as input

Use an arbitrary object that in your pocket as medium device. Put it one the scale, and assign information to its data space, using the prototype of Passage.

Test dimension 5: referential environment

1.5.1: spatial referential method

Select a file icon by tapping it on the thumbnail of your computer on a map, using prototype of Radar View.

1.5.2: non-spatial referential method

Logon MSN messenger and select a file to transfer via mouse.

Test dimension 6: input model type

1.6.1: planar input model type

Select a file icon by tapping it on the miniature representation of your computer on a map, using prototype of Radar View.

1.6.2: literal input model type:

Select a file icon from the local tablet by tapping the pen on the surface, using the physical prototype of Pick-and-Drop.

Test dimension 8: feedback

1.8.1: visual feedback

Select an object by mouse clicking, note that the selected file name is highlighted.

1.8.2: audio feedback

Select an object by mouse clicking, note that there will be alert tone when the target is successfully selected.

Test dimension 9: parallelism

1.9.1: multi-user parallel operation

Select one file to transfer, using MSN messenger as the file transmission technique.

1.9.2: single user parallel operation technique

Select 3 files to transfer, using MSN messenger as the file transmission technique.

1.9.3: single user serial operation technique

Select a file icon via tapping the capacity stylus on the screen of iPad

1.9.4: multi-users serial operation

Bumping two tablets into each other using the physical prototype of Synchronous Gesture.

Test dimension 10: identification

1.10.1: distributed user ID

Select a file icon using the prototype of Radar View, note that you can identify yourself via portrait and colorable view rectangle.

1.10.2: centralized user ID

Logon MSN messenger and select a file to transfer.

1.10.3: device ID

Select a file icon using the prototype of Pick-and-Drop.

Task group 2: moving the object towards destination

Dimension 1, 2, 4, 5, 6, 9, 10 are tested in similar ways as the tasks done in task group 1.

Test dimension 1: input method

2.1.1: Move the selected file icon by dragging it using the capacity stylus on the screen of iPad.

2.1.2: repeat 2.1.1 but use finger.

2.1.3: Move the selected file icon via mouse on the screen of a desktop computer.

2.1.4: Move the selected file icon using the TractorBeam paper prototype.

2.1.5: Select a receiver (Bluetooth) from a list using cell phone keypads.

Test dimension 2: positional mapping

2.2.1: relative mapping

(repeat 2.1.3) Move the selected file icon via clicking the mouse on screen of a desktop computer.

2.2.2: absolute mapping

(repeat 2.1.1) Move the selected file icon by dragging it using the capacity stylus on the screen of iPad.

2.2.3: rate-controlled mapping

Move a real object by flicking it along physical surface.

Test dimension 3: Power of working area

2.3.1: within hand's reach.

Drag a file icon via mouse towards the thumbnail of the desired computer, using the map of Radar View prototype.

2.3.2: within arm's reach

Select and drag the file icon closer to you on the surface, using the Tractor- Beam paper prototype, note that your arm needs to be lifted.

2.3.3: beyond arm's reach

Select an object from the remote tablet by tapping the pen on that surface, and take the pen to other tablet, using the physical prototype of Pick-and-Drop.

Test dimension 4: replace-ability of input device

2.4.1: compatible device

(repeat 2.1.3) Move the selected file icon via clicking the mouse on screen of a desktop computer.

2.4.2: alternative device

Take the pen and walk towards the destination tablet, using the prototype of Pick-and-Drop.

2.4.3: use anything as input

Use the prototype of Passage. Take the chosen medium device to the destination.

Test dimension 5: referential environment

2.5.1: spatial referential method

(repeat 2.3.1)Drag a file icon via mouse towards the thumbnail of the desired computer, using the map of Radar View prototype.

2.5.2: non-spatial referential method

Use MSN messenger and transfer a file to the receiver.

Test dimension 6: input model type

2.6.1: planar input model type

(repeat 2.3.1)Drag a file icon via mouse towards the thumbnail of the desired computer, using the map of Radar View prototype.

2.6.2: literal input model type:

(repeat 2.4.2)Take the pen and walk towards the destination tablet, using the prototype of Pick-and-Drop. Test dimension 7: feed-forward

2.7.1: open-loop control method

Flick an object along the surface to change its location.

2.7.2: closed-loop control method

Drag an object to other position via mouse on the screen of desktop computer

Test dimension 8: feedback

2.8.1: visual feedback

Drag an object towards destination folder on the screen of desktop computer using a mouse, note that the file icon follows the cursor and becomes semi-transparent.

Task group 3: placing the object into the target position

Test dimension 1: input method

3.1.1: Drag the selected file icon to the folder 'office' using the capacity stylus on the screen of iPad.

