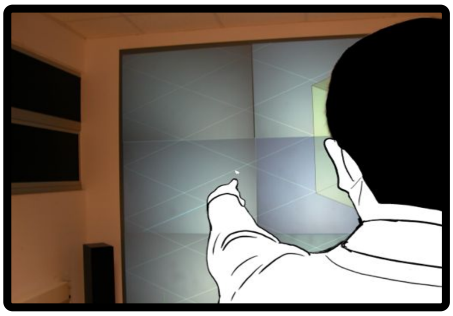
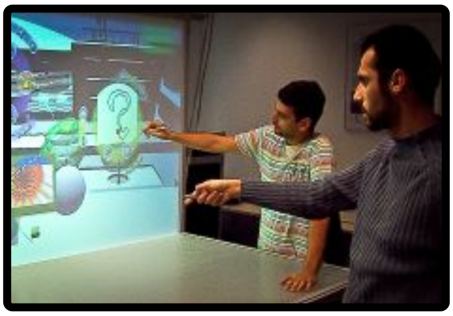
UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO





Understanding the Effect of User Position on Ray-Pointing Techniques for Large Scale Displays

Ricardo Jorge Jota Costa

Supervisior: Doctor Joaquim Armando Pires Jorge
Co-Supervisior: Doctor Saul Greenberg

Thesis approved in public session to obtain the PhD Degree in Information Systems and Computer Engineering

Juri final classification: Pass with Merit

Jury

Chairperson: Presidente do Conselho Científico do IST
Members of the Committee:

Doctor Saul Greenberg
Doctor Joaquim Armando Pires Jorge
Doctor Nuno Manuel Robalo Correia
Doctor Tobias Isenberg
Doctor Daniel Jorge Viegas Gonçalves
Doctor Carlos António Roque Martinho

2011

UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Understanding the Effect of User Position on Ray-Pointing Techniques for Large Scale Displays

Ricardo Jorge Jota Costa

Supervisior: Doctor Joaquim Armando Pires Jorge
Co-Supervisior: Doctor Saul Greenberg

Thesis approved in public session to obtain the PhD Degree in Information Systems and Computer Engineering

Juri final classification: Pass with Merit

Jury

Chairperson: Presidente do Conselho Científico do IST
Members of the Committee:
Doctor Saul Greenberg
Doctor Joaquim Armando Pires Jorge
Doctor Nuno Manuel Robalo Correia
Doctor Tobias Isenberg
Doctor Daniel Jorge Viegas Gonçalves
Doctor Carlos António Roque Martinho

This work was funded in part by the Portuguese Foundation for Science and Technology, grant **SFRH/BD/17574/2004**, The European Commission and the FP6 Specific Targeted Research Project **IMPROVE** (IST-2003-004785), and the European Commission and the FP7 Specific Targeted Research Project **MAXIMUS (**IST-2007-1-217039).

2011

Resumo

Ecrãs de larga escala permitem que os utilizadores sejam fisicamente móveis. Como tal, a utilização de teclado e rato é desaconselhada, porque a sua utilização limita a mobilidade. Para permitir que o utilizador se mova livremente, foram propostas técnicas de interacção baseadas no gesto de apontar. No entanto, existe uma lacuna de informação sobre quais os factores humanos que afectam esta classe de técnicas. Em particular, não é sabido se a posição física do utilizador tem algum efeito na utilização de técnicas de apontar. O objectivo deste trabalho é investigar que efeito a posição do utilizador tem nestas técnicas.

Os nosso resultados demonstram que a distancia do utilizador ao ecrã não parece afectar a interacção. No entanto, para as tarefas executadas, os utilizadores demonstraram uma preferência para uma distância em particular. Também descobrimos que os factores humanos tipo de controlo e parallax influenciam a interacção. Técnicas com tipo de controlo rotacional são adequadas para tarefas de visar e técnicas com parallax reduzido são adequadas para tarefas de rastear. Para concluir, estes resultados foram aplicados num cenário de ecrã de larga escala, para revisão de modelos arquitectónicos, onde foi desenvolvida interface multi-modal baseada em técnicas de apontar.

Palavras Chave

Interacção Homem Máquina

Ecrãs de larga escala

Técnincas de apontar

Factores Humanos

Parallax

Distancia do utilizador ao ecrã

Abstract

Large-scale displays support interaction scenarios where the user is physically unrestrained. However, standard input devices – such as keyboards and mice – are cumbersome and might restrain user mobility in large scale displays.

Pointing has been suggested as an interaction technique adequate for such scenarios. However, little is known regarding what human factors affect pointing performance. In particular, whether user position and distance to display are relevant to pointing is yet to be researched. Our proposal is to understand how user position and location affect pointing as an interaction technique.

Our results suggest that user distance to the display does not affect pointing, but users exhibit preferences for a specific distance according to the given task. We also found that control type and parallax have influence the way users perform. The results show that techniques based on rotational control perform better for targeting tasks, and techniques with low parallax are best for tracing tasks. To conclude, these results were applied in a real-world scenario where a prototype for architectural design review was build for a large-scale display with a multi-modal interface based on a pointing modality.

Keywords

Human Computer Interaction

Large-scale Displays

Pointing Techniques

Human Factors

Parallax

User Distance

Acknowledgements

First and foremost this thesis would not have been possible without the utmost support of my advisors, Joaquim A. Jorge and Saul Greenberg. I will be forever in debt for their guidance in the best of times and beacon of optimist in the worst of times.

Secondly, I want to share this achievement with my dear colleagues at Inesc-ID, who saw me grow and vividly shared their energy with me. Bruno Araújo, Alfredo Ferreira, Luís Bruno, Diogo Mariano and Pedro Lopes were always available and are second to none. Also, I would like to show my gratitude to the researchers at iLab, who gave the warmest welcome in Calgary. Their contributions are memorable and I would not be who I am today if it was not for them. Sheelagh Carpendale, Ehud Sharlin, Miguel Nacenta, Nicolai Marquardt, Sean Lynch, Paul Lapides and Matthew Tobiasz made working in Calgary a fantastic experience. I am glad to have turned those collaborations into long-lasting friendships.

Finally, I would like to leave a written statement of deep love for father, mother and closest friends. Their unconditional patience towards me is one of the best things in life. A man is only truly alone when he has no friends.

Thank you,
Ricardo Jorge Jota Costa

Contents

List of Figures

List of Tables

1
Introduction

Recent developments in technology are changing the concept of computer by expanding the spectrum of possible form factors. In fact, available devices range from the almost invisible [Ni & Baudisch, 2009] to the mobile device, tablets, desktops and, at the other side of the spectrum, very large displays [Buxton *et al.* , 2000]. We look at the large end of this spectrum and focus on interaction with large-scale displays. To increase resolution, large displays are often built using a number of desktop-size tiles. Depending on the number of tiles used, they can offer dozens of times more pixels than a desktop. There are two main kinds of displays namely, tabletops and display walls. Tabletops are horizontal surfaces used by multiple users in collaboration tasks while display walls are vertical displays mostly used to visualize large quantities of information by one or more users.

When interacting with large displays users often take advantage of being physically unrestrained to a position, as opposed to desktops which force them to be stationary. In tabletop scenarios, users walk around the table for collaboration purposes or to access information available on the other side of the table. When working with display walls users approach the display either to focus on particulars or to get a more detailed view. In both scenarios users often move sideways, in order to access information not in front of them. Because they are free to move, standard input devices, such as keyboards and mice, are inadequate for very large displays. Indeed, keyboards are cumbersome to carry, while conventional mice require a tracking surface that constrain movement. Because of this, new input devices have been proposed for interaction with large-scale displays.

For tabletops, touch technology is the default modality, but for vertical displays there has not been a single device that enjoys acceptance comparable to that of the touch for tabletop or the input solution in the desktop. However, pointing is likely the set of interaction techniques most commonly used with very large displays. When pointing, users designate information by either performing an appropriate gesture or by aligning an input device in an adequate manner towards the target. Pointing at large-scale displays is dependent on the input device [Myers *et al.* , 2002] and on the pointing technique adopted [Grossman & Balakrishnan, 2005; König *et al.* , 2009; Tse *et al.* , 2007]. This duality of input device and interaction technique is reflected on the previous work for pointing interfaces for large-scale displays. Some focus on input devices [Baudisch *et al.* , 2006; Jiang *et al.* , 2006; Oh & Stuerzlinger, 2002; Wilson & Cutrell, 2005] while others focus on and interaction techniques [Kobayashi & Igarashi, 2008; Nacenta *et al.* , 2006a; Tse *et al.* , 2007].

It is still not clear what affects pointing performance. For large-scale displays, the general problem is, how to interact in these scenarios? How do their characteristics affect the way we use input devices? What techniques better suit those devices to support interaction with such large surfaces? More specifically, when interacting with display walls, current research has not explored whether user position and distance to display are relevant to pointing techniques. Although there is research work that considers how distance affects interaction [Ball *et al.* , 2007; Harrison & Dey, 2008; Myers *et al.* , 2001, 2002; Raja *et al.* , 2004; Usoh *et al.* , 1999; Vogel & Balakrishnan, 2004, 2005], there is little understanding on how large-scale displays can affect the way users interact. Within the context of a single user working with very large displays, we state the problem as follows:

How does a user's position affect interaction with pointing based techniques on very large displays?

Our goal is to identify what variables affect pointing as an input technique, thus providing insights to practitioners developing pointing-based techniques for very large displays. To achieve our goal we took the following approach. (1) Build an environment where users interact using pointing techniques with a large-scale display. (2) Run user tests to get a basic understanding of pointing techniques and obtain qualitative user feedback regarding the experience. (3) Apply controlled experiments to study the relation between pointing performance and external factors such as user distance, different pointing techniques or human factors such as control type and parallax. (4) Apply these results in a real-word application to evaluate user satisfaction regarding the user experience developed.

1.1 Results

Our results are described in Chapter 3 and Chapter 4. A short description of the main results is provided next.

Distance & Technique Preference Distance does not affect interaction as much as the technique used or the task performed. When comparing three pointing metaphors: *Point*, *Grab* and *Vertical Mouse*, we found that, for tasks that require speed, *Point* was the fastest. However, for tasks that require precision, *Grab* presents better results. On the same study, we asked the users to give their preference regarding distance and effort. Users classified *Grab* as less tiresome and stated preference to interact on the 2 meter mark.

Control Type & Parallax We looked at how control type and parallax affect ray pointing techniques. Four pointing variants were tested on a large-scale display with both horizontal and vertical targeting tasks and tracing tasks. Furthermore, given the lack of distance influence, the selected user position was such that the display covered the users' center of gaze and whole peripheral view. The results show that (a) techniques based on *rotational control* perform better for targeting tasks, and (b) techniques with low parallax are best for tracing tasks. They also show that (c) a Fitts' law analysis based on angles (as opposed to linear distances) better approximates people's ray pointing performance.

User Preference for Pointing Techniques Based on the our previous findings we developed a multi-modal pointing interface to be applied on a large-scale display within the context of architectural design review. User studies conducted with architects reveal that users feel comfortable with the system and suggest that, for the tasks proposed, they prefer the multi-modal approach instead of more conventional interactions, such as menu based interaction.

1.2 The Evolution of the work presented

The work that will follow, was developed during a period of six years between 2005 and 2011. While some of the work executed was left aside for this thesis (see Chapter 5 for a quick description on those), most work executed during that period found its way into this document. For clarity purposes, the timeline execution of those work was not reproduced in the document, and the body of work was re-arranged according to the desired argument and flow of the document, instead of when (or where) it was produced.

1.2.1 Timeline

To help readers understand what was produced, but also how and when it was produced, we now present a timeline of the body of work described, including hardware build to create the scenarios required for tests, direct collaborations for each project, context in which the work was developed and related publications achieved during this period.

First Year: 2005 - 2006

The PhD program requires their applicants to execute a first year of classes, where applicants can choose from a number of possible advanced classes that fit their intended plan of work. Therefore, the first part of the year was utilized in attending those classes and satisfying this pre-requisite. At the same time, we initial vision of what would be the "Lourenço Fernandes" laboratory was being devised. At this time, we had already understood that a Large-Scale Display was a requirement for the following work. With no previous expertise on how to build - from scratch - such environment, I collaborated with Bruno AraÃžjo and Tiago Guerreiro, with guidance professor Joaquim A. Jorge. During this first year we focus on the construction of the large-scale display, including the deployment of a cluster of computer to power the display, physical frame to support the displays at the right weight to form the tile, aluminum platforms to fine-tune the displays position (thus creating the idea of a seamless display), the projection screen installation and, finally, the display hardware and auxiliary computers to create an intra-net. I was personally involved in all steps, but had direct responsibilities with the deployment of cluster and infra-structure to support the display. The end result was a large-scale display scenario, unique - at the time - at the academia in Portugal that would be used in the next six year by multiple projects and presentations. This resulted in the following publications:

Leme Wall: Desenvolvendo um sistema de Multi-projecção, Bruno Araujo, Tiago Guerreiro, Joaquim Jorge, João Pereira, **Ricardo Jota**. 13 Encontro Português de Computação Gráfica, Vila Real, Portugal, Oct 2005

Automatic Color Calibration for Commodity Multi-projection Display Walls Soares, L. P., **Costa, R. J.**, Araujo, B. R. and Jorge, J. A. 2007. In Proceedings of the IX Symposium of Virtual Reality (Petrópolis , RJ/Brazil , May, 2007)

Second Year: 2006 - 2007

For the second year of the work-plan, we focus on adding input technology to the recently created large-scale display environment. At the same time, the kickstart of the IMPROVE project (an european project with the focus on large-scale display application for design and architecture) provided us with a end-user application were we could apply our ideas. I was responsible for task analysis which directly influenced the interfaces available in "Lourenço Fernandes" laboratory. During the second year, I was involved in IMPROVE, as task analyst, was the primary software developer for the user interface and provided technical solutions to implement the pointing device within a large-scale display. The work for the first two years is available as parts of Section 2.7 and Chapter 4 and resulted in the following publications:

Collaborative Visualization of Sensor Data Through a Subscription based Architecture. M. Witzel, M. Andreolli, G. Conti, R. De Amicis, B. De Araujo, **R. Jota** e J. Jorge Eurographics Italian Chapter annual event, 02/2007

IMPROVE: Designing Effective Interaction for Virtual and Mixed Reality Environments. Pedro Santos, Andre Stork, Thomas Gierlinger, Alain Pagani, Bruno Araujo, **Ricardo Jota**, Luis Bruno, Joaquim Jorge, Joao Madeiras Pereira, Martin Witzel, Giuseppe Conti, Raffaele de Amicis, Inigo Barandarian, Celine Paloc, Oliver Machui, Jose M. Jimenez, Georg Bodammer, Don McIntyre Fraunhofer-IGD, A2, TU-Darmstadt, FB21,GRIS, INESC-ID, GraphiTech, VICOMTech., Trivi. HCI International 2007 - 12th International Conference on Human-Computer Interaction, July 2007, Beijing

IMPROVE: Collaborative Design Review in Mobile Mixed Reality. Pedro Santos, Andre Stork, Thomas Gierlinger, Alain Pagani, Bruno Araujo, **Ricardo Jota**, Luis Bruno, Joaquim Jorge, Joao Madeiras Pereira, Martin Witzel, Giuseppe Conti, Raffaele de Amicis, Inigo Barandarian, Celine Paloc, Oliver Machui, Jose M. Jimenez, Georg Bodammer, Don McIntyre Fraunhofer-IGD, A2, TU-Darmstadt, FB21,GRIS, INESC-ID, GraphiTech, VICOMTech., Trivi. HCI International 2007 - 12th International Conference on Human-Computer Interaction, July 2007, Beijing

IMPROVE: Advanced Displays and Interaction Techniques for Collaborative Design Review. Pedro Santos, Andre Stork, Thomas Gierlinger, Alain Pagani, Bruno Araujo, **Ricardo Jota**, Luis Bruno, Joaquim Jorge, Joao Madeiras Pereira, Martin Witzel, Giuseppe Conti, Raffaele de Amicis, Inigo Barandarian, Celine Paloc, Oliver Machui, Jose M. Jimenez, Georg Bodammer, Don McIntyre Fraunhofer-IGD, A2, TU-Darmstadt, FB21,GRIS, INESC-ID, GraphiTech, VICOMTech., Trivi. HCI International 2007 - 12th International Conference on Human-Computer Interaction, July 2007, Beijing

Third Year: 2007 - 2008

For the third year, PhD applicants are required to present a candidacy and to defend their work-plan in a public defense. Based on the early finding of IMPROVE, the candidacy occupied the first part of the third year, was successfully defended and resulted in Section 2. Following the candidacy defense, during the second semester of the year, we conducted the first pointing user study that can be read in the first part of Chapter 3. Finally, based on the ideas presented on the candidacy (the idea that user position affects interaction with large-scale displays) and the preliminary results of the conducted user tests, we conducted the final users tests with the IMPROVE prototype and presented the results in the conclusion of the three year european project. The conclusions on the user tests (Chapter 4) were obtained during this period and later revisited, through user videos, annotations and application logs, to fit the frame of this document (Further finding can be found in the documents published during the third year). The work executed during this period resulted in the following publications:

IMMIView: a multi-user solution for design review in real-time. **Ricardo Jota**, B. Araujo, L. Bruno, J. Pereira and J. Jorge. In Journal of Real-Time Image Processing, Volume 5, Issue 2. Springer. 91-107

A comparative study of interaction metaphors for large-scale displays. **Ricardo Jota**, M. Pereira, and J. Jorge. 2009. In Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems. ACM, New York, NY, 4135-4140.

Fourth Year: 2008 - 2009

My fourth year was spend abroad, at the university of Calgary, under the supervision of Saul Greenberg. With collaboration with Miguel Nacenta, we conducted a new user study on pointing techniques which is detailed in the second part of Chapter 3. Along with the previous user study (executed in year three) this represents the theoretical contribution of this thesis and resulted in the following publication:

A comparison of ray pointing techniques for very large displays. **Ricardo Jota**, M. Nacenta, J. Jorge, S. Carpendale, and S. Greenberg. 2010. In Proceedings of Graphics Interface 2010 (GI '10). Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 269-276.

Fifth Year: 2009 - 2010

This document was produced during the fifth year, mostly during 2010. Alongside with the writing of the thesis, two side-projects were also executed: Maximus, a european project on large-scale display for car design review applications and an internship at Microsoft Research, under the supervision of Hrvoje Benko. Both of these projects are not fully integrated within this document as they were considered not within the scope of the argument presented. However, the tabletop presented in Section 2.7 is a direct result of my involvement in Maximus; it was produced with collaboration with Diogo Mariano and integrated in the "LourenÃğo Fernandes" laboratory. Further information regarding publications or work produced in these is detailed in Chapter 5. The Microsoft internship produced an accepted publication at EACHI'10.

Constructing virtual 3D models with physical building blocks. **Ricardo Jota** and Hrvoje Benko. 2011. In Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems (CHI EA '11). ACM, New York, NY, USA, 2173-2178.

Final Year: 2010 - 2011

The thesis document was concluded during the first semester of 2011 and defended on September 12, 2011. Alongside with we conclusion of the thesis, I collaborated with Pedro Lopes towards a short paper, accepted at Interactive Tabletop Surface. This work was conducted using the hardware described in this thesis

Augmenting touch interaction through acoustic sensing. Pedro Lopes, **Ricardo Jota**, and Joaquim A. Jorge. 2011. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11). ACM, New York, NY, USA, 53-56.

1.3 Dissertation Outline

The remaining of this document is organized in four chapters. The next chapter presents the research background, organized around how humans interact on large-scale display scenarios. We start by defining theoretical frameworks that provide insights on how users perceive interactive environments. This is followed by a classification of large-scale displays. We classify them by a number of factors, including display size and orientation. Afterwards we introduce previous work on input devices, interaction techniques and distance related research. We conclude with a discussion that identifies what are the issues that we feel require further research.

In Chapter 3 we describe in detail the two studies conducted to evaluate our hypothesis of user position affecting interaction performance. In the first study, we tested three metaphors in three different distances. The second user study was conducted to understand what factors affect the interaction metaphor. In the second study we focus on ray-pointing, the class of technique that showed the best promise. In Chapter 4 we take the lessons learnt in the user studies and apply them to a real world design review application that uses ray pointing techniques to interact with a large-scale display.