3.1.2: repeat 3.1.1 but use finger.

3.1.3: Drag the selected file icon to the folder 'destination' via mouse on screen of a desktop computer.

3.1.4: Move the selected file icon using the TractorBeam paper prototype.

3.1.5: Send the selected picture to receiver (Bluetooth) by pressing cell phone keypads.

Test dimension 2: positional mapping

3.2.1: relative mapping

(repeat 3.1.3) Drag the selected file icon to the folder 'destination' via mouse on screen of a desktop computer.

3.2.2: absolute mapping

(repeat 3.1.1) Drag the selected file icon to the folder 'office' using the capacity stylus on the screen of iPad.

Test dimension 3: Power of working area

3.3.1: within hand's reach

Drag a file icon via mouse into the thumbnail of the desired computer, using the map of Radar View prototype.

3.3.2: within arm's reach

Drag the file icon to a position that is close enough to you, so that you can manipulate it with ease, using the Tractor-Beam paper prototype.

3.3.3: beyond arm's reach working area

Select an object from the local tablet by tapping the pen on that surface, and take the pen to the required remote tablet, tap the pen on that surface again, using the physical prototype of Pick-and- Drop.

Test dimension 4: replace-ability of input device

3.4.1: compatible device

(repeat 3.1.3) Drag the selected file icon to the folder 'destination' via mouse on screen of a desktop computer.

3.4.2: alternative device

repeat 3.3.3, using the prototype of Pick-and-Drop.

3.4.3: use anything as input

Use the prototype of Passage. Take the chosen medium device to the destination and put it on the scale that connected to the destination computer.

Test dimension 5: referential environment

3.5.1: spatial referential method

(repeat 3.3.1)Drag a file icon via mouse towards the thumbnail of the desired computer, using the map of Radar View prototype.

3.5.2: non-spatial referential method

Use MSN messenger and transfer a file to the receiver.

Test dimension 6: input model type

3.6.1: planar input model type

(repeat 3.3.1) Drag a file icon via mouse into the thumbnail of the desired computer, using the map of Radar View prototype.

3.6.2: literal input model type

repeat 3.3.3, using the physical prototype of Pick-and-Drop.

Test dimension 7: feed-forward

3.7.1: Place an object into the target position by open-loop control method

(Flick an object to the destination folder using physical prototype)

3.7.2: Place an object into the target position by closed-loop control method

(Drag an object into the target folder via mouse on the screen of desktop computer)

Test dimension 8: feedback

3.8.1: Place an object into the target device and receive visual feedback

(Send a file to other device via mobile phone and receive the prompting message)

3.8.2: Place an object into the target position and receive audio feedback

(Send a file to other device via mobile phone and receive the ringing tone)

3.8.3: Place an object into the target position and receive haptic feedback (Send a file to other device via mobile phone and receive the vibration feedback)

Bibliography

Dzimitry Aliakseyeu, Pourang Irani, and Andrés Lucero. Multi-flick: An evaluation of flick-based scrolling techniques for pen interfaces. In *In CHI '08: Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*, pages 1689–1698, 2008.

Dzmitry Aliakseyeu and Jean-Bernard Martens. *Sketch Radar: a novel technique for multi-device interaction*, volume 2, pages 45–49. 2006.

Patrick Baudisch, Edward Cutrell, Mary Czerwinski, Daniel C. Robbins, Peter Tandler, Benjamin B. Bederson, and A. Zierlinger. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In *INTERACT*. IOS Press, 2003.

Jacob T. Biehl and Brian P. Bailey. Aris: an interface for application relocation in an interactive space. In *Proceedings of Graphics Interface 2004*, GI '04, pages 107–116, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.

Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. The design space of input devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, CHI '90, pages 117–124, New York, NY, USA, 1990. ACM.

Tsung-Hsiang Chang and Yang Li. Deep shot: a framework for migrating tasks across devices using mobile phone cameras. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 2163–2172, New York, NY, USA, 2011.

M. Collomb and M Hascoët. Speed and accuracy in throwing models for large displays. In *HCI'2004*, volume 2, British HCI Group, 2004.

Henrik Gelius. Computer-supported collaboration using pick-and-drop interaction on handheld computers, 2003.

Andy Wilson Raman Sarin Gonzalo Ramos, Kenneth Hinckley. Synchronous gestures in multi-display environments. In *Human–Computer Interaction*, volume 24, pages 117–169, 2009.

Carl Gutwin, Saul Greenberg, and Mark Roseman. Workspace awareness support with radar views. In *Conference companion on Human factors in computing systems: common ground*, CHI '96, pages 210–211, New York, NY, USA, 1996.