Finally, in Chapter 5 we present an overall discussion of our work, delineating conclusions and introducing perspectives for future research.

2

Previous Work

Interactive surface scenarios include: (1) multiple users, free to move around; (2) one or more output devices to visualize information, generally a display is presented as the primary output device; and (3) multiple input devices to manipulate information and control the environment. This chapter describes the related work for each component and presents related work that focuses on distance-aware interaction. We start by describing theories that provide an insight on how the user behaves and how distance affects user behavior, and follow this with a classification of primary output devices, focusing on large displays, and input devices for interactive surface scenarios. Afterwards, we describe the laboratory *Lourenço Fernandes*, a large-scale environment where the presented work was partially produced. Finally, we finish the chapter with a discussion of distance-aware interaction research.

2.1 Theoretical Frameworks

Descartes suggested that the body works like a machine, that it has the material properties of extension and motion, and that it follows the laws of physics. The mind (or soul), on the other hand, was described as a nonmaterial entity that lacks extension and motion, and does not follow the laws of physics [Descartes, 1641]. In Descartes' view, the subject is essentially an immaterial mind having a body. Thus, the body acts like an object for mind to control. Descartes ended up with an extreme rationalism. Trusting only mathematical reason, he generalized his own ideal and created a world where man became a detached observer to himself. Some of the most influential 20th century attempted at breaking out of the Cartesian paradigm, in Germany by Heidegger, and in France by Merleau-Ponty.

2.1.1 Heidegger

Martin Heidegger (1889-1976) belongs to the phenomenological tradition in continental philosophy. In "being and Time" (1927) [Heidegger, 1979] he breaks with the tradition of exploring ideas without reference to our factual existence as human beings. In trying to develop a philosophy starting out with our factual human existence, he found himself trapped in the web of meaning produced by the basic assumptions of Western civilization. He found it necessary to develop a set of new concepts better suited for the task. Heidegger is relevant for us because of the concepts regarding tool usage: *breakdown*, *readiness-to-hand*, *present-at-hand*. He describes a *breakdown* as following:

> *To the person doing the hammering, the hammer as such does not exist... The hammer presents itself as a hammer only when there is some kind of breaking down or unreadiness-to-hand.* [Heidegger, 1979]

Thus, users are only conscious of their tools when the tools do not work as expected. The concepts *readiness-to-hand* and *present-at-hand* are related to this. A tool is *ready-to-hand* when it is available to the user and works as expected. In this case, the user does not perceive the tool as one, but as a part of the task at hand. If the user becomes aware of the tool, because of a *breakdown*, then the tool is *present-at-hand* and recognized as a separate object.

2.1.2 Merleau-Ponty

Taking the study of perception as his point of departure, Merleau-Ponty was led to recognize that one's body (*le corps propre*) is not only a potential object of study for science, but is also a permanent condition of experience; a constituent of the perceptual openness to the world. He underlines the fact that there is an inherence of consciousness and body, of which the analysis of perception should take account. The primacy of perception signifies a primacy of experience, so to speak, insofar as perception becomes an active and constitutive dimension. Merleau-Ponty demonstrates what he calls a "corporeality of consciousness". He argues that the body has as much intentionality as the mind and so stands in contrast with the dualist ontology of mind and body of René Descartes. In the Phenomenology of Perception [Merleau-ponty, 1976] Merleau-Ponty wrote:

> *Insofar as I have hands, feet; a body, I sustain around me intentions which are not dependent on my decisions and which affect my surroundings in a way that I do not choose.* - Merleau-Ponty [Merleau-ponty, 1976]

The question concerning corporeality connects also with Merleau-Ponty's reflections on space (*l'espace*) and the primacy of the dimension of depth (*la profondeur*) as implied in the notion of being-in-the-world (*être-dans-le-monde*) and of one's own body (*le corps propre*). For our work, Merleau's work has the following relevant aspects:

- **Perception is embodied.** We perceive the world with and through our active bodies: "The body is our general medium for having a world".
- **Tool use.** When we learn to use a tool, it becomes integrated into our body both as potential for action and as medium for perception.
- **Bodily space.** When we act in the world, our body has a dual nature. On the one hand, we can see it as an object among other objects in the "external" world. On the other hand, it exists to us as our experiencing/living body (*le corpse propre*). As a living body, we move within a space given by the structure and limitations of our own body; our bodily space.

2.1.3 Embodied Cognition

Embodied Cognition is a growing research program in cognitive science that emphasizes the formative role the environment plays in the development of cognitive processes. The general theory contends that cognitive processes develop when a tightly coupled system emerges from real-time, goal-directed interactions between organisms and their environment; the nature of these interactions influences the formation and further specifies the nature of the developing cognitive capacities. Embodied accounts of cognition have been formulated in a variety of different ways in each of the sub-fields comprising cognitive science (i.e., developmental psychology, artificial life, robotics, linguistics, and philosophy of mind), and a rich interdisciplinary research program continues to emerge. Yet, all of these different conceptions do maintain that one necessary condition for cognition is embodiment, where the basic notion of embodiment is broadly understood as the unique way an organism's sensory motor capacities enable it to successfully interact with its environmental niche. In addition, all of the different formulations of the general embodied cognition thesis share a common goal of developing cognitive explanations that capture the manner in which mind, body, and world mutually interact and influence one another to promote an organism's adaptive success. Embodied cognition offers different claims [Wilson, 2002], the most prominent being the following.

1. **Cognition is situated.** It takes place in the context of a real-word environment.

2. **Cognition is time pressured.** It must be understood in terms of how it functions under the pressures of real-time interaction with the environment.

3. **We off-load cognitive work onto the environment.** We make the environment hold or even manipulate information for us.

4. **The environment is part of the cognitive system.** The information flow between mind and world is so dense and continuous that, for scientists studying the nature of cognitive activity, the mind alone is not a meaningful unit of analysis.

5. **Cognition is for action.** The function of the mind is to guide action, and cognitive mechanisms such as perception and memory must be understood in terms of their ultimate contribution to appropriate behavior for a given situation.

6. **Off-line cognition is body based.** Even when decoupled from the environment, the activity of the mind is grounded in mechanisms that evolved for interaction with the environment.

Embodied cognition presents some claims that are already used in HCI. That cognition is situated and time pressured is already accepted on multimodal interaction, for example, where input integration has timing restrictions [Oviatt *et al.* , 1997]. For our work, we are more interested in claims 3 and 4. These claims focus on the environment as a means to enhance our cognitive workload. Generally humans off-load work onto the environment either by preloading representations acquired throughout prior learning (pre-learned models), or by making use of the environment itself in strategic ways. Moreover, claim 4 defends that the environment is part of the cognitive system, creating a larger system that includes humans plus the environment. Thus changes between the participants, such as humans moving to a new location or beginning interaction with a different set of objects, affect cognition.

2.2 Theoretical Interaction Frameworks

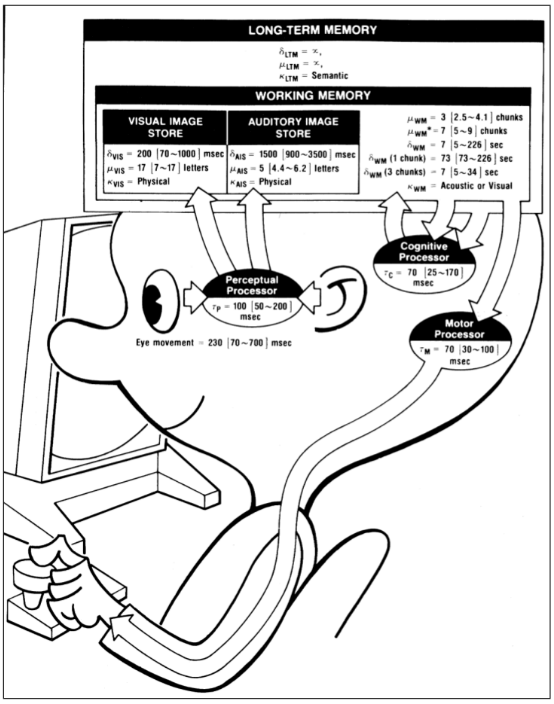


Figure 2.1: Model Human Processor, as depicted originally on Card *et al.* . 1983 [Card *et al.* , 1983].

Theoretical interaction frameworks can be seen as cognitive theories directly applied to interaction. They are too abstract to be used for practical interaction, but present concepts that allow us to understand interaction and why we interact the way we do. The main question is not "What is interaction?" but "Why do we interact like this?".

Regarding interaction frameworks, HCI research is heavily based on Cognitive Science. Indeed, Donald Norman presented a great contribution on *The Design* *of Everyday Things* and later in *The Invisible Computer* [Norman, 1999, 2002]. There are, however, other interaction frameworks that try to explain how users perceive interaction. Dag Svanæs presents a survey on *Understanding interaction* [Svanæs, 2000]. Out of the seven frameworks presented by Svanæs, the frameworks Cognitive Science, Speech Act Theory, and Activity Theory present concepts relevant to our work. Computer semiotics [Andersen, 1990] repeat concepts, already described on other frameworks, thus will not be explained here. We will also describe Model Human Processor for its significant contribution to HCI today.

The Model Human Processor (MHP) was published by Card *et al.* in 1983 [Card *et al.*, 1983]. Figure 2.1 depicts the model. The model describes the user's cognitive architecture as consisting of three processors, four memories, 19 parameters and 10 principles of operation. In their model, information reaches the perceptual processor as *chunks*. These pieces of information are stored in image store, transferred to working memory, manipulated by the cognitive processor, and some times stored in long-term memory. Some activity in the cognitive processor leads to commands being sent to the motor processor, leading to muscles being contracted and actions being performed. MHP is relevant because it can predict human reactions. Given a stimulus, a reaction can be calculated according to some processing rules (cognitive processor) and background information (memory banks).

Cognitive science is defined as the study of the mind (or intelligence). In reality, cognitive science is a broad term used for any kind of mental operation or structure that can be studied in precise terms. Cognitive science was "adapted" to HCI by Norman in *The Design of Everyday Things* [Norman, 2002]. Norman's work has two relevant points to our work: he models user intention (Figure 2.2 second step) and defends that users use multiple mental models. Furthermore, when confronted with new situations (like a new interaction device or application) users try to satisfy their intention using older mental models, constructed for similar situations. For example, when interacting with a ambient display, users may adopt the white board mental model and try to approach the ambient display and touch it. This action can be a source for incidental interaction. Moreover, Norman also presents the concept of affordance: the way objects expose their function according to their form. Finally, Norman defines automatic actions as actions that the users automate. He does not go into great detail about automatic actions except that users create them to abstract interaction.

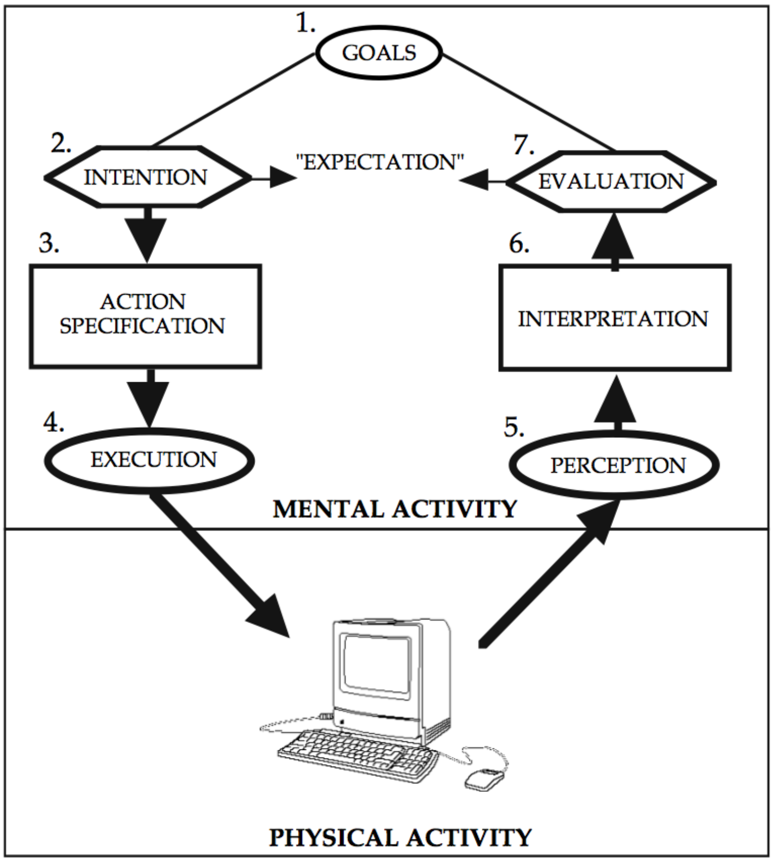


Figure 2.2: Norman's model of human-computer interaction [Norman, 2002].

Speech Act Theory was presented by Winograd and Flores in their book *Understanding Computers and Cognition, a New Foundation for Design (1986)* [Winograd & Flores, 1987]. A considerable part of their book is devoted to interpreting Heidegger. The concept of interaction using mental models is rejected. They do, however, present significant concepts: *breakdown*, *readiness-to-hand* and *present-at-hand*. Although the *hand* concepts are tool based, *breakdowns* can be interpreted as context switch or, connecting with Norman's concepts, mental models not being adequate for the new situation. For example, using a white board pen in an ambient display and then realizing that the pen cannot be erased can be considered a misusage of the white board mental modal and the user realizing this fact can be called a *breakdown*.

Finally, Activity Theory (AT) is the interaction framework that relates the most to our proposed work. It was developed by Bødker in 1990 [Bodker, 1990]. The theory breaks human work down into three levels and two aspects. The topmost level in the analysis is called activity, the second level is called actions and the third is operations. Figure 2.3 depicts the three levels. Examples of activities are: traveling from town A to town B; cleaning the house; writing a letter.

Activities are composed of actions. An action is something the subject is conscious of doing. The activity of writing a letter on a computer could include the following actions: turning the computer on, starting the word processor, typing in the letter, saving it, printing it out and turning off the computer. Actions are composed of operations, which are usually not articulated. They can be articulated in retrospect, but in the actual work situation the worker is not conscious of performing the operation. Examples of operations are: pressing a key to type in a letter, moving the mouse to place a cursor, taking a piece of paper out of the printer.

Bødker also defines a *breakdown* as when an action fails to give the anticipated result, a *breakdown* situation occurs. The operations that the action is built up from then get conceptualized and might become actions in the next try. In the same manner, an action that is done many times gets operationalized (automated) and becomes an operation.

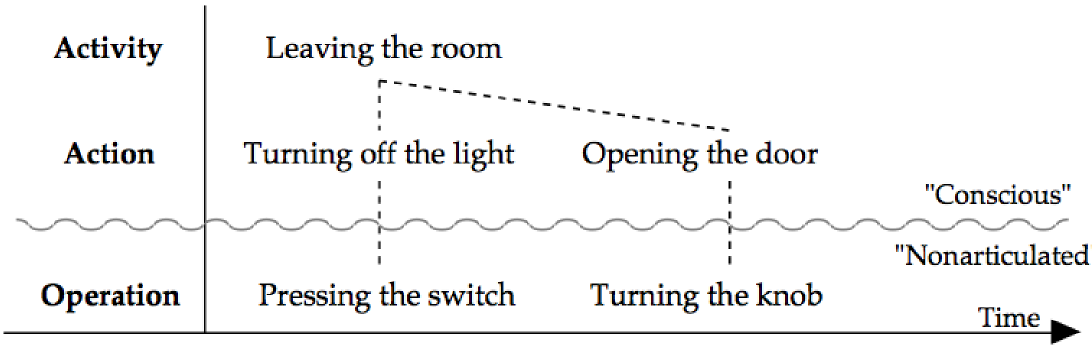


Figure 2.3: Activity Theory three levels. As seen on *Understanding Interactivity* [Svanæs, 2000].

2.2.1 Overview

These frameworks present concepts from which an interface can be built and evaluated. Context, automatic actions, and breakdowns are concepts that can be directly applied to the understanding of incidental interaction, such as changing the user distance to the display. The following themes appear in multiple frameworks:

- When we use a computer, this involves both "automatic" and "conscious" operations. As we learn to master a tool, using it becomes a second nature to us and we do not have to think about the details of how it works. We simply use it.

- Breakdown situations are particularly interesting from a research perspective because the "silent" part of the user's practice then becomes articulated and more easily available for investigation. Breakdowns are often followed by "repair" attempts, either by fixing the artifact or by learning more about how it works. After a successful "repair", the articulated operations again become "silent" and no longer have the attention of the user.

- For most theories, the non-verbal/bodily aspects of interaction are treated as more primitive processes, *i.e.* animal like. Nevertheless, they can be relevant for interaction.

- The body gives meaning to the interactions. Thus, context is also created by the body, as a subject.

2.3 Task Models and Cognitive Architectures

Task models represent the refined knowledge on HCI. A comparison study is presented by Limbourg and Vanderdonckt [Limbourg & Vanderdonckt, 2003]. They intend to abstract actions and model how humans execute the steps required for completion of a task/goal. They achieve a range of objectives [Bomsdorf & Szwillus, 1999]:

- To *inform* the designers about potential usability problems, as in hierarchical task analysis (HTA);

- To *evaluate* human performance, as in Goals, Operators, Methods, and Selection rules (GOMS);

- To *support design* by providing a detailed task model describing task hierarchy, objects used and knowledge structures, as in Task Knowledge Structures(TKS);

- To *generate* a prototype of a user interface.

Most of task models abstract user action as a *method* for executing a task. Furthermore, task models seldom acknowledge cognitive actions as relevant. Thus, task models are of no use to us, because of two main arguments: (1) they model task execution not interaction itself. Some, like, GOMS try to go further and model the lowest level of task decomposition (KLM is an example of a physical extension to model user physical actions), but they still assume a given user interface. (2) The ones that try to model cognitive aspects of the interaction are too complex to be applied to simple interfaces (see Figure 2.4). Rather, they are applied to evaluate existing user interfaces. As Limbourg and Vanderdonckt argue: task models that are primarily intended as support for evaluation and user training, like HTA and GOMS, are not suitable for supporting user interface modeling. Rather, these models require an initial design of the interface.

Cognitive architectures attempt to model the human cognitive system. They are developed as broad theories of human cognition, based on a wide selection of human experimental data. Compared to task models, they have a broader understanding regarding human factors, as they may include attention, memory, problem solving, decision making, and so on. Alternatively, cognitive architectures, when applied to HCI, are large software systems that execute programs written in the "language" of the cognitive architecture. This provides predictive and quantitative information regarding user actions. Bryne [Byrne, 2003] divides cognitive architectures into two major groups, the past systems which include Model Human Processor (MHP) and GOMS, Cognitive Complexity Theory (CCT) and collaborative activation-based production system (CAPS). The second group, which he named "contemporary architectures" include the CoLiDes, Soar, EPIC and ATC-R/PM architectures. His concept of past systems is similar to the previous definition of task models. They model how users interact, their goals and objectives. Yet they omit modeling many details of user behavior. In general, anything that falls outside the domains of procedural knowledge is not modeled.

Contemporary systems are very complex, but they can be applied to model real-time predictions. However, this is limited to very simple actions, thus not yet applicable to a physical multimodal interaction with incidental actions. Furthermore, some technical issues arise. Executive-Process/Interactive Control (EPIC), for example, is developed in *LISP*. This might present a limiting factor for technical integration with applications not implemented in LISP. Moreover, the best characteristic of models and architectures—the predicament nature of actions—is lost when modeling incidental interaction. The concept of incidental prevents us from predicting when a user is going to execute an incidental action, leaving us the option of acting upon recognizable incidental—not predicted—actions. This may go according to the EPIC concept of sub-goaling, but it is a interpretation different from the intended one. Indeed, the best contribution for our work is the description of perception and motor mechanisms, relevant for identifying incidental actions.

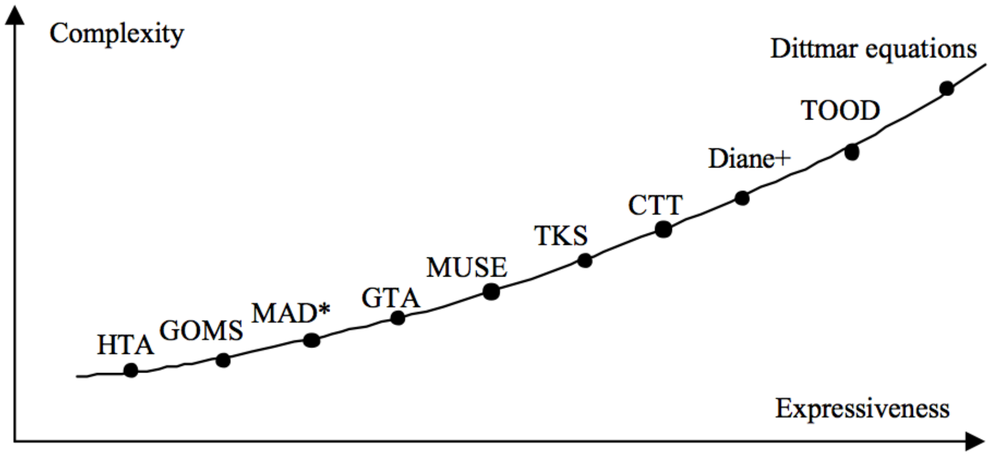


Figure 2.4: Expressiveness versus complexity of task models. (Extracted from [Limbourg & Vanderdonckt, 2003])

2.4 Proxemics

The term *Proxemics* was coined by Edward Hall [Hall, 1990]. It studies the social nature of how people use space and how differences in space (distance) can affect interaction between people. According to Hall the effects of proxemics are summarized by the following:

> *...the influence of two bodies on each other is inversely proportional not only to the square of their distance but possibly even the cube of the distance between them.* - Edward Hall

Hall argues that social distance between people is correlated with physical distance, as are intimate and personal distance, according to the following four general categories of space, each divided into two phases, *close phase* and *far phase*:

> **Intimate distance**. This area begins at the person's body and goes out to about 46cm away. It is the domain of the most intimate interactions with people, typically a small handful of people with whom a person has the closest relationships. This includes kissing, hugs, whispers and close conversation, and intimate types of touch. The *close* phase is between 0 cm and 15 cm and the *far* phase is between 15 and 46 cm.

Personal space. This area begins at about 46cm away from the person body and goes out to about 1.2 meters. It is the domain of interactions with people the person knows well, such as good personal friends. Personal space is also sometimes referenced as "personal bubble", and is the space that varies the most based on culture. The *close* phase is between 46 cm and 76 cm and the *far* phase is between 76 ncm and 1.2 meters.

Social space. This area begins at about 1.2 meters away from the person body and goes out to about 3.7 meters. It is the domain of interactions such as meeting someone new, greeting a familiar acquaintance, or generally interacting with someone who is not particularly well known to you. The *close* phase is between 1.2 meters and 2.1 meters and the *far* phase is between 2.1 meters and 3.7 meters.

Public space. This area begins about 3.7 meters away and goes out to 7.6 meters or more. It is the domain of public interactions such as taking a stroll through the mall, walking down the street, or passing other people in the grocery store. The *close* phase is between 3.7 meters to 7.6 meters and the *far* phase is above 7.6 meters.

Proxemics goes further and defines types of spaces according to the way users and object behave on that space. As such, three different kind of spaces are available:

1. **Fixed-feature space** This compromises things that are immobile, such as large displays, walls and boundaries.

2. **Semi fixed-feature space** This compromises objects that can be moved, such as mobile input devices, furniture or other movable boundaries.

3. **Informal space** The informal space compromises the space around the body, that travels along with it.

   These spaces vary from culture to culture and, although, not as relevant for interaction as the social distance, it is relevant to understand that people will react differently to objects that they can move from those that are stationary.

2.5 Large-Scale Displays

The display is such a fundamental piece of large-scale display scenarios that the environment is often named after the display technical specifications. In this section we describe what is a large-scale display and review the different types of large-scale display. We start by dividing large-scale displays into the more common vertical and horizontal surfaces. Less common solutions have also been proposed, for example the slanted table of the Perceptive Workbench [Leibe *et al.* , 2000] or the combination of vertical and horizontal presented in bendDesk [Weiss *et al.* , 2010b].

2.5.1 Vertical surfaces

Vertical surfaces are a logic extension of display technology available in desktop scenarios. Larger LCD screens are already classified as large-scale displays [Grubert *et al.* , 2007]. However, initial large scale display research was based on a single projection augmented to a display surface, significantly larger than a 56cm display [Bolt, 1980]. Large vertical surfaces range from 100cm LCD displays up to multi-projection environments, such as caves [Buxton *et al.* , 2000]. Although there are other features that can be used to define a large-scale display such as high resolution or number of computers that are dedicated to visualization, in this thesis we consider a large scale surface to be defined in relation to a user's reach and height. Therefore, any surface that is, at least, two thirds of user's height and has a width of, at least, double the users arm span is considered, by us, to be a large-scale display. Figure 2.5 depicts size comparisons. In the related work there are multiple classifications for large displays. We present a brief definition for each classification mentioned in the related work:

Interactive LCD. Some large LCDs are considered to be large-scale displays. In general, these are the smallest of the large-scale displays but may include interaction technology, such as touch-sensitive displays. They are often positioned at shoulders height where users interact by standing in front of them. Because of their size, they do not provide an immersive environment or a very high pixel count. On the other hand, they are the setup that requires the least physical space, for example a flat LCD positioned on a wall does not require physical space for a back-projection. Moreover, they are often connected to single desk-

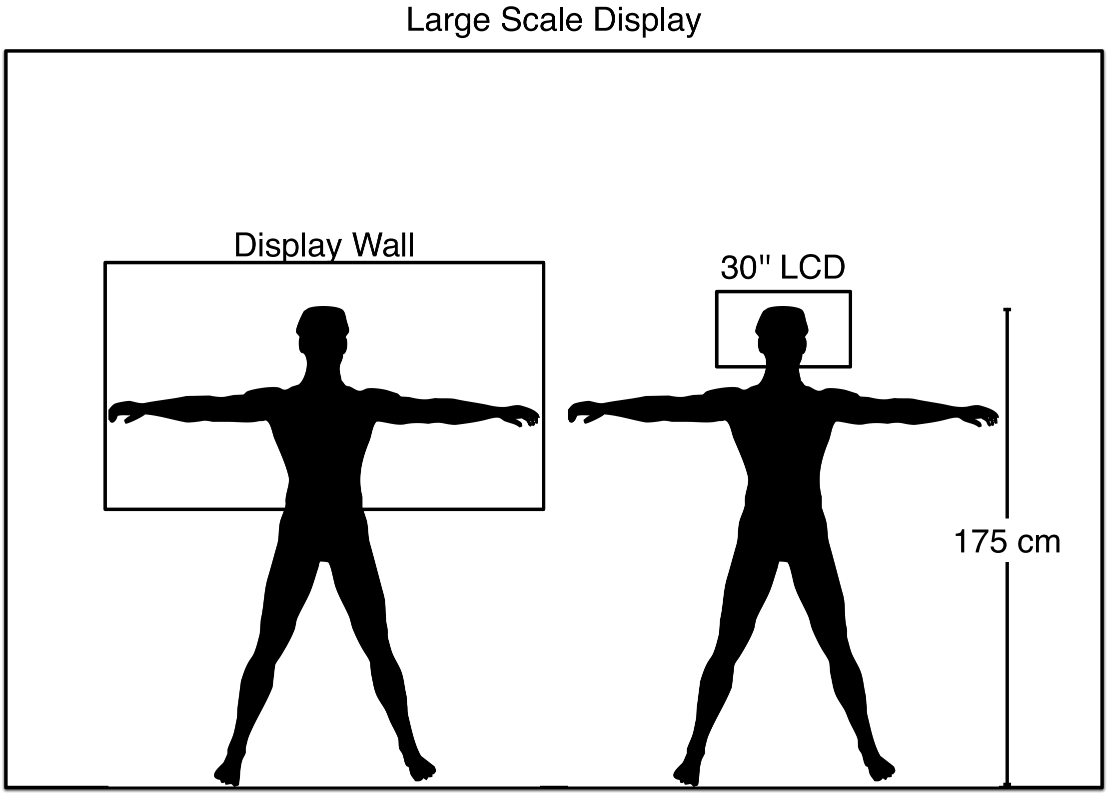


Figure 2.5: Vertical Surface size comparison.

top computer and act at a single display, thus enabling desktop applications to be used with minimal modifications.

Display Wall. Although very similar to the Interactive LCD classification, this class provides a interactive surface that is significantly larger than that of a desktop display [Weiss *et al.* , 2010a]. With current technology, a Display wall is often based on a projector. With future technologies, we foresee the use of meter-wide LCD or other technology such as OLDEs becoming available. We define a display wall as a system where users can reach the visualization sides by extending their arms (as depicted on Figure 2.5). This setup requires more physical space, if projector technology is used, as a light path is required. Interactive technology is, generally, developed independently to the visualization. Although it provides a larger interactive area, pixel count is on par with LCD technology.

Large-Scale Display. Both interactive LCDs and Display Walls provide visualization surfaces that are smaller or equivalent to the length of an human with arms stretched. We define a large-scale display as a display that is significantly larger than a user, in both height and length. Moreover, we distinguish this class from display walls by describing a large-scale displays as a display that supports three or more users simultaneously interacting side-by-side, as depicted on Figure 2.5. To achieve greater visualization sizes, large-scale displays use either high-end projectors that requires a large physical space for projection and increases costs, or multiple off-the-shelf LCD which still require some physical size for support structures but reduces costs. In large-scale displays, the multiple projectors are still powered by a single machine with multiple graphics cards. This allows developers to still use the large-scale display as a single output source and push the projection details to the operating system and the graphics drivers. Depending on the number of the displays or projectors used, pixel counts can be very large and often is two to three times bigger that display walls.

Powerwall. Large-scale displays are being applied to a number of research areas. Some areas focus on the development of new projection technologies, while others have requirements that end up focusing on specific characteristics of large-scale displays, such as high dynamic range imaging. For example, the visualization of large quantities of data requires two things: (1) an large visualization surface, with a high density of pixels and (2) high processing power. This requirement led to a class of large-scale displays based on cluster computing. Powerwalls are similar to the large-scale display in size but differ in implementation. They are powered by multiple computers, called clusters, where each computer is responsible for one output device. The result is a tiled display powered by a high-performance cluster that is able to provide high pixel count and visualize large sets of data such as cloud points or physics simulations. If presentation is not a concern, to simplify hardware implementation, standard LCDs can be used. However, visible seams will be present in the tiled display. When presentation is a concern, a solution is to replace the LCD with projectors that, when accurately positioned, do not present seams and provide the illusion of a large single display. In this case, issues such as color calibration become relevant [Soares *et al.* , 2007]. Depending on the number of output devices, Powerwalls can be connected to a desktop-like environment [DMX, 2011] but distributed visualization tools are often used [OpenSG, 2010]. These tools enable the user to abstract the cluster configuration and look at the multiple displays as a single device. However, these tools generally require the use of a specific graphical API, such as OpenGL, for visualization.

CAVE. Cave scenarios bring a new approach to large-scale displays. Other classes of large-scale display focus on presenting a single display, in front of the user. Instead of a single display, CAVE scenarios provide an immersive environment that surrounds the user with display technology. That is, CAVE displays provide visualization in front of the user, to the left, to the right, and sometimes down, up, and back. The multiple visualizations makes them more adequate for navigation scenarios or other scenarios that benefit from an immersive experience. Similar to Powerwalls, CAVEs generally require multiple computers to power the environment and require specific distributed visualization algorithms to provide a correct scene with corner correction (to increase immersion). More so than Powerwalls, caves require specific developing tools and applications developed are often developed in graphical api such as OpenSG[1].

2.5.2 Horizontal surfaces

Horizontal surfaces provide different affordances that vertical surfaces. In vertical surfaces, two users collaborate side-by-side. With horizontal surfaces users can collaborate face-to-face. This makes them more adequate for collaborative supported work. The CSCW community has long adopted tabletop as a relevant collaborative setup for multiple users. However, horizontal surfaces have not evolved on the same direction as vertical surfaces. In particular, although horizontal surfaces have been increasing in size, thus generally vary between a large LCD and a display wall size. This is due to the fact that horizontal surfaces are seen as workbench replacements for non-interactive tables and therefore follow ergonomic restrictions to how users position in collaborative environments. Indeed, bigger tables would provide more space for display and interaction but, without proper awareness techniques [Isenberg, 2009], they might not create an environment where users are aware of others and collaborate. Horizontal surfaces have developed to provide interaction and collaboration support, which we will further expand on section 2.6.1.

Horizontal surfaces are generally a single output device environment and often use projector display technology. This helps simply application development and cost. Although there is a slight variation on size, this can be solved by positioning the projector farther away from the surface or use mirrors to fold the light path. There are two main variants for projection: back-projection and front-projection. Although very similar in results, there are some differences between them.

[1] www.opensg.org

Back-projection provides an experience similar to vertical displays where objects on top of the table cannot be augmented with visualization because objects are positioned between the viewpoint and the projection origin, thus projection hits the back of the object. Back-projection often provides a smaller physical footprint, where the light path is folded and the tabletop system resembles a closed cube or rectangle. This has implications on overall size, because the position of the display forbids user interaction from that side and human comfortable table height limits projection size due to light path physics. At the time of this dissertation, LCD technology is emerging and modern LCD tables are now available[2]. For argument sake, these share the same implications of back-projection solutions, where the light path is small. Thus, they effectively remove ergonomic limitations of back-projection.

Front-projection allows for bigger sizes and does not hinder user movements because the light path can be folded above the user interaction space and pointed directly down to the surface. However, it provides a projection where any object between the projector and the table creates shadows, thus occluding the visualization. Although this might prove useful to augment objects, it can also be seen as an hinderance and is thus often avoided.

2.6 Interaction

The previous section describes the display; the definitive element in a large-scale display scenario. This section describes how such scenarios provide interactivity, thus enabling users to visualize and manipulate information.

To provide interactivity, one has to capture user actions and convert them to commands. We divide this section into two subsections: devices and techniques. Devices describe how to capture user information. Techniques describe how that information is converted to interaction commands.

2.6.1 Devices

A device is any kind of technological apparatus that is able to capture user actions and act an an input device. Input devices are the focus of multiple research communities. In this section we present input device research related

[2]version 2 of Microsoft surface. www.microsoft.com/surface

to large-scale displays. Moreover, we divide input devices according to the distance the user is to the display.

Close Interaction

A user is considered interacting at close distance if he or she is within arm's reach. This kind of interaction is common on tabletops were users touch the surface, but can also be applied to display walls.

Touch-sensitive displays. Based on capacitive or resistive technologies have been under research for at least 25 years [Lee *et al.* , 1985; Sutherland, 1963] and are now available as commodity LCD that enhance existing (non-interactive) LCD [Russell *et al.* , 2005]. One of the first interactive tabletop to have multi-touch support was the *DiamondTouch* [Dietz & Leigh, 2001]. The prototype uses modulated electric fields which are capacitively coupled through the users to receivers installed in the work environment to detect touches. Furthermore, by tracing the electric spike origin, the diamondTouch can identify which person is touching where. On the other hand, by using a modulated electric field, they still have hardware limitation in the number of touches they are able to recognize. The early units supported up to four users and later prototypes up to eight users.

To overcome hardware limitations regarding the number of detectable touches, research has also adopted computer-vision based solutions. At first, solutions like *Touchlight* had no limitations regarding the number of touches but, as a trade-off, were less precise than electrical solutions. *Touchlight* [Wilson, 2004] uses simple image processing techniques to combine the output of two video cameras placed behind a semi-transparent plane in front of the user. A touch is detected whenever the two image rectified and fused and the same information appears on the same position for both images. Another example is *PlayAnywhere* [Wilson, 2005], where Wilson *et al.* used a shadow-based touch algorithm to detect touch. Whenever the shadow fingertip size goes below a threshold, the system considers the finger to be touching a surface. In addition to touch, this prototype is also able to recognize tangibles and project on top of physical objects.

Touch detection solutions have matured and, at the time of writing this thesis, there are a number of solutions for multi-touch. *FTIR* was first introduced by

Jef Han [Han, 2005]. It works by filling a sheet of acrylic with infra-red light that bounces around inside due to total internal reflection. When a person touches the acrylic surface, the finger interrupts the total internal reflection by creating a diffusive spot and frustrating the light. This results in the infrared light being send outside the acrylic and being captured by a camera, as a bright blob. A DI setup shines infra-red light from the opposite side of the touch surface (back). The surface scatters the light and no reflection is re-projected to the camera (next to the light emitter). Whenever a user touches the surface, the finger and the acrylic create a blob that reflects back the infra-red back to the camera that can be interpreted as a touch event. DI is often used to recognize fiducially markers. Reactable [Jordà *et al.* , 2007], for example, uses this setup to recognize a number of objects and provide an interactive music experience [Fu *et al.* , 2010; Takeoka *et al.* , 2010]. *LLP* works by creating a plane of light just above the surface, on the interaction side (front). Touches refract the laser light that navigates through the acrylic to reach the infra-red camera on the other side (back) of the acrylic. The UnMousePad uses interpolating force sensitive resistance to detect touch [Rosenberg & Perlin, 2009]. Finally, Smart Technologies [Tech., 2010] provides frames and LCD that use DViT^{TM} to detect a number of touches (early system detected only two).

Other devices. are also available for close interaction. Subramanian proposes multi-layer interaction that adapts the interaction accordingly to the distance of the user's hand to the display and supports both hover and touch interaction [Subramanian *et al.* , 2006]. Stødle proposes a gesture-based technique based on vertical and horizontal gestures to control applications, again, providing very close to the display [Stødle *et al.* , 2008]. In FlowMouse, Wilson uses hand movements as a vertical mouse to control a desktop cursor [Wilson & Cutrell, 2005]. Finally, techniques such as acoustic sensing can also be applied to touch detection [Paradiso *et al.* , 2002]. While not as accurate as computer vision they can recognize a number of hand actions, such as knuckles or palm taps. With the arrival of new camera technology, such as depth cameras, new methods to enable touch surfaces are being developed. Using the Kinect, Wilson *et al.* explored how these sensors enhance non-flat surfaces and present *Lightspace* [Wilson & Benko, 2010], a prototype where normal (not instrumented) surfaces are augmented using multiple depth-cameras as input devices. Here they use the 3D information available from the depth cameras to track users' hand and create touch events whenever the user touches the surface.

Distant interaction

We define *Distant Interaction* as any interaction that is conducted outside arms reach. In contrast with close interaction, users cannot touch the display. Therefore, they require interaction metaphors that are not based on direct touch interaction. This has lead to two main solutions: gestural interfaces, based on computer vision, and pointing devices.

Gestural Interfaces Computer vision along with machine learning algorithms are often used in gestural interfaces to segment skin color and detect hand gestures. Several recognition methods have been documented. Quek [Quek & Zhao, 1996] and Wu [Wu & Huang, 2000] use inductive learning in order to reduce computation time but this required a large training set. Nolker *et al.* [Nolker & Ritter, 1996] use Hidden Markov Model to identify simple gestures. By 1998, one of the first papers describing model-based tracking for gesture recognition was presented [Lien & Huang, 1998]: yet the main problem with model-based tracking is that its computation algorithm weight did not allow for a real-time recognition.

More recently, some works [Oka *et al.* , 2002; Sato *et al.* , 2000; von Hardenberg & Bérard, 2001] focus on tracking fingertips as a gesture recognition strategy. Sato [Sato *et al.* , 2001] also presented a neural network approach, which also required a good training set. In 2003, Wu *et al.* [Wu & Balakrishnan, 2003] published a paper using hand gestural interaction, where they use a touch surface to aid gesture recognition. Access to a touch surface is not always possible, thus we do not view this as a desired setup. Rivière and Guitton [de la Rivière & Guitton, 2003, 2005] use model-based tracking to recover postures and image moments to extract translation and rotation for 3D objects. It is not clear whenever the work is rotation independent or if its recognition speed allows real-time. Kim and Fellner [Kim & Fellner, 2004] use marked fingertips and infrared light to track hand motion and recognize gestures, where they applied their work to 3D object manipulation and deformation. Malik *et al.* [Malik & Laszlo, 2004; Malik *et al.* , 2005a] use hand gestures over a tabletop as a two-handed input device for large displays from a distance. They consider fingertips and gesture recognition as two completely distinct processing steps.