Carl Andrew Gutwin. *Workspace awareness in real-time distributed groupware*. PhD thesis, Calgary, Alta., Canada, Canada, 1998.

Mountaz Hascoët. Throwing models for large displays. In *HCI'2003*, volume 2, pages 73–77, 2003.

Ken Hinckley. Synchronous gestures for multiple persons and computers. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, UIST '03, pages 149–158, New York, NY, USA, 2003.

Ken Hinckley, Gonzalo Ramos, Francois Guimbretiere, Patrick Baudisch, and Marc Smith. Stitching: pen gestures that span multiple displays. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '04, pages 23–31, New York, NY, USA, 2004.

Lars Erik Holmquist, Johan Redstrm, and Peter Ljungstr. Token-based access to digital information. pages 234–245, 1999.

Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '97, pages 234–241, New York, NY, USA, 1997.

Shahram Izadi, Harry Brignull, Tom Rodden, Yvonne Rogers, and Mia Underwood. Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, UIST '03, pages 159–168, New York, NY, USA, 2003.

Brad Johanson, Greg Hutchins, Terry Winograd, and Maureen Stone. Pointright: experience with flexible input redirection in interactive workspaces. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*, UIST '02, pages 227–234, New York, NY, USA, 2002.

Maria Karam and m.c.schraefel. A taxonomy of gestures in human computer interaction. *ACM Transactions on ComputerHuman Interactions*, pages 1–45, 2005.

Shin'ichi Konomi, Christian Müller-Tomfelde, and Norbert A. Streitz. Passage: Physical transportation of digital information in cooperative buildings. In *Proceedings of the Second International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture*, CoBuild '99, pages 45–54, London, UK, UK, 1999. Springer-Verlag.

Michael Moyle and Dr. Andy Cockburn. A flick in the right direction - an evaluation of simple gesture based controls, 2001.

Miguel Nacenta, Carl Gutwin, Dzimitri Aliakseyeu, and Sriram Subramanian. There and back again: Cross-display object movement in multi-display environments. 24(1):170–229, 2009.

Miguel A. Nacenta, Dzmitry Aliakseyeu, Sriram Subramanian, and Carl Gutwin. A comparison of techniques for multi-display reaching. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '05, pages 371–380, New York, NY, USA, 2005.

Miguel A. Nacenta, Samer Sallam, Bernard Champoux, Sriram Subramanian, and Carl Gutwin. Perspective cursor: perspective-based interaction for multi-display environments. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 289–298, New York, NY, USA, 2006.

J. Karen Parker, Regan L. Mandryk, and Kori M. Inkpen. Tractorbeam: seamless integration of local and remote pointing for tabletop displays. In *Proceedings of Graphics Interface 2005*, GI '05, pages 33–40, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.

Joseph Pato and Jason Rouault. Identity management: The drive to federation. aug 2003.

Adrian Reetz, Carl Gutwin, Tadeusz Stach, Miguel Nacenta, and Sriram Subramanian. Superflick: a natural and efficient technique for long-distance object placement on digital tables. In *Proceedings of Graphics Interface 2006*, GI '06, pages 163–170, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.

Jun Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, UIST '97, pages 31–39, New York, NY, USA, 1997.

Jun Rekimoto. Synctap: synchronous user operation for spontaneous network connection. *Personal Ubiquitous Comput.*, 8(2):126–134, May 2004.

Jun Rekimoto and Masanori Saitoh. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, CHI '99, pages 378–385, New York, NY, USA, 1999.

Jun Rekimoto, Yuji Ayatsuka, and Michimune Kohno. Synctap: An interaction technique for mobile networking. In *Mobile HCI*, volume 2795 of *Lecture Notes in Computer Science*, pages 104–115, 2003.

Larry Smarr and Charles E. Catlett. Metacomputing. *Commun. ACM*, 35(6):44–52, jun 1992.

Randall B. Smith. What you see is what i think you see. *SIGCUE Outlook*, 21(3):18–23, feb 1992.

Norbert A. Streitz, Jörg Gei, Torsten Holmer, Shin'ichi Konomi, Christian Müller-Tomfelde, Wolfgang Reischl, Petra Rexroth, Peter Seitz, and Ralf Steinmetz. i-land: an interactive landscape for creativity and innovation. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, CHI '99, pages 120–127, New York, NY, USA, 1999.

Pasi Valkkynen. *Physical Selection as Tangible User Interface*. Human Computer Interaction, Shane Pinder (Ed.), 2008.

Shumin Zhai, Barton A. Smith, and Ted Selker. Improving browsing performance: A study of four input devices for scrolling and pointing tasks. In *Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction*, INTERACT '97, pages 286–293, London, UK, UK, 1997. Chapman & Hall, Ltd.

Index

Typeset May 29, 2012