Lawson and Duric [Lawson & Duric, 2006] recognize gestures by analyzing the hand silhouette and convex hull. Their recognition is both scale and rotation independent, but they can only recognize gestures that have non-convex silhouettes, thus limiting the set of identifiable gestures. Jota *et al.* use Cali [Fonseca *et al.* , 2005], a generic recognition library initially devised for recognition in calligraphic interfaces [Fonseca *et al.* , 2002], to identify specific shapes or gestures from sketches or classify hand shapes for retrieval [Jota *et al.* , 2006]. Gestural interfaces culminate with the release of depth cameras for console gaming and its use for gesture recognition within the context of gaming [Microsoft, 2010].

Pointing Devices. Pointing can be interpreted through computer vision algorithms to segment the arm and obtain an interaction point [de la Hamette *et al.* , 2002; Leubner *et al.* , 2001], or through the use of artifacts as pointing devices [Grubert *et al.* , 2007]. Vogel combines pointing with hand gestures to explore freehand pointing [Vogel & Balakrishnan, 2005] and Sato proposes the same approach but with no marker tracking [Sato *et al.* , 2001]. Kela presents a study where accelerometers are used to recognize the viability of gestures as an interaction technique [Kela *et al.* , 2006]. Kela found gestures to be good for commands with spatial association in design environment control. Kim supports 3D object manipulation using gestures and uses a grab gesture to move objects [Kim & Fellner, 2004]. Others focus on interaction based on artifacts. Jiang *et al.* use a USB camera as a pointing device [Jiang *et al.* , 2006]. The camera tracks a distinctive cursor (a bright red dot) and moves the cursor to the center of the usb camera image. That is, if the cursor is detected to the left of the image, it moves the cursor right. Effectively positioning the cursor in the general direction the user is pointing. Cao explores a colored stick as an interaction artifact for large displays [Cao & Balakrishnan, 2003] and multiple works adapt lasers for large-scale displays [Davis & Chen, 2002; Lapointe & Godin, 2005; Oh & Stuerzlinger, 2002]. Baudisch presents Soap, a pointing device based on an optical sensor device moving freely inside a hull made of fabric. Soap behaves like a mouse, but does not require a surface, thus is adequate for distant interaction [Baudisch *et al.* , 2006]. Shoemaker presents an interaction technique that makes use of a perspective projection applied to a shadow representation of a user [Shoemaker *et al.* , 2007]. He uses a light to cast shadows onto a large-scale display and, by moving towards the light, thus increasing the shadow reach, thus the user can access out-of-reach objects. Finally, Malik explores the idea of having a small tabletop with gesture recognition to interact with the display [Malik *et al.* , 2005b]. Pointing devices have also been developed in the gaming community. For example, the *Wiimote* and *PS3 Move* are two products that can be used as interaction devices in the context of gaming [Nintendo, 2011; Sony, 2011].

2.6.2 Interaction Techniques

We define an interaction *technique* as a different way to convert device input information into an interface action. Because this thesis focuses on pointing as an interaction technique, this section will focus on pointing techniques or techniques that result in cursor actions. We will further focus on ray pointing techniques, a sub-set of pointing techniques where the cursor position is defined by a ray cast from the interaction device position. The related work presented falls mostly into three categories. Laser pointing studies for distant displays, cursor enhancing techniques, and virtual reality techniques for object manipulation in 3D worlds. In this section we also discuss modeling of targeting performance and existing enhancements to ray pointing.

Laser Pointers for Large Displays. Thanks to the studies in this first category we now know a great deal about laser pointing. For example, MacKenzie and Jusoh [MacKenzie & Jusoh, 2001] and Stuerzlinger and Oh [Oh & Stuerzlinger, 2002] showed that laser pointing targeting performance is poor compared to the mouse (and around 1.4b/s vs 3.0b/s, respectively). Peck [Peck, 2001] parameterized the jitter of a laser pointer spot in terms of angle, and suggests that grip affects it. Myers and colleagues [Myers *et al.* , 2001] studied the effect of different grips and postures, and found reduced jitter with a PDA-pointer held with two hands.

Laser Pointer Enhancements. Several enhancements have been proposed that modify or improve the operation of ray pointing and distant pointing interaction. For example, laser pointers are often filtered [Davis & Chen, 2002; Vogel & Balakrishnan, 2005] or the CD gain altered [König *et al.* , 2009]. We know that some of those mechanisms may improve pointing (e.g., filtering) but these also imply trade-offs (e.g., filtering implies delay [Pavlovych & Stuerzlinger, 2009], and semantic snarfing [Myers *et al.* , 2001] makes it harder to operate with empty space). These modifications can introduce a large number of parameters that can change fundamental of ray pointing.

Cursor Enhancing Techniques. Given a pointing device and a cursor, one can enhance the mouse with more that just filtering or gain. Bolt mixes pointing with speech to create an early multi-modal interface, thus providing one of the first large-scale display cursor enhancing techniques [Bolt, 1980]. The bubble cursor is target acquisition technique based on area cursors [Grossman & Balakrishnan, 2005]. In bubble cursor, the user pointing to a desired position and the cursor automatically resizes depending on the proximity of surrounding targets, such that one target is selectable at any time. This concept is further explored in the speech-filtered bubble ray technique [Tse *et al.* , 2007], where the authors use pointing to indicate the approximate location and speech to filter targets inside the cursor bubble according to their visual characteristics, such as color. The *Ninja cursor* technique uses multiple cursors to improve target acquisition where each cursor moves synchronously following mouse movement [Kobayashi & Igarashi, 2008]. Having multiple cursors present new problems, such as which cursor is the active one. Kobayashi handles this ambiguity by only allowing a single cursor over a target, effectively disambiguating which cursor is active. However, blocking cursors may reduce the regularity of the cursor distribution and reduce ninja cursor effectiveness. Myers *et al.* use lasers to copy objects onto a handheld device and use take advantage of handheld familiar manipulation techniques to interact with the object [Myers *et al.* , 2001]. Finally, Apitz and Guimbretière present CrossY, a drawing application that demonstrates feasibility of goal crossing as the basis for a graphical user interface. Although not directly applied to large-scale displays, the crossing technique helps reduce jitter and provides a smooth interface for pointing based devices [Apitz & Guimbretière, 2005].

Pointing in VR. The variety of pointing techniques studied in the Virtual Reality literature is broader, since image-plane techniques are easy to implement (the required tracking of head or eyes is already present). Studies comparing image-plane selection to ray casting (laser pointing) for manipulation of 3D objects in 3D spaces have found that the image plane method is generally faster [Argelaguet & Andujar, 2009; Bowman & Hodges, 1997; Bowman *et al.*

, 2001; Lee *et al.* , 2003; Ware & Lowther, 1997; Wingrave *et al.* , 2005]. This led Hill and Johnson [Hill & Johnson, 2008] to propose an interaction manipulation framework based on image-plane techniques. However, most of the above-mentioned studies concern 3D tasks, which can be radically different to the 2D tasks that are our concern. It is not yet clear whether image-plane techniques will provide performance advantages for pointing to large 2D surfaces.

Modeling of the Targeting Task. Kondraske [Kondraske, 1994] suggested that rotational tasks (e.g. rotating a knob to a particular position) are better modeled by Fitts's law if the angular distances are taken into account. This modeling approach is relevant for ray pointing because the movement of the hand is often rotational, but also because, with large displays, linear measures do not represent faithfully perceived distances and sizes as seen by the user (e.g., targets of the same size on distant areas of the display appear narrower than targets close to the user). Kopper et al. [Kopper *et al.* , 2010] explored a number of models, including some based on angles, and others that include a distance parameter. To date, however, linear models for ray pointing tasks are still predominant, since there is no strong evidence supporting a substantial modeling benefit of using angles.

2.7 The *Lourenço Fernandes* environment

The Lourenço Fernandes environment is where most of the work presented in this thesis was conducted. In order to execute future steps, the first year of this work-plan was applied to building this large-scale display environment.

The environment is based on a Powerwall as a primary display and a tabletop as an horizontal display. These can be used together or as separate displays, according to the needs of the project. Users have multiple interaction devices available in the environment. For the Powerwall, users can use laser pens, track objects using a marker-based tracking system and Wiimotes. The tabletop supports multi-touch interaction and can also use tracked objects as interaction devices.

2.7.1 Powerwall

The *Powerwall* is composed by three main elements: a projection screen, a support structure to hold the projectors, and a cluster of computers to power the projectors.

Display. For projection, we use a flexible display of 4x2.25 square meters. Although acrylic solutions were available at the time, we selected using a flexible display (projection screen) as a trade-off between low cost, no visible seams, and medium contrast. One limitation of the flexible display is that physical touches make the screen move (oscillate), which makes computer vision algorithms more error-prone.

Projector Matrix and supporting frame. The display is projected using 12 DLP projectors organized on a 3 x 4 matrix. To hold the projectors in place, an aluminum structure was developed (see Figure 2.6). For calibration purposes, each projector rests on top of a mechanism that allows for precise adjustments along 6 degrees of freedom. By positioning the structure at a certain distance and adjusting each individual projector, the 12 projectors form a tiled display, with minimal overlap (around 5 pixel) between projectors.

Figure 2.6: Left: Display size compared to average human height. Center: projector and calibration mechanism. Right: Support structure used to hold the 12 projectors.

Cluster. The projectors are powered by a Linux Cluster. This solution has a lower cost than a simple SGI and allows equivalent computing power. We have one computer for each projector plus a server to control the Powerwall cluster and turn on/off the projections. They are linked with a private gigabit network to secure low latency between the server and each cluster node.

Interaction devices. There are three interaction systems available in the environment. The first solution is a pointing device based on commodity laser pens. The cluster has two infra-red cameras positioned behind the display (on the supporting frame). Those cameras are capable of detecting lasers pointed at the display. With a simple calibration step, this input device provides an accurate cursor position. The second solution is the integration of *OptiTrack* marker tracking system. This system introduces five infra-red cameras and a new cluster node, responsible for the tracking application and for providing applications with marker position. Our system is capable of tracking both single markers and marker patterns. Finally, we introduced marker enhanced Wiimotes.

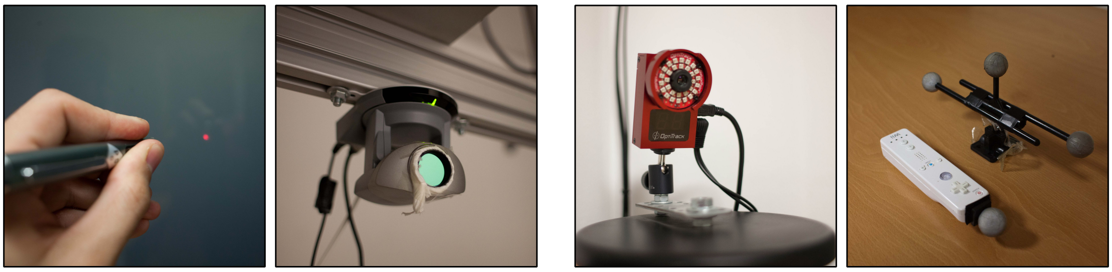


Figure 2.7: Left: Laser pointer setup (pen and camera). Right: OptiTrack tracking system and enhanced Wiimote.

2.7.2 Tabletop

As a secondary display, a tabletop is available (1.58 meters x 0.87 meters). To provide multi-touch input an LLP solution was included. To detect fingers, the LLP solution uses six lasers (780nm wavelength), distributed along the tabletop rim to casts an invisible laser light plane very close to the tabletop surface. When a finger intersects the plane, the laser light is reflected and captured by the infra-red sensitive camera (PointGrey firefly MV with 780nm IR filter). The information received by the camera is then converted to touch information. The solution presents a tabletop with enclosure projection, with three mirrors to shorten the projection light path and a laser system for interaction. Figure 2.8 shows the final prototype; the right picture shows the adopted mirror solution.

2.8 Distance Aware

Distance-aware research studies the effect user position can have on interaction. Tan shows how wider fields of vision that large-scale displays can pro-

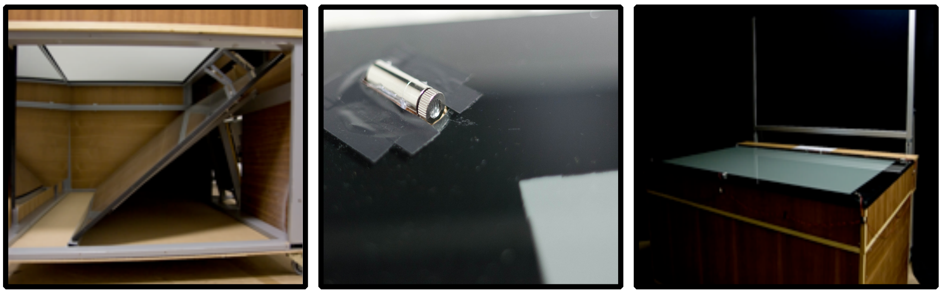


Figure 2.8: Left: Mirror solution to fold projector image path; Center: Laser devices used to create a IR light plane used for multi-touch support; Right: Final prototype.

vide can bring improvements for women on 3D navigation [Tan *et al.* , 2003]. Ball argues that increased physical navigation on larger displays improves user performance [Ball *et al.* , 2007]. Wigdor *et al.* study how the relation between display position and control space orientation affects user performance [Wigdor *et al.* , 2006]. They show that users do not always prefer the display position where they performed best and suggest that the trade-off between performance and comfort should be considered when designing multi-display environments. Teather *et al.* present a study on the effects of user position and orientation on interaction [Teather & Stuerzlinger, 2008]. They suggest that the orientation of the device, in relation to the display, is not a relevant factor. Cockburn *et al.* study the effect of visual feedback has no spatial targeting acquisition [Cockburn *et al.* , 2011] and present three different pointing techniques. They conclude that a technique that translates the pointer from a 2D plane to a screen positions presents the best results out of the three tested.

Another approach relevant for our study is how distance can be used to adapt the interface. Harrison magnifies information according to the user proximity to the screen [Harrison & Dey, 2008]. Ball applies the same technique to large-scale displays [Ball *et al.* , 2007] and acknowledges physical navigation as beneficial on the tasks tested. Vogel proposes an interactive system for public interfaces based on distance driven context [Vogel & Balakrishnan, 2004]. Shoemaker presents an interaction metaphor based on shadows [Shoemaker *et al.* , 2007]. With his shadow technique, users are required to move away from the display to reach higher targets. Shoemaker *et al.* study how distance affects interaction on another paper, where he tests text input techniques for large-scale Displays [Shoemaker *et al.* , 2009]. In this study, Shoemaker defines text input techniques accordingly to two factors, *visibility* and *distance*, and found that distance-independent techniques provide better results on large-scale displays.

Grubert and Myers present studies similar to ours [Grubert *et al.* , 2007; Myers *et al.* , 2002]. Grubert compares direct versus indirect input on a wall display. Although multiple techniques are studied (direct versus indirect), the study does not focus on understanding how distance affects each technique and does not provide an insight on how different techniques are affected by distance. Myers *et al.* tests the laser pointer metaphor on three different distances and concludes that hand jitter affects precision. However, they only study a single technique (laser pointer) and focuses on reducing hand jitter. A second experiment adds a mouse and a handheld device as interaction metaphors but does not test these on different distances. Finally, Greenberg *et al.* presents a framework for proxemics in ubicomp [Greenberg *et al.* , 2011]. In this work, they propose distance and orientation as dimensions and present a number of applications where user distance to the display affects interaction. Although they approach user position as a factor, they do not focus on interaction metaphors, in particular ray-pointing, in their framework.

2.9 Discussion

Related work shows us that the way humans interact with the world is the subject of some debate. Starting with philosophical arguments such as those by Merleau-Ponty and Heidegger and moving on to interaction specific frameworks, such as Norman's model or MHP, a lot of ground work has been done to try to understand how humans interact.

Interaction can be seen as branch of design. In fact, one main difference between interaction design and equipment design is that the premise that *"function follows form"* is weaker for interaction design[3]. Interaction design can then be described as a discipline that attempts to convey one or more functions to *objects* that are not recognizable by form. A chair's function, for example, is easily recognizable by its form, however, traditional input devices are not. A bad example in form is that of apple's original circular puck mouse as a bad example of form because its round shape does not help users understand how to hold the mouse (affordance). Changes in form can represent changes in functionality, for example, flip phones [Wikipedia, 2011] where closing the shell turns off the display, and opening the shell activates the screen and makes the phone ready for use.

[3]Argument discussed in the documentary film Objectified

In fact, recent works [Cohé *et al.* , 2011; Morris *et al.* , 2010] adopt the approach of showing the function (the initial state and the end result) and asking users to explain the form (the gesture used to obtain the end result). Under this assumption, if we focus on large-scale displays, Vogel [Vogel & Balakrishnan, 2004] identify user position as form and suggest how the display function should change according to the form. This was further explored by Harrison on the desktop [Harrison & Dey, 2008] and by Ball for navigation tasks [Ball *et al.* , 2007]. What they identify is that making interactive environments follow social theories of space, such as proxemics, makes them more interesting. However, their work only focus on the information itself.

The possibility of issuing commands introduces input devices that, in turn, changes what is perceived as form. Indeed, some research works have looked at how input devices should be used for large-scale displays [Grubert *et al.* , 2007; Myers *et al.* , 2002], but they do not describe user position as a factor for interaction, focusing instead of the actual device and interaction technique.

We identify the intersection of user position and interaction technique to be an important research subject for large scale displays that has yet to be fully explored in the related work. In the next section we present user studies that look at this problem. First, we present a study about the effect that user position, at a particular distance, has on interaction techniques. We follow this study with another study that looks at what factors actually affect a specific interaction technique called ray-pointing, which is widely used in the related work and considered to be a strong solution for interaction with large-scale displays.

3
Pointing for Large-Scale Displays

Pointing is an interesting approach to interaction for large-scale displays. However, very little is known on how pointing works on large-scale displays. This section describes the approach taken towards a better understanding of how pointing is affected by the large-scale display environment characteristics.

The approach is composed by two tests, one for external factors and another one for internal factors. External factors study how the relation between the user and the display influences pointing, in particular distance and different pointing techniques. Based on the results from the first test, the second test looks on how human factors such as control type (the way we interpret pointing) and at parallax (the way we perceive targets) affect pointing. This chapter is divided in three sections: the first two describe each test and corresponding discussion followed by an overall discussion on the findings presented.

3.1 Distance in Pointing Techniques

The way we perceive information is intrinsically related to distance. For example, in advertising billboards font size is selected according to the preview viewing distance. Some actions require users to be physically distant from the object at hand. Painters, for example, might move away from the canvas to get a clear view of the global composition. Moreover, they use different techniques and tools, which might require different interaction distance, for different sections of the painting. Therefore, distance is known to influence how we perceive the world but also what tools we select. This, however, is not applied to interaction. Most interaction techniques assume that the user will interact at one single, pre-defined distance.

As we stand today, systems either provide interactivity at one distance and ignore the rest of the distance spectrum or provide the same experience regardless of the user's distance to the display. This particularly affects large-scale scenarios where users are free to move around, in contrast to desktop scenarios, where the user is usually stationary. Few research studies take user distance into account when developing suitable metaphors for large-scale display interaction. Often, the solution adopted allows user movement, but does not adapt interaction accordingly to user position and task.

We study how distance affects interaction metaphor design for large-scale displays. We selected and implemented *Grab* [Kim & Fellner, 2004; Malik *et al.* , 2005b; Shoemaker *et al.* , 2007], *Point* [Grubert *et al.* , 2007; Leubner *et al.* , 2001; Myers *et al.* , 2002; Stødle *et al.* , 2008; Vogel & Balakrishnan, 2005] and *Mouse* [Cao & Balakrishnan, 2003; Wilson & Cutrell, 2005]; three metaphors, proposed in the previous work. We then tested a simple puzzle task with users interacting from three distances (1.5 meters, 2 meters, and 5 meters). We conducted tests with 32 users, divided into two sessions. Out of those 32 users, 9 were asked to execute both sessions, bringing the total of experiences to 41. In the first session we focused on how distance can affect how fast a user finishes a task, while on the second we focuses on how precision is affected by distance. Conducted tests show that user distance does not affect interaction. Moreover, we show that, for speed tasks, the *Point* metaphor yields the best results while, for precision tasks, the *Grab* metaphor is the most precise. Results also show that users state preference for the 2 meter interaction distance and classify *Grab* as less tiresome and easy to understand.

3.1.1 Pointing Variants

An overview of the related work indicates that most interaction techniques for large-scale displays are inspired by familiar actions. Metaphors such as grab a virtual object or point to an area of interest are mentioned in the related work and mouse-like devices are already used to interact with large-scale displays. We implement three metaphors, featured in the related work: *Grab*, *Point* and *Mouse* (Figure 3.1). All techniques define a pointing line using different user information. The intersection of the line with the display position results in the cursor position.

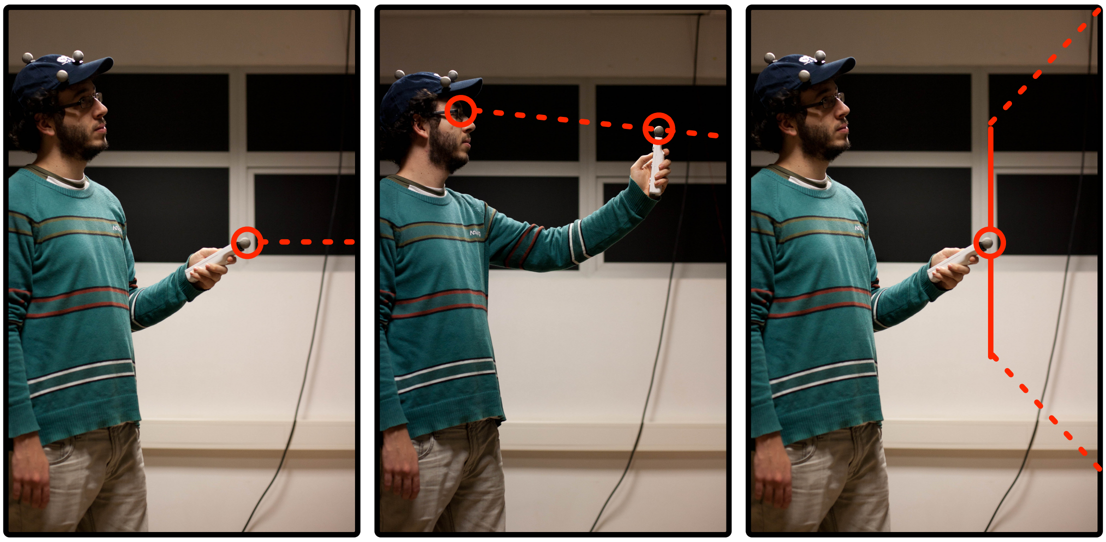


Figure 3.1: (From left to right: *Grab, Point, and Mouse.*). *Grab:* the cursor position is defined by the intersection of the display with a line controlled by the user hand (perpendicular to the display). *Point:* the cursor position is defined by the interaction of the display with a line defined by the user eye and hand position. *Mouse:* the cursor position is defined by mapping the user's hand position to a display position.

The *Grab* metaphor defines a straight line, perpendicular to the display, from the user hand to the display. Therefore, the cursor is always positioned in the display position closest to the user hand. This resembles how users physically grab objects. Imagine a user reaching for a book on a tall shelf, they are required to be in front of the shelf and reach the object to grab it. To move an object, a user has reach and grab it, to drop the user has to physically walk to the target position and release in the desired position. Because of its tactile nature, we expect that this technique is more adequate for near interactions to the display. To provide support for different distances, grab does not require the user to be close to the display, regardless of the user distance, the cursor will always be on the display point closest to the hand position.

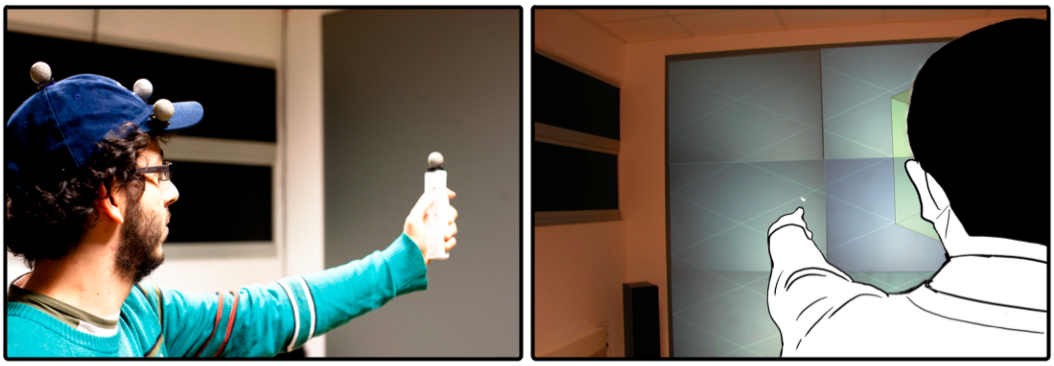


Figure 3.2: Left: Third person perspective of the pointing metaphor. Right: Over the shoulder perspective of the pointing metaphor. Notice that cursor is positioned behind the hand position.

The *Point* metaphor is based on the pointing gesture that humans execute when they want to refer to an object or location. It calculates a line between the user's dominant eye and the user's hand. The intersection of that line with the display defines the cursor position. From the user's perspective, the cursor is always behind the hand position. Therefore, the user can move the cursor over an object by occluding the object with their hand (see Figure 3.2). Contrary to the *Grab* metaphor, with *Point* users are not required to physically move or be in front of objects to select them. We expect this metaphor to be suited for medium distance. In short distance, because the objects are closer to the laser, the user has to execute a wider angle for the cursor to move to an object not in front of the user, reducing the effectiveness of this metaphor in close distances. Similarly, hand jitter can be an issue when interacting in distances away from the display because small hand movements are magnified because of the distance [Myers *et al.* , 2002].

The *Mouse* metaphor works by mapping the display space to a square 50x50 cm^2 region parallel to the display, positioned 30cm in front of the user and centered around the user torso height. This size and position were selected to give a conformable space without requiring the user to extend his arm in an uncomfortable way in order to move the cursor to the limits of the display. The region was programmed to follow the user, that is if the user position changes the region is recalculated to reflect the user's new position. When users move their hand inside this region, the cursor is moved to the correspondent position on the screen. That is, if the user moves to the top-left position of the region, the cursor moves to the top-left of the display. Because the region follows the user and there is a direct mapping of the region to the display, distance does not affect this metaphor. Thus, the *Mouse* metaphor should be more suitable for large distances.

3.1.2 Evaluation

We conducted two user studies to evaluate our hypotheses. The first focused on how fast a user could execute a task using the different metaphors on different distances. The second study looks at how distance affects a task that requires precision targeting. Our main goal was to understand how distance affects interaction. Thus we propose the following questions:

- Does distance affect interaction, regardless of the metaphor used?
- Does a metaphor have an optimal interaction distance?
- Are metaphor significantly affected by different task?

Apparatus

The user study was conducted on large-scale display (4m x 3m) composed of 12 projectors in a 4 x 3 tiles setup. Each projector had a resolution of 1024x768px (for a total of 12288x9216px). The projectors reflected on a single projection surface, with no seams. The projectors overlapped by an average of 10px and there was color different between adjacent projectors. The display rested 30cm above ground so the subjects head lined up approximately with the center of the projection (see Figure 3.3). We asked the users to stand on a location, centered in relation to the display.

We implemented the interaction metaphors using a Wiimote and a cap tracked by a marker tracking system. Tracking was available on an area ranging between 1.5 and 5 meters away from the screen. The cap was tracked using a marker artifact, and provided head position and orientation. For hand position, we used a single marker to provide the hand position and the Wiimote's *A* button to select. Whenever a user pressed the button A, feedback was provided in the form of a cursor change from a dotted circle (seen in Figure 3.5) to a full circle. We decided to use the single marker, instead of the Wiimote camera, to provide the users with a more robust tracking to handle the case where the camera would point away from the Wiimote bar (required for tracking). This was especially relevant on the *Grab* and *Mouse* interaction metaphors were pointing is not defined as a direction of the device but as function of device and head position.

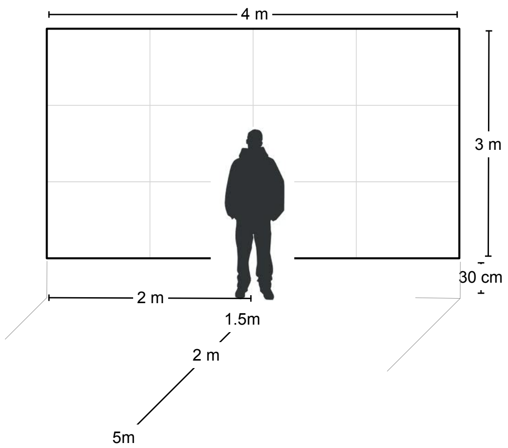


Figure 3.3: Left: Location of the user in relation to the display, distances tested. Right: Cap and Wiimote used to implement the metaphors.

Tasks

The tests were divided into two tasks: speed and precision. The goal of the first one was to conclude the task as fast as possible, the second one was to finish the task as accurately as possible. Both tests share the same task, which was to move a number of shapes into their final position. The speed test included twelve shapes, scattered on the display and the precision test included three shapes. We decided for for this test, instead of the more common ISO 9241-9 recommendation, because we wanted obtain data on how distance affects precision, and not just targeting or tracing actions. With this puzzle we test target acquisition, and also how precise (in pixels) users can position an object. Users executed the puzzle task from three distances: 1.5 meters, 2 meters, and 5 meters (calling these distances Near, Middle, and Far).

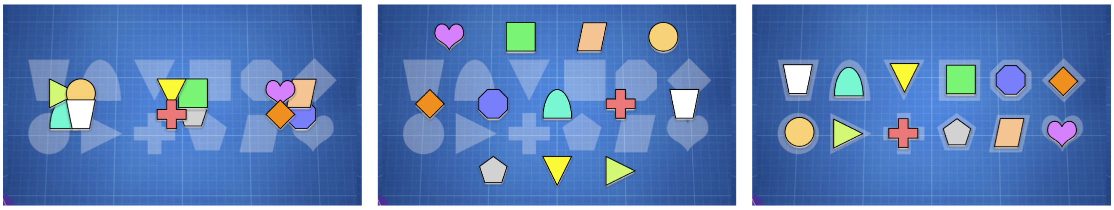


Figure 3.4: Speed session. From left to right: cluster start, spread start, final position

Speed Test. The speed session presented the users with 12 shapes. The goal was to position the shapes on top of the equivalent gray outline as fast as possible. The outlines were slightly bigger that the actual shape, and users were instructed to position the shapes completely inside the outlines. Compared to the other metaphors, the *Grab* metaphor required significant movement to move a shape across the display, therefore, to properly evaluate the metaphors, we had two puzzle starting setups (see Figure 3.4). The first positioned the shapes close to their ultimate target (cluster sub-task), the second positioned the shapes far away from their ultimate target (spread sub-task). Depending on the starting position, to solve the puzzle users had to execute short paths or long paths (respectively cluster sub-task and spread sub-task).

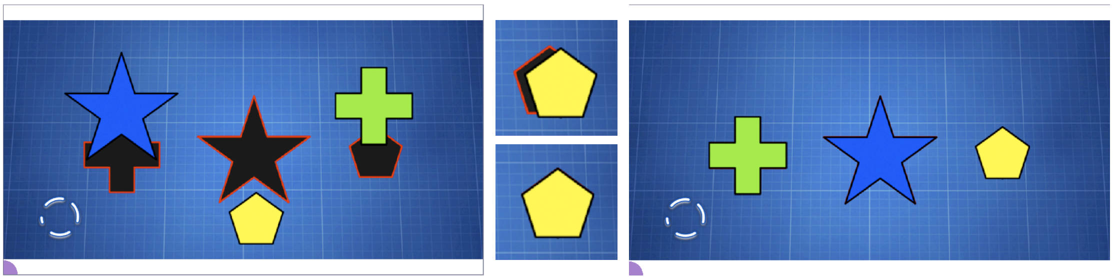


Figure 3.5: Precision session. From left to right: start position, examples of correct and incorrect shape positions, final position

Precision Test. In the precision test, users had to position three shapes. The goal was to cover the correspondent black shapes including the red border. The test was implemented so that both color shapes and black correspondents have the same size, therefore the color shapes would be able to hide the red border. Users were instructed that, if the final position was missed, the red border could still be seen (see Figure 3.5). For this session we asked the users to conduct two tests: to position the three shapes as accurate as possible without time limit (precision sub-task) and to position the 3 shapes within a time limit of 10 seconds (timed sub-task). The ten seconds limitation was observed during preliminary tests, to be enough to successfully conclude the task. With this test we expect to discover how precise each metaphor can be and understand the relation between precision and the time to successfully complete the task.

Participants

We recruited 32 users, 5 females and 27 males, nine of which executed both tests. Twenty-one tests were conducted for the speed task and twenty for the precision tasks. User age was between 22 and 56 years old (*M*=29.85, *Mdn*=28.00, *SD*=7.93), and height between 1.51 and 1.86 meters (*M*=1.73, *Mdn*=1.75, *SD*=0.09). Prior experience with large-scale display interaction ranged from beginners to experts. All subjects were right-handed.

Procedure and Design

After signing a consent form, subjects were introduced to the metaphors and explained the task. In the case of the speed test, users were asked to solve the problem as fast as possible. In the precision test, users were asked to position the shapes as accurately as possible. To prevent biasing the test outcome, input technique order was counterbalanced across subjects using a random Latin square design. Briefing lead to an open session where users got acquainted with the metaphors and task. Once users stated they were comfortable with the system, the tests would begin. During the tests, users would execute the test without our interference. After users concluded the test, they were presented with a questionnaire and debriefed regarding their experience. To conclude, users were rewarded for their time and the subject study closed.

Data Collection

During the tests we logged head and hand positions with a 60HZ frequency, the completion time in milliseconds, the distance in pixels to the optimal solution (for the precision session), the physical distance to the display (1.5, 2, and 5 meters), the sub-tasks (cluster, spread, timed or precision), and the metaphor used (*Grab*, *Point*, *Mouse*).

3.1.3 Results

Our initial belief was that distance would affect interaction. Moreover, each metaphor would present different behaviors on the three distances and we could infer the best metaphor based on the distance alone. Specifically, we thought that *Grab* was suited for near distance and precision tasks. *Point*, would fare best on middle distance and during the speed task. Finally, *Mouse* would not be affected by distance.

		Cluster sub-task			Spread sub-task		
		Grab	*Point*	*Mouse*	*Grab*	*Point*	*Mouse*
Near	μ	43.83	**30.53**	34.00	57.08	**36.00**	41.92
	σ	16.48	5.11	7.42	13.30	8.15	9.95
Middle	μ	36.02	**28.88**	32.00	57.00	**37.84**	38.50
	σ	8.50	6.66	7.42	13.35	7.55	9.94
Far	μ	42.44	**31.02**	34.62	56.43	**38.00**	40.00
	σ	9.72	6.32	10.01	10.30	12.08	8.31

Table 3.1: Speed Test - Results in seconds

Distance x *Task*	*F Score*	*p-value*	*Metaphor* x *Task*	*F Score*	*p-value*
Near x Spread	22.48	< 0.001	Grab x Spread	0.01	0.98
Near x Cluster	8.69	< 0.001	**Grab** x **Cluster**	2.49	0.09
Middle x Spread	4.77	< 0.001	Point x Spread	1.24	0.29
Middle x *Cluster*	22.27	< 0.05	Point x Cluster	0.72	0.49
Far x Spread	22.84	< 0.001	Mouse x Spread	0.69	0.50
Far x Cluster	9.14	< 0.001	Mouse x Cluster	0.71	0.49

Table 3.2: Anova results for the speed test. Left: Compare result metaphors by distance & task. Right: Compare result distances by metaphor & task.

Speed

Table 3.1 shows the times obtained for both Cluster and Spread sub-tasks and Table 3.2 shows the ANOVA results for the speed test (H0: $\mu_1 = \mu_2 = \mu_3$). We used the completion time (in seconds) to evaluate the speed task and tested each metaphor across the three distances. Contrary to our expectations, our results exhibits no evidence that completion times are affected by distance (right side of Table 3.1), that is: for the same metaphors we did not obtain significant time differences when varying the distance. However, there is clear evidence that the *Point* is the fastest on both cluster and spread sub-tasks. In addition, with the exception of the spread puzzle on the far distance, the *Point* presented smaller standard deviations (less variation between subjects). The faster times and reduced deviations means that users were more efficient using pointing and that less users had difficulty. One reason for this result is that users seemed to understand the *Point* metaphor faster or were already familiar with it. Moreover, the questionnaires also support this result (Figure 3.8), where 10 out of 21 users selected *Point* as their metaphor of choice. We also point out that, on average, *Grab* took more than ten seconds on the spread sub-task that on the cluster

		Timed sub-task			Precision sub-task		
		Grab	*Point*	*Mouse*	*Grab*	*Point*	*Mouse*
Near	μ	19.73	5.92	8.17	3.41	4.01	3.84
	σ	28.77	2.62	13.28	0.32	0.54	0.46
Middle	μ	36.31	14.42	9.25	3.39	3.90	3.97
	σ	58.15	26.75	17.71	0.38	0.63	0.97
Far	μ	25.74	19.69	14.04	3.30	4.06	4.03
	σ	31.80	34.18	25.41	0.38	0.75	0.92

Table 3.3: Precision test results in pixels

sub-task (Figure 3.6). We explain this with the fact that the *Grab* metaphor requires users to physically move in order to position the objects in their ultimate target, thus significantly increasing the completion time on the spread sub-task.

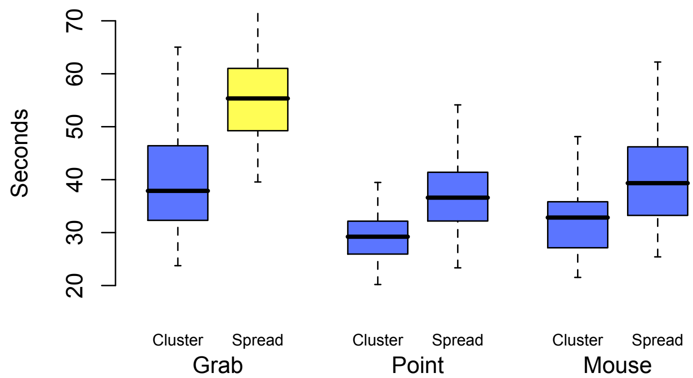


Figure 3.6: Median results for the Cluster and Spread Tasks.

Precision

We evaluated the precision results by averaging the difference between the optimal shape final position and the user's solution (Equations 3.1.1 and 3.1.2). We collected results from two tasks: Timed (users had ten seconds to finish) and Precision (no time limits). Table 3.3 shows the results in pixels, according to the metric in 3.1.1. Our results were validated using ANOVAs (H0: $\mu_1 = \mu_2 = \mu_3$) and are presented in Table 3.5.

		Grab	Point	Mouse
Near	μ	27.94	29.55	32.25
	σ	15.49	18.53	17.38
Middle	μ	23.36	30.74	25.81
	σ	10.10	19.13	12.83
Far	μ	27.94	30.84	28.02
	σ	14.09	15.54	12.28

Table 3.4: Precision test times in seconds for the precision task

$$D_{avg} = \frac{1}{3}\sum_{i=1}^{3} dist(center_i, center_{opt}) \tag{3.1.1}$$

$$dist(a,b) = (x_a - x_b)^2 + (y_a - y_b)^2 \tag{3.1.2}$$

As seen on the speed task, the precision test results shows no correlation between distance and pixel precision. None of our ANOVA results supports the hypothesis that the metaphor is affected by distance. However, when we compare the three metaphors, for distance & task, we see that the further away the users are, the less confidence the ANOVAs results show (results highlighted in bold in Table 3.5). Cross-referencing this result with the data in Table 3.3 (right side) we can see a slight decrease in precision and larger standard deviation in the middle and the far distances. Although we cannot support this with statistics, we believe that the larger standard deviation might eventually suggest that for distances larger that those tested distance might be relevant. However, to claim this further tests are required.

In the speed scenario, *Point* was the fastest metaphor. In the timed sub-task, the fastest results were divided between *Point* and *Mouse*. *Point* achieved better results (in pixels) on the near distance and *Mouse* is best on both middle and far distance. However, when given no time limits, the *Grab* metaphor present the best precision results, with both *Point* and *Mouse* achieving slower times. The users competed the precision puzzle in average 29 seconds, regardless of the metaphor (Table 3.4).

Distance x *Task*	*F Score*	*p-value*	*Metaphor* x *Task*	*F Score*	*p-value*
Near x Timed	3.26	$<$ **0.05**	Grab x Timed	0.80	0.45
Near x Precision	9.10	$<$ 0.001	Grab x Precision	0.52	0.59
Middle x Timed	2.80	$<$ **0.1**	Point x Timed	1.53	0.22
Middle x Precision	4.03	$<$ 0.05	Point x Precision	0.29	0.74
Far x Timed	0.72	$=$ **0.48**	Mouse x Timed	0.51	0.59
Far x Precision	7.06	$<$ 0.01	Mouse x Precision	0.29	0.74

Table 3.5: Anova results for the precision test. Left: Compares result metaphor by distance & task. Right: Compare results distance by metaphor & task.

Questionnaire Results

We used a six-point scale for the questionnaire questions. We asked the users to qualify their preference for each metaphor and distance (Figure 3.8), and to evaluate the experience accordingly to fatigue, learning, and usage (Figure 3.7). Eighteen out of forty-one selected *Grab* as their favorite. However, out of the nine that executed both tests, eight selected different metaphors for each test. The only user that did not change metaphor preferred grab on both tests. As we planned to ask some users to execute the two tests, not to bias the second tests distance preference with the results chosen for the first test, we only asked the second group of users for distance preference. Twelve out of twenty users stated that middle is the best interaction distance, followed by the near distance with seven users. Only one user prefers the far distance.

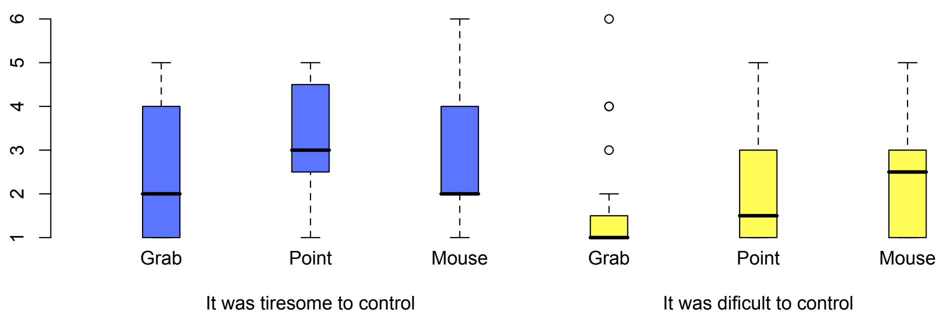


Figure 3.7: Users answers for difficulty in learning and usage. Lower means better results.

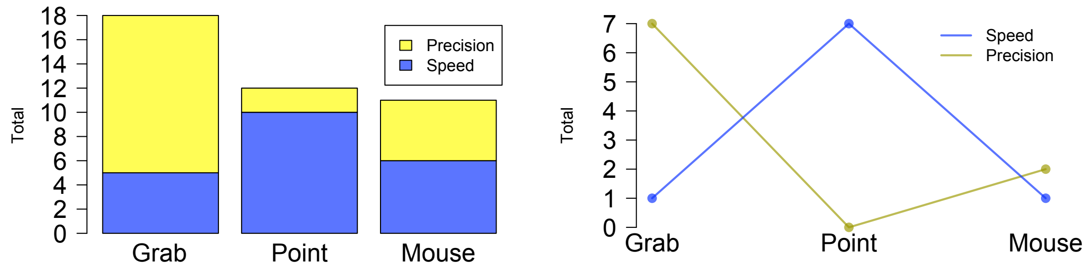


Figure 3.8: User questionnaire answers for metaphor preference. Left: User overall metaphor preference. Right: Comparison between preferences in users that conducted both tests

3.1.4 Discussion

Our main focus on this study was to understand how distance affected interaction. Both speed and precision results suggest that distances between 1.5 - 5 meters does not significantly affect interaction. Moreover, results also show that, for the tasks tested, the metaphors are not significantly affected by distances between 1.5 and 5 meters. The only result that supports our initial claim is that the timed sub-task ANOVA results show less confidence when the distance is increasing (shown on table 3.5, in bold). This might suggest that distance makes some users slightly slower at greater distances. However, without further results we do not believe this to be proof enough to claim that distance affects interaction. A corollary to this conclusion is that the metaphors do not have a optimal interaction distance. Metaphors like *Point* might still have a distance in which, due to hand jitter, interaction becomes prohibitive. However, given the user preferences for middle and near distances on the tasks tested, we do not predict this to be an issue with large-scale displays.

There is proof that the metaphor results are affected by the task, Figure 3.6 shows the differences in results for the speed test and users have different preferences for precision and speed tasks (see Figure 3.8). This is consistent with Grubert [Grubert *et al.* , 2007]. His results show that participants chose to switch between techniques to achieve different levels of precision and control for different tasks. Our questionnaire answers seem to corroborate this result as they show that the users prefer the point metaphor for speed tasks and the *Grab* metaphor for precision tasks (Figure 3.8). Moreover, different metaphor show statistical differences in completion time. This can be explained the physical

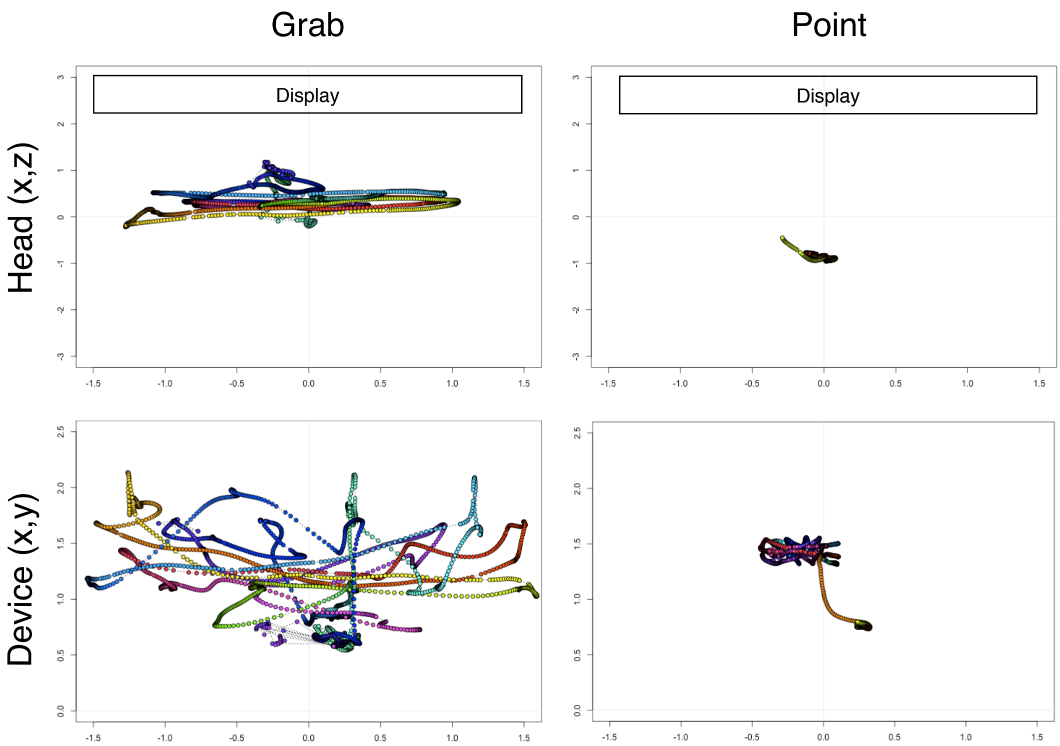


Figure 3.9: Example of user movement during a test (Speed tests; spread task). Left: head and device movement when using a grab metaphor. Right: head and device movement when using a point metaphor.The ground is (x,z) and height is y: Head is plotted from above (x,z), Device is plotted from perpendicular to the display (x,z).

movement that each metaphor requires to execute the same movement. When comparing the *Grab* and *Point* metaphors (see Figure 3.9), one can see that the *Grab* requires more physical movement to hit objects out-of-reach (by moving the device to reach or walking towards the object). This has repercussions on the completion times. *Point*, on the other hand, requires no head movement (no walking) and almost no cursor movement and is, therefore, faster.

also see an increase in deviation by distance (Figure 3.3). However, this is not enough to significantly affect precision and can be overcome with interaction techniques that either minimize jitter [Davis & Chen, 2002; Lapointe & Godin, 2005] or improve target selection [Tse *et al.* , 2007]. When we limited the test to ten seconds, the *Grab* yields worst results that both the *Point* and the *Mouse*. This is consistent with the *Grab* metaphor that requires more physical actions from the user to achieve the same result. With no time limitations, *Grab* presents the best results (in pixels) of the three metaphors. The slower movements allow for more precise movements and is less affected by jitter. This suggests that, although *Grab* is slower when moving objects, it provides more adequate movements useful when precisely position objects. On the precision task without limit, all tests had similar results, regardless of the metaphor. Users took about 30 seconds (μ=28,49s SD=3.01s) to conclude the test. This shows that the users had a good understanding of the limitation for each metaphor and gave up when they realized they had reached the best possible solution for each metaphor.

On the precision tests users felt more fatigue when using the *Point* that when using the *Mouse* or the *Grab*. We argue this is because most users point with their arm straight (almost locking their elbow), a position that quickly tires the user's arms (see Figure 3.1). For the precision test, users maintained this position for up to 30 seconds. On the other hand, with *Grab* users could quickly place an object and move the arm to a resting position. Users classified *Grab* as the easiest metaphor and *Mouse* as the hardest to control. During debriefing, most users felt that the *Grab* metaphor was an interesting way to interact even if just for selected tasks. Users also stated a preference for the middle distance, even though our results do not reflect any effect of distance of interaction. During tests, one user asked if he could change distance (move closer) because he was having trouble executing the task (the user did not have any physical reason to do so nor did he had bad eye sight).

For large-scale displays, distance does not affect interaction as much as task and the interaction metaphor. However, different metaphors seem more adequate to certain tasks. On our tests the *Point* metaphor seemed more appropriate to cover a large distance in the minimum amount of time, and *Grab* is best for precision tasks or tasks where all the objects can be reached without physical movement. We also conclude users achieve a better understanding of the *Point* metaphor, especially when compared with *Mouse*, and that fatigue only seems relevant in precision tasks, which are perhaps better executed with a *Grab* metaphor. Finally, our tests show the users found the metaphors easy to understand and seem likely to accept a system where the metaphor adapts to the active task.

This test provided some insight on what kind of variant users expect. The results indicated that people prefer variants inspired in pointing to those inspired by desktop analogies. Furthermore, based on the results obtained there is a strong suggestion that user movement does not affect interaction. Therefore, a logical next step is to try to better understand how a user, anchored to a position, interacts with a large-scale display using a pointing technique. The second tests, discussed next, follows up on the first test results and further explores pointing as an interaction technique, including discovering limitations of pointing for a stationary user on a large-scale displays.

3.2 Control Type and Parallax

People often interact with a large digital display by distant pointing, or directly pointing at the display with their finger, laser pointer, or other input devices. Ray pointing is a class of techniques that uses ray casting (the intersection of a ray with a surface) to determine where a person is pointing to, i.e., the precise cursor position on the distant display. Ray pointing is advocated as a natural interaction technique with these displays [Bolt, 1980; Olsen & Nielsen, 2001] as: it allows people to interact from the distance (as opposed to direct-input) [Parker *et al.* , 2005; Voida *et al.* , 2005]; it does not require any physical surface to operate on (as opposed to mouse-based pointing) [Oh & Stuerzlinger, 2002; Teather & Stuerzlinger, 2008]; it is easily understood by people as it builds upon everyday pointing gestures [Voida *et al.* , 2005]; and it allows multiple users to interact on the same display without their bodies physically getting in the way [Nacenta *et al.* , 2007]. Thus it is no surprise that ray pointing is increasingly used in both commercial and research systems, especially for large horizontal and vertical displays [Davis & Chen, 2002; Nacenta *et al.* , 2007; Parker *et al.* , 2005; Voida *et al.* , 2005]. Even game consoles are exploiting ray pointing for interaction (e.g., Nintendo Wii).

As large and very large displays (i.e., wall-sized displays) become widespread, ray pointing will likely become a primary way to interact with all kinds of interfaces from a distance or to access hard to reach areas of the display. This is why there is already a significant amount of literature devoted to the performance and different modes of ray pointing; for example, different variants of ray pointing are used and studied by Virtual Reality researchers [Argelaguet & Andujar, 2009; Bowman *et al.* , 2001; Pierce *et al.* , 1997], and there have been some efforts to characterize laser pointing (one of the possible ray pointing variants) for interaction with 2D interfaces [MacKenzie & Jusoh, 2001; Myers *et al.* , 2002; Oh & Stuerzlinger, 2002; Pavlovych & Stuerzlinger, 2009]. However, in our own design and implementation of large-display interfaces we have found that previous work in the area does not suffice because of several reasons. A)

Although the largest diversity of ray pointing alternatives has been studied in VR, ray pointing tasks and setups (e.g. CAVEs, stereoscopic displays) are substantially different to the more common 2D tasks that we are interested in. B) Previous empirical work on 2D tasks is focused almost exclusively on laser pointing or studies only small displays [MacKenzie & Jusoh, 2001; Myers *et al.* , 2001; Oh & Stuerzlinger, 2002]. C) With a few exceptions [Kopper *et al.* , 2010; Myers *et al.* , 2002] previous work for 2D environments does not try to provide explanations, or general principles of the differences in ray pointing.

We build upon this previous research by contributing a new experiment that a) tests of two fundamental 2D interaction tasks (targeting and tracing), one of which has not previously been studied in the context of ray pointing; b) compares of four different ray pointing variants (laser pointing, arrow pointing, image-plane pointing, and fixed-origin pointing) which map to two previously unstudied factors relevant to ray pointing in 2D tasks (parallax and control type); and c) identifies specific issues related to the large size of the displays, like the effect of the location of targets with respect to users. Our experimental results show that targeting performance is best explained by the control type factor, with rotational control being generally superior to position control, whereas for tracing tasks it is the presence of parallax that better explains differences between variants. The study also contributes rigorous support for the use of an angular formulation of Fitts's law [Kondraske, 1994; Kopper *et al.* , 2010] for large-displays, as opposed to the traditional linear formulation for both tasks. These results have important implications for designers and researchers alike. First, designers must now consider image-plane techniques and how the pointing device is held, especially if they want to support tracing tasks (e.g., menu selection, drawing). Second, researchers can now consider parallax and control type as important factors, and have solid evidence to start using angular adaptations of Fitts's law for interaction in large displays.

3.2.1 Ray Pointing Fundamentals

We define generic ray pointing as any cursor-movement technique that determines the position of the cursor through the intersection of a ray with a distant object or surface (see Figure 3.11, left). For our purposes, the distant object or surface is a large display. We consider only monocular absolute 2D pointing.

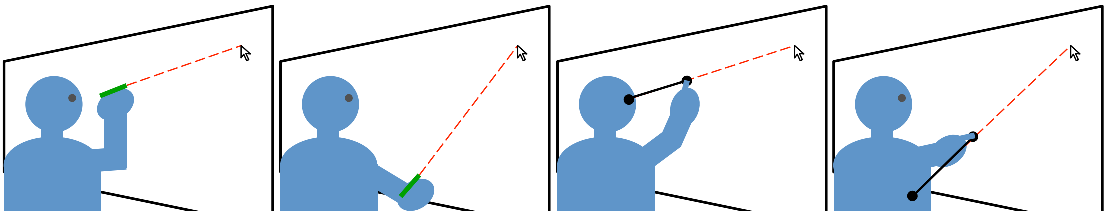


Figure 3.10: Four variants of ray pointing. *Left to right:* Arrow Pointing, Laser Pointing, Image-Plane pointing and Fixed-Origin pointing.

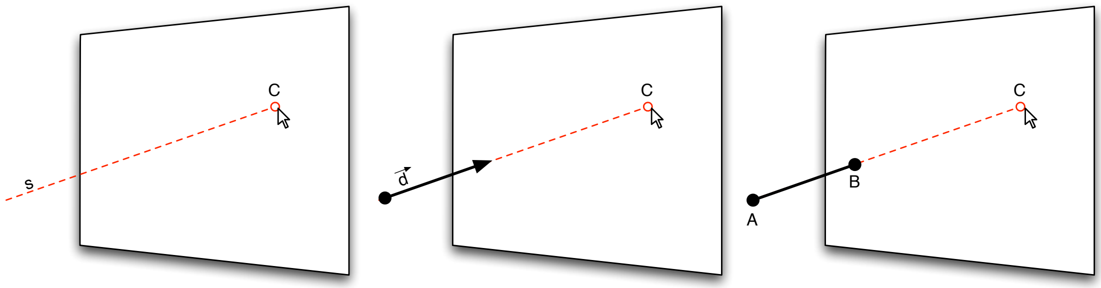


Figure 3.11: *Left:* Ray pointing specifies the location of the cursor through the intersection of a ray (s) with the display surface. *Center:* the ray (s) can be specified through a point (A) and a direction, or *Right:* through two points (A and B)

Regular Laser Pointing

The most common ray pointing variant is laser pointing. Here, the ray is specified directly by the position and direction of a physical device (Figure 3.10). The device might or might not be an actual laser; in fact, the only requirement is that the computer system has a way of determining the intersection of the ray with the screen surface. For example, vision technology or special markers on the hand recognize finger postures as a pointing device [Vogel & Balakrishnan, 2005]. Laser pointing has been proposed and implemented for cursor control in 2D interfaces many times (e.g., [Bolt, 1980; Cavens *et al.* , 2002; Davis & Chen, 2002; Oh & Sturzlinger, 2002; Olsen & Nielsen, 2001; Tse *et al.* , 2007]). It is often referred to as distant pointing, remote pointing, or virtual pointing. In our study, we implement a laser pointer via an infrared-marked wand tracked in 6DOF.

Arrow Pointing

Arrow pointing is a variant of laser pointing where we constraint the use of the pointer to be somewhat aligned with the user's eye (Figure 3.10). This mimics the real life way people aim when great precision is required (e.g., when using bow and arrow, or playing darts). Our implementation is identical to laser pointing, except now people are instructed to constrain their use of the wand by looking down its shaft at the screen, i.e., as if it were an arrow.

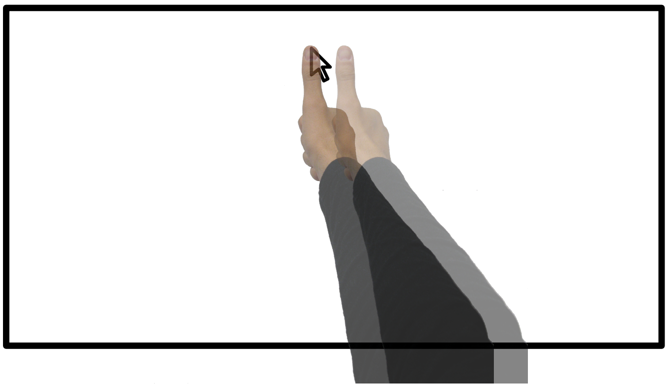


Figure 3.12: Image-plane pointing seen binocularly and focused on the distant display (cursor displayed on screen).

Image-Plane Pointing

An alternative ray pointing technique comes from the visual arts. Painters are often taught to place their thumb at arm's length between their eye and a painting to estimate the sizes and positions of painted objects. This technique has long been adopted in the field of virtual reality for the selection of 3D objects, where it is referred to as image-plane manipulation, occlusion selection, or the crushing heads technique [Argelaguet & Andujar, 2009; Bowman & Hodges, 1997; Hill & Johnson, 2008; Lee *et al.* , 2003; Ware & Lowther, 1997; Wingrave *et al.* , 2005]. The mechanism of image-plane pointing is simple: instead of determining the ray through the position and orientation of a pointing device, the ray is determined through two points in space: the user's eye location and another point in space that the user can control (e.g., the position of the tip of the thumb, of a pen, or the point of a pointing device; Figure 3.10). The effect is that the user can see the cursor aligned with the thumb (or device) in his/her field of view, even if they are actually at different depths (Figure 3.12). To a certain extent, image-plane pointing is similar to direct-input techniques (e.g., direct-touch) in that feedback and input overlap in the visual space of the user. Image-plane techniques require tracking (or approximating) the eye position, and are usually calibrated so that the dominant eye image aligns the finger or device with the cursor (however, binocular focusing on the distant surface still implies that two separate images of the finger or device are perceived by the user, as in Figure 3.12). In our studies we approximate eye position, the first point of the ray, in real time by placing markers on a hat. A person calibrates the vector between hat and eye before interaction by specifying the position of their dominant eye with another marker. We use the tip of a wand to specify the second point of the ray.

Fixed-Origin Pointing

We can relax image-plane pointing by placing one of the two points of the ray onto any fixed location (instead of the eye). This was explored somewhat by Shoemaker and colleagues in shadow reaching [Shoemaker *et al.* , 2007]. Shadow reaching allows the control of a large display through the shadow cast by a person on a large display illuminated from a fixed point behind the person. Because shadows are cast in straight lines, shadow reaching is geometrically equivalent to fixing point A on the location of the light source and using the pointing gesture of the person (usually the finger) as point B. Shoemaker *et al.* also speculate using a virtual light source that would move with the user at a certain fixed distance. We tested fixed-origin pointing, where the origin point of the ray is fixed in space. The user controls the other point to specify the ray's direction. We use an origin point near the bellybutton of the user so that the required pointing device movements are somewhat similar to shadow reaching, where the light source is located close to the floor and behind a person.

Ray Pointing Performance Factors

There are many possible factors that might affect ray pointing performance (e.g., grip, number of hands and filtering [Davis & Chen, 2002; Myers *et al.* , 2002]). In our study we concentrate only on control type and parallax, as described below. We chose these two factors based on previous empirical results from the literature (e.g., [Nacenta *et al.* , 2006b]) that have not been studied in relation to ray pointing.

Control Type

As explained previously, the ray of ray pointing can be specified through two points, or through a point and a rotation. Although geometrically equivalent,

		Control Type	
		Rotational	*Positional*
Parallax	*None*	Arrow Pointing	Image-Plane
	some	Laser Pointing	Fixed-Origin

Table 3.6: Technique classification according to the factors

our four control types (Table 3.6) result in different types of end-user movement. For example, people that use laser pointing and arrow pointing specify the position of the cursor mostly through the rotation of the device (we call this the rotational control type), whereas image-plane and fixed-origin techniques only require the specification of a single position on space and the orientation of the limbs or the device is mostly irrelevant (we call this the positional control type). Within these categories, we studied our four previously described methods, chosen as they represent design points in the design space defined by these two factors (see Table 3.6).

Parallax

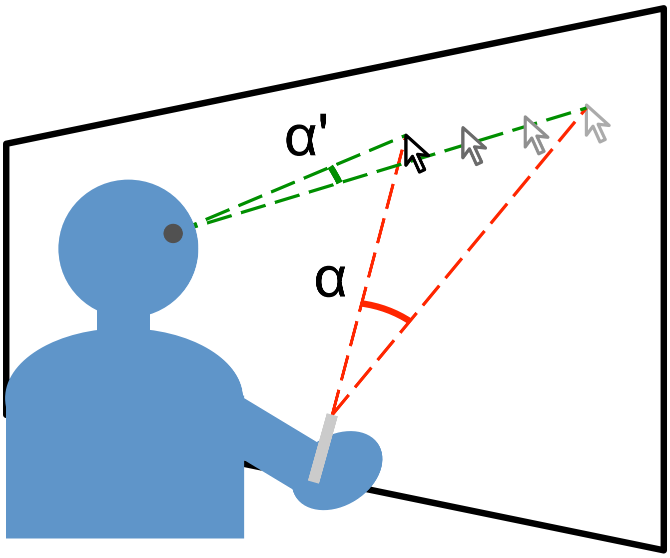


Figure 3.13: Parallax causes $\alpha' \neq \alpha$

Our other factor of interest is visual parallax (Figure 3.13). We define visual parallax as the distance between the center of rotation used to specify the pointing direction (usually a device) and the point of view of the user. In real-life aiming activities, parallax is usually avoided if precision is important. For example, sharp shooters align themselves in the direction of their weapons so that the line of view coincides with the shooting direction.

Our four techniques vary how people perceive parallax. The image-plane technique is, by definition, devoid of parallax. Arrow-pointing transforms laser pointer into an almost parallax-free technique, as the person aligns the pointing device with his/her line of sight.

3.2.2 Experiment 1 - Horizontal Targeting

Our first experiment tested targeting in the horizontal dimension. We were interested in testing targeting separately on this dimension because large displays (e.g., room-sized displays) tend to be much broader than tall, which implies that any effects due to the size of the display and the obliquity of distant areas would be most evident in these tasks, especially if the participant is close to the screen (Figure 3.14).

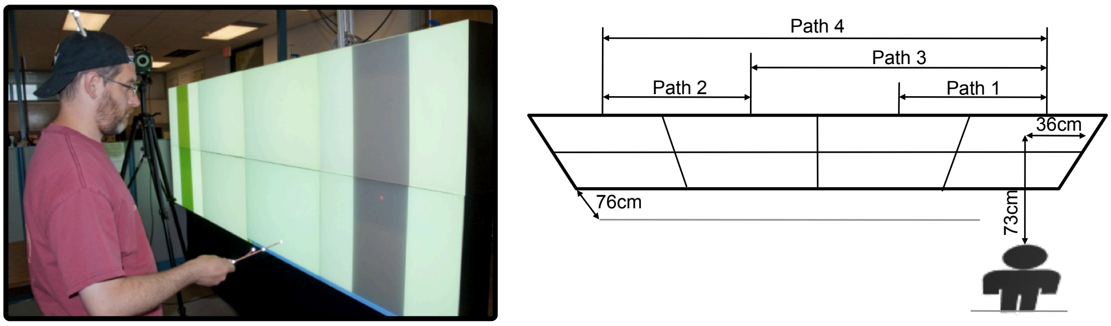


Figure 3.14: *Left:* The experimental setup during a horizontal targeting task. *Right:* Location of the participant and paths for the horizontal targeting task.

Method Adopted

We used a large wall display (292cm x 109cm) composed of 4x2 modular back-projected displays, each one with a resolution of 1024x768px (for a total of 4096x1536px). The modular displays are adjacent to each other with very narrow image seams (under 2mm). The displays rest on a table 76cm high so that the participant's head lines up approximately with the center of the top row of displays (see Figure 3.14). To accentuate the effects of large display widths, we asked participants to stand on a location approximately 73cm from the display and 36cm from its right edge, see Figure 3.14). From this point of view, the display covered approximately 100^o of the user's horizontal field of view, and 68^o vertically. We implemented the ray pointing variants using a 25cm wand and a cap equipped with reflective markers, whose positions were tracked by a VICON motion capture system. The position of the dominant eye of the user was updated in real time by using the position and orientation of the cap and the calibration data obtained before each block that involved the image-plane technique. Participants selected targets by clicking a mechanical button held in their non-dominant hand (we used a separate button, as pressing a button on the wand could affect its stability). In all techniques, a circular cursor was displayed on the screen. Our experimental software ran on a Core 2 Quad PC running Windows XP. The software was built on the .NET platform and used WPF for presentation. Both image and input were refreshed at a rate well above interactive rate (approximately 50Hz for display and input).

Task. The horizontal task follows the ISO 9241-9 one-direction tapping test recommendation [ISONorm, 2002]. Participants had to alternately move the cursor onto each of the target positions that composed a path, and click within its boundaries. The targets were vertical bands that covered the height of the display (see Figure 3.14). Targeting tasks varied in the width of the targets (100, 200, and 400px; 7.1, 14.2, and 28.5cm), the distance between target centers (1024, 2048, and 3072px; 73, 146 and 219cm), the position of the targets along the screen, and the direction (left-right or right-left). Figure 3.14 (right) shows a diagram with the four different paths, which multiplied by three different widths and two directions result in 24 distinct targeting tasks. Visual feedback of errors was provided in the form of color changes of the target.

Participants. Twelve participants recruited from a local university (4 female, 8 male; 24 to 36 years old) took part in the study for $15 remuneration. All participants were right-handed.

Procedure and Design. After signing a consent form, each participant provided some basic demographic information, was tested for eye dominance (to determine the dominant eye for the image-plane technique), and received instruction in the four ray pointing techniques.

Technique order was counterbalanced across subjects using a random Latin square design. Participants underwent a block of training for each technique (24 individual trials per technique involving all distances, positions, and target widths), and then, in the same order, two separate blocks of actual trials for each technique with three repetitions per individual task. Tasks were presented in order of increasing distance between targets and decreasing target width. After the end of the each technique's trials of the second block, the participants were asked to rate the perceived workload through a NASA TLX questionnaire [NASA, 1987].

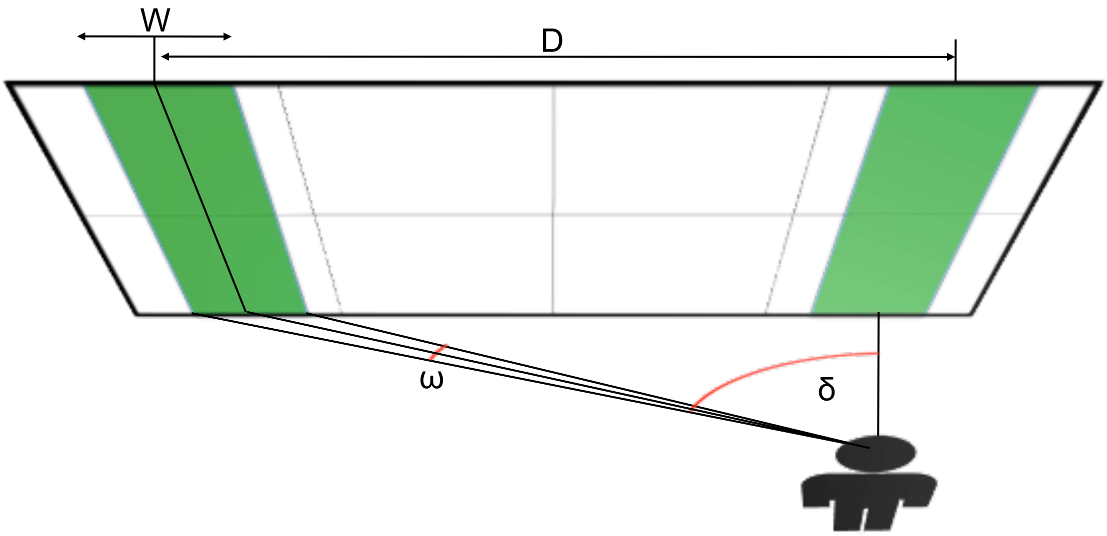


Figure 3.15: Geometrical relationships between D, W, δ and ω

At the end of the experiment, participants were asked to rank the techniques according to speed, accuracy, physical effort, and general preference. The entire experimental procedure took approximately 1 hour.

Measures and Analysis Methodology. For each trial we measured completion time, location of the cursor during the click, and whether it missed the target (error). We designed the experiment and the analysis to conform to the ISO 9241-9 recommendations [ISONorm, 2002] as well as the Fitts's study guidelines provided in [Soukoreff & MacKenzie, 2004]. As Soukoreff and MacKenzie recommend, we planned error and completion time comparisons as well as throughput comparisons. This requires the calculation of the index of difficulty of each task according to Fitts's law. D is the distance between targets and W the width of targets.

$$ID = \log_2(\frac{D}{W} + 1) \tag{3.2.1}$$

However (see Equation 3.2.1), in a very early stage of the research we realized that targeting tasks have different difficulties depending on their location on the display and the direction of targeting. Consider Figure 3.14: at the very least, targeting into the farthest region of the display (a distant target) should be harder than targeting onto the near target. Following reasoning parallel to Kopper and colleagues [Kopper *et al.* , 2010], we anticipated that the standard Fitts's model would not capture targeting time differences that can be derived from the geometrical relationships between the person, the display, and the target. Therefore, we performed two regressions on the data, one with the standard (linear) version of Fitts's index of difficulty(ID_{Linear}), and one with a variant of the formula that substitutes D and W for the subtended angles of D and W (δ and ω, see Equation 3.2.2) from the location of the user (see Figure 3.15):

$$ID_{angular} = \log_2(\frac{\delta}{\omega} + 1) \tag{3.2.2}$$

$$\alpha = atan(\frac{x_1}{d_p}) - atan(\frac{x_2}{d_p}) \tag{3.2.3}$$

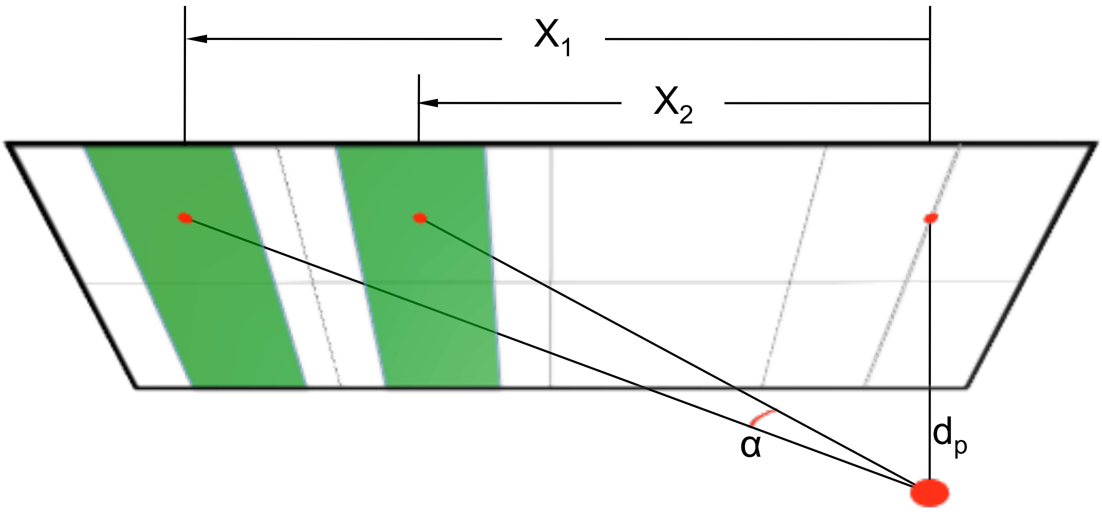


Figure 3.16: Calculation of subtended angle between X_1 and X_2

The subtended angles are calculated through standard trigonometric procedures with the formula described in Equation 3.2.3. X_1 and X_2 correspond to the horizontal coordinates of the extreme points of the linear distance whose angle we are calculating (see Figure 3.16). In our experimental setup, ID_{Linear} and $ID_{Angular}$ calculations proved substantially different from each other because of the large size of the display and the position of the user. Figure 3.17 plots the ID_{Linear} of all tasks against their $ID_{Angular}$. If, as we hypothesize, $ID_{Angular}$ predicts performance significantly better than ID_{Linear} across participants, it would make sense to use this instead to calculate throughput. In either case, for the throughput calculation we apply the effective width corrections as argued in

[Soukoreff & MacKenzie, 2004]. The calculation of the angle was done using a point 73cm in the direction perpendicular to the top right modular display, which approximates the position of the head of the user.

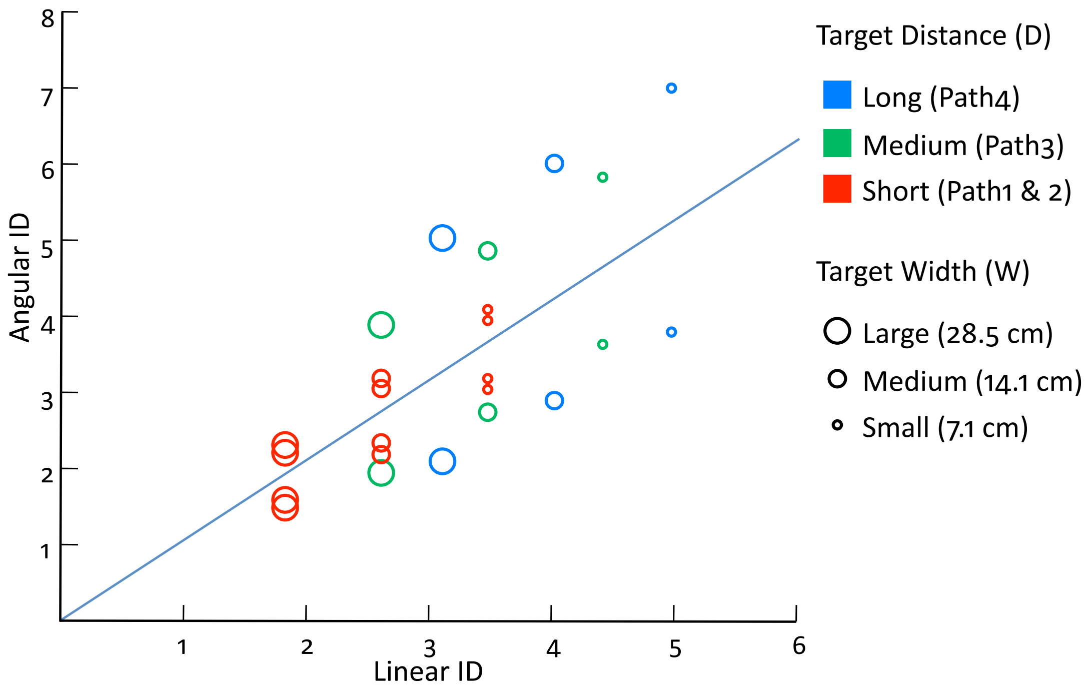


Figure 3.17: Relationships between linear IDs and angular IDs

Results

We begin with our analysis of fit of the linear and angular models, follow by the performance analysis, and end with a summary of the subjective measures results. We performed analysis on throughput, time and error for all tasks. For clarity, we omit reporting those analyses that are redundant.

Analysis of fit. We did a per-participant, regression analysis of trial completion time for each technique. Using ID_{Linear} as a predictor variable shows an average $R^2 = 0.33$. For $ID_{Angular}$, the average is $R^2 = 0.61$. That is, using the standard ID accounts for only 33% of the variance when used to predict the duration of a horizontal targeting movement. The angular model is much better, as it accounts for roughly twice that proportion. In every single case (all techniques, all participants), the fit of the angular model was superior to the linear model.

All p-values of the regression's ANOVA for both models are under 0.001, indicating that the probability of getting these results due to chance is extremely low.

The same two-fold improvement is apparent when we try to fit all data (including differences between participants and between techniques) to particular IDs. With the linear model, the fit is 20% ($R^2 = 0.20$). The same regression with the angular model results in an average fit of 38% ($R^2 = 0.38$).

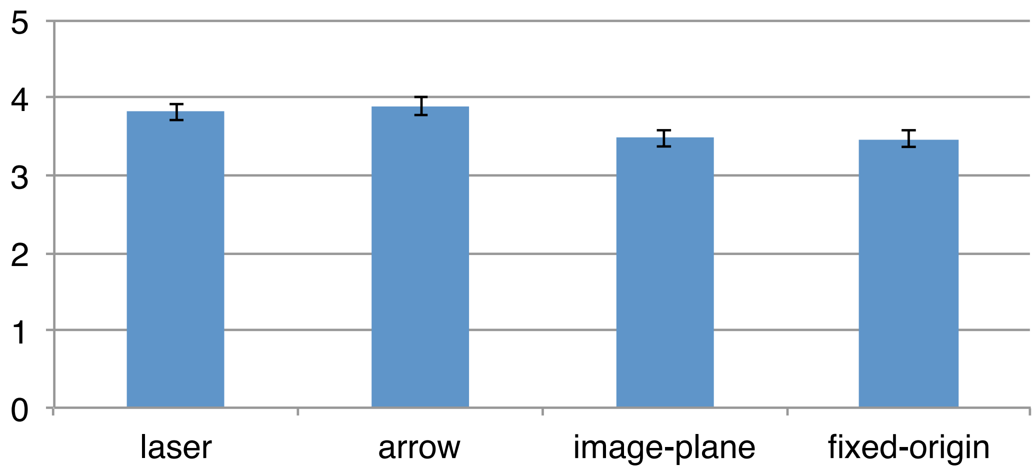


Figure 3.18: Throughput values (in bits/s) for the horizontal targeting task. Error bars represent standard error.

Analysis of performance measures. On average, the fastest technique was laser pointer (μ = 1015ms) followed by arrow pointing (μ = 1057ms), image-plane pointing (μ = 1139ms) with fixed-origin pointing as the slowest (μ = 1168ms; see Figure 3.18). A repeated-measures ANOVA of throughput (which amalgamates accuracy and speed measures) with technique and task as factors shows a strong effect of technique (F3,33 = 9.99, p < 0.001, x^2 = .47), where the ordering of average throughput is the same as for completion times (μ_{laser} = 4.05 b/s, μ_{arrow} = 3.8 b/s, $\mu_{image-plane}$ = 3.6 b/s, $\mu_{fixed-origin}$ = 3.4 b/s). Post-hoc tests (corrected for multiple comparisons) show strong statistical differences between laser and the two worst performing techniques (image-plane and fixed-origin; both p < 0.004), while it shows differences that approach statistical significance between laser and arrow (p < 0.02) and arrow and fixed origin pointing (p < 0.022). For error analysis we performed a non-parametric paired-samples test (Friedman) which shows a significant effect of technique on number of errors ($x^2(12)$ = 10.4, p < 0.015). Fixed-origin pointing was the technique with the largest number of errors (6.3%) followed by arrow pointing (5.15%), laser pointing (4.1%)

and image-plane pointing (3.5%). To summarize, the results of this section indicate a general advantage of laser pointing over image-plane and fixed-origin pointing, whereas arrow pointing finds itself somewhere in between these two groups.

Analysis of subjective measures. Non-parametric paired-measures tests of the subjective workload TLX questionnaires only yielded significant differences between techniques in the physical demand ($x^2(12) = 9.4$, $p < 0.024$) and effort questions ($x^2(12) = 8.3$, $p < 0.039$). The mean answers to these questions are shown in Figure 3.19). Consistent with performance measures, participant preference rankings favored laser and arrow and placed fixed-origin pointing as the least-liked (see Table 3.7).

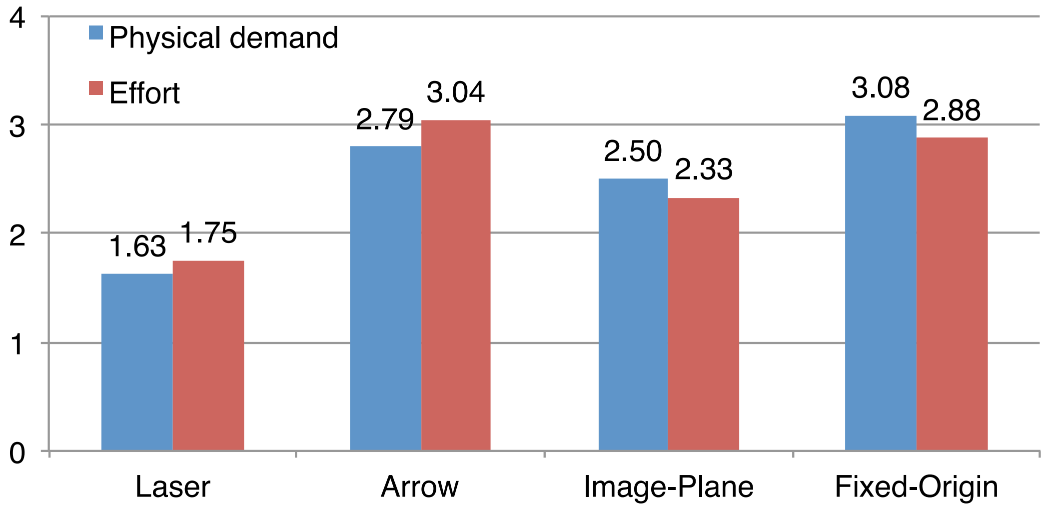


Figure 3.19: Average physical demand and effort responses (out of a 7-point Likert scale) in the horizontal targeting task (lower means less effort and less physical demand respectively).

	Best 1	2	3	Worst 4
Laser	**7**	3	2	0
Arrow	4	2	3	3
Image-Plane	1	**5**	4	2
Fixed-Origin	0	2	3	**7**

Table 3.7: Preference ranks for the horizontal targeting task

3.2.3 Experiment 2 - Vertical Targeting

Our second experiment tests vertical targeting tasks with a double purpose: to generalize the performance results of the horizontal task to vertical movements, and to investigate the effects of the different aspects of parallax in performance. Our techniques were chosen to vary in the amount of parallax (laser pointing and fixed-origin pointing have large parallax, whereas arrow pointing and image-plane pointing have little or no parallax). However, parallax does not affect the horizontal targeting task because the direction of movement is perpendicular to the direction of parallax (i.e., horizontal movement angles are similar for hand and eye, whereas vertical movement angles are very different). Even so, we hypothesized that parallax might affect a vertical task since the targets cover different angles from the eye or from the pointing device.

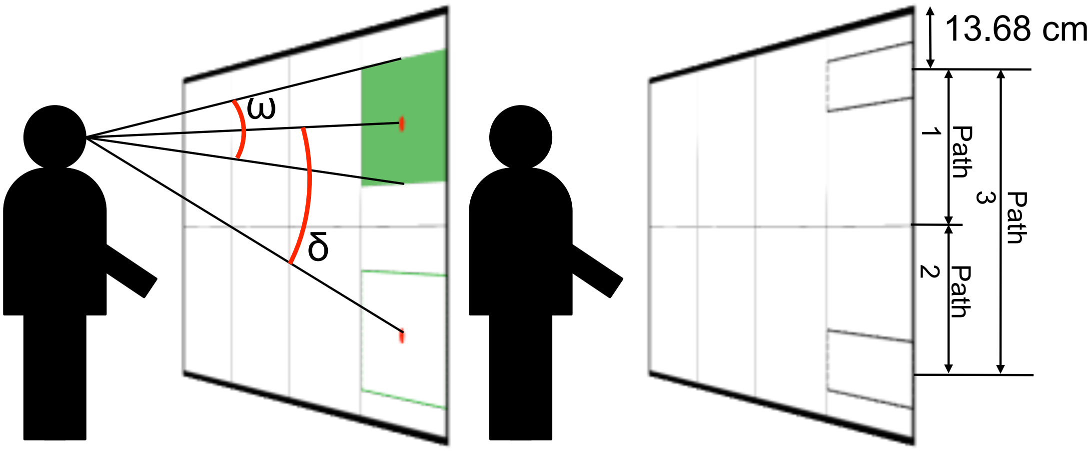


Figure 3.20: Vertical task angles and paths

Method Adopted

The apparatus and location of the participant with respect to the screen was identical to experiment 1. The task was performed on only two rightmost display modules (see Figure 3.20).

Task. The vertical task was equivalent to the horizontal task, but in the vertical direction. Pairs of targets were as wide as the modular displays in front of the user, and located at different heights. Targeting tasks varied in the height span of the targets (50, 100, and 200px; 37, 73, and 146cm), the position of the targets along the screen (centered at 13.7, 54.7, and 95.8cm from the top of the display) and the path. Figure 3.20 (right) shows diagrams of the three different paths.

This results in 18 distinct targeting tasks when combined with three different widths and two directions (3x3x2).

Participants. A different set of 12 participants (5 female, 7 male; 20 to 40 years old) took part in the study for $15 remuneration. Experiment 2 and 3 shared the same participants.

Procedure and Design. The general procedure was identical to experiment 1, except that participants performed all tasks from experiment 2 and then all tasks from experiment 3 within the same session before they filled the questionnaires (ranking and NASA TLX). For the vertical targeting task participants carried out three blocks of trials, the first of which was considered training. Each block contained five repetitions of each of the different targeting tasks with each of the techniques. The tasks were presented in increasing order of distance between targets and decreasing target height.

Measures and Analysis Methodology. For each trial we measured completion time, location of the cursor during the click, and whether it missed the target (error). Since the display we used for our experiment is much broader than tall, we did not expect to find important differences in fit between the angular and linear models; nevertheless we ran regressions with both and used the better fitting model to calculate throughput. For the angular model calculations we used again a fixed point in space that approximates the user's eye position.

3.2.4 Results

Analysis of fit. We ran regressions of the data with the angular and linear models and found the linear model to have a slightly (but statistically significant) better fit than the angular model ($R^2_{Linear} = 0.37 > R^2_{Angular} = 0.33$). Calculations of throughput were thus based on the linear model.

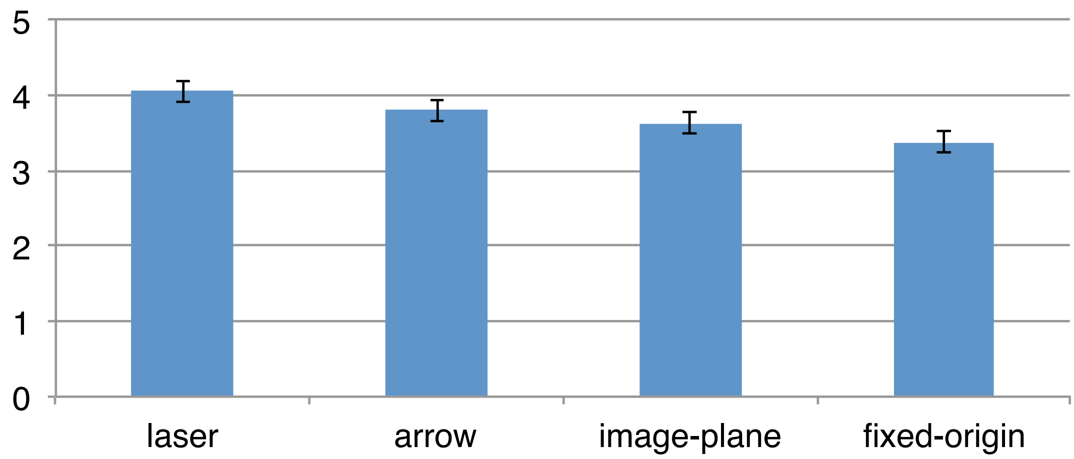


Figure 3.21: Throughput (in bits/s) for the vertical targeting task. Error bars represent standard error.

Analysis of performance measure. The averages of task completion times are in identical order to those found in the horizontal tasks (μ_{laser} = 391ms, μ_{arrow} = $421ms$, $\mu_{image-plane}$ = 453ms, $\mu_{fixed-origin}$ = 453ms; see Figure 3.21). For throughput, the repeated-measures ANOVA shows a strong effect of technique as well (F3,33 = 8.5, p < 0.001, x^2 = .43). For the vertical task, arrow had the highest average throughput (μ = 3.89 b/s), followed by laser (μ = 3.82 b/s), image-plane (μ = 3.48 b/s), and fixed-origin pointing (μ = 3.47 b/s). The throughput rankings are in slightly different order because throughput depends of both speed and errors.

The post-hoc tests show statistically significant differences between laser and image-plane (p<0.001), laser and fixed-origin (p<0.003), arrow and image-plane (p<0.003) and arrow and fixed-origin (p<0.009), this last one is only marginally significant with a Bonferroni adjustment for multiple comparisons a = 0.05/6 = 0.0083).

The Friedman test for errors shows statistical differences (x^2(12) = 9.14, p < 0.027). For vertical tasks, fixed-origin pointing has again the largest error rate (3.6%), followed by laser (3.4%), arrow (3.2%), and with image-plane again as the most accurate (2.1%).

In summary, performance in the vertical task is similar to the horizontal task, except that post-hoc tests show more power: we can completely separate techniques in two groups of performance, with laser and arrow outperforming the rest.

Analysis of subjective measures. Because experiments 2 and 3 were grouped, the post-study questionnaire will be discussed as part of experiment 3.

3.2.5 Experiment 3 - Tracing

To conclude the experiments, the performance of the techniques was examined under a tracing task.

Method Adopted

Apparatus and participants. The apparatus and location of the participant with respect to the screen was identical to experiment 1 and 2, although tracing tasks took place across the whole width of the display.

Task The screen presented a rectangle (a tunnel) with a square at one end (Figure 3.22). Participants were instructed to enter the rectangle through the non-square entrance, and to reach the square while remaining within the rectangular tunnel. When the square was reached, it disappeared and a square appeared at the other end starting a new trial.

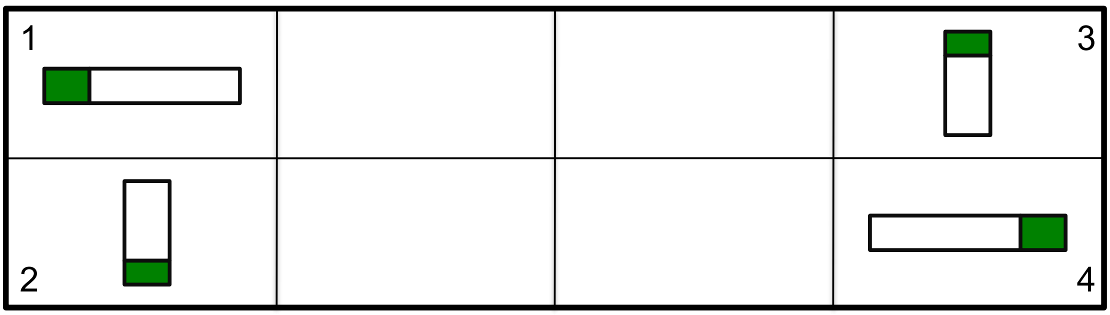


Figure 3.22: Tracing task tunnels and positions

The tunnels were always 384x96 pixels (27.4x6.8 cm^2), which results in a tracing index of difficulty of 4 (length/width). However, they appeared in four different locations (centered on the modular displays at the corners of the display wall) and with two different orientations (horizontal and vertical), for a total of eight different tracing tasks (see Figure 3.22). We did not consider different directions (e.g., top-bottom and bottom-top) as different tasks because the angles covered are the same, i.e., they are independent of the tracing direction. For error feedback, we changed the tunnel's outline color when the cursor was moved outside of it.

Design Participants carried out three blocks of tracing tasks for each technique, with the first block for training. For each block, participants performed six repetitions of each of the eight tasks for each technique. Tasks were presented always in the same order (vertical to horizontal, top to bottom, left modular display to right modular display).

Measures and Analysis Methodology. For each trial we measured completion time and average distance to the longitudinal line of the tunnel. We tested the model fit with linear and angular measures (analogous to those in experiment 1). For the calculation of ID in tracing we used the formula suggested by the ISO 9241-0 standard. D is the length of the tunnel and W is the width that the cursor can move transversally without leaving of the tunnel (our cursor was considered to be of zero diameter). For the angular calculations, we used approximations based on the perceived angles of width and length at the center of each tunnel. Angles were calculated from the same fixed point used for experiment 1. Angular IDs for the eight tasks range from 1.27 to 12.60 b/s. As for experiment 1, we would use the IDs of the model that offered the best fit for the throughput calculation, and apply the corresponding adjustment for accuracy.

3.2.6 Results

Analysis of fit. The angular model shows an average fit of the data comparable to the fit for the horizontal targeting task (R^2 = 0.50). Since we only tested one linear index of difficulty, comparing the fit of both models is equivalent to testing whether the angular regression is significant. The ANOVA test measure of the angular regression does exactly that. This result held in all but one of the 48 regressions (12 participants x 4 techniques), with p < 0.002. An omnibus regression using all participants and all techniques simultaneously produced a p < 0.0001, and R^2 = 0.37.

Analysis of performance measures. The repeated-measures ANOVA of throughput calculated from the angular indexes of difficulty shows a strong main effect of technique (F3,33 = 12.227, p < 0.001, x^2 = .53). Image-plane had, on average, the highest throughput (μ = 107 b/s), followed by arrow (μ = 83 b/s). Laser and fixed-origin exhibited much lower performance (μ_{laser} = 56 b/s, $\mu_{fixed-origin}$ = 65 b/s; see Figure 3.23). Post-hoc tests statistically differentiate image-plane

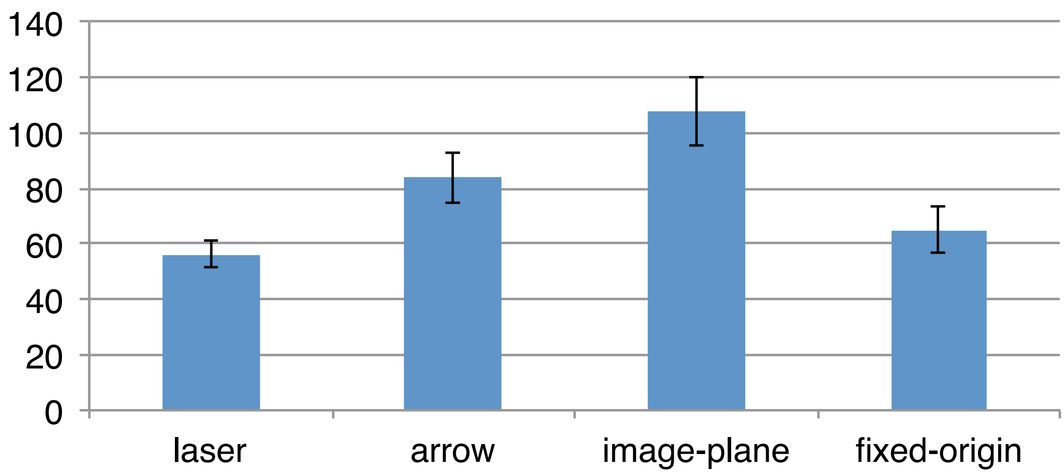


Figure 3.23: Throughput (in b/s) for the tracing task. Error bars indicate standard error.

from laser and fixed-origin (both p < 0.002), and arrow from laser (p < 0.001). Note that the throughput values for tracing are not necessarily comparable with those of targeting, and that the adjustment for accuracy suggested in [Soukoreff & MacKenzie, 2004] greatly increases the effective angular IDs. A repeated-measures ANOVA of the average deviation with respect to the middle of the tunnel also shows a strong effect of technique (F3,33 = 4.9, p < 0.006, x^2 = 0.30). The most accurate technique was image-plane (μ = 18.4px) followed by the rest in a very tight group (μ_{arrow} = 21.7px, μ_{laser} = 22.8px, $\mu_{fixed-origin}$ = 22.8px). Post-hoc tests only show statistically significant (or marginally significant) differences between image-plane and the rest.

Analysis of subjective measures. As the questionnaire encompassed experiments 2 and 3, the subjective measures reflect the opinions of subjects in both tasks (vertical targeting and tracing). The measures of the NASA TLX only showed statistical differences between techniques for the physical demand question ($x^2(12)$ = 7.8, p < 0.024). Interestingly, subjects judged the arrow as the least physically demanding technique, whereas the horizontal targeting task arrow was considered the second most demanding. The preference rankings (Table 3.8) also show very different results than experiment 1.

3.2.7 Discussion

As large-scale display are now more available, it becomes increasingly important to develop suitable input mechanisms that account for interacting at a dis-

	Best			Worst
	1	2	3	4
Laser	2	2	5	3
Arrow	**5**	**5**	2	0
Image-Plane	1	**5**	1	**5**
Fixed-Origin	4	0	4	4

Table 3.8: Preference ranks for experiment 2 and 3

tance. Moreover, ray pointing is the class of pointing techniques that currently offers the most promise. As such, these techniques deserve very close scrutiny and refinement under various conditions. Specifically, we looked at two factors that influence ray pointing: control type and parallax. We tested four ray casting variants and two tasks (targeting and tracing), and found that control type affects targeting and parallax affects tracing. Furthermore, we provide strong evidence that supports the use of angular indexes of difficulty for targeting and tracing tasks with any of the absolute ray pointing techniques. We discuss the findings around three perspectives: the ray pointing technique variants, the effect of the large display, and the limitations of the experiments.

Differences between Ray Pointing Variants

Our targeting experiments revealed differences in targeting performance of up to 10%. Arrow and laser (both rotational techniques) performed better than their positional counterparts, even in vertical targeting for which we had hypothesized that parallax would have a strong influence. Counter to our intuitions, this suggests that parallax is not crucial for targeting tasks, at least when cursor feedback is present (as in our experiment). The advantage of rotational techniques over image-plane pointing is somewhat surprising because it contradicts some of the evidence from VR studies that found image-plane to be more efficient. This evidence implies that targeting tasks are different for 3D and 2D environments, and that practitioners and designers alike should be careful not to extrapolate results from one field into the other, regardless of the apparent similarity of the tasks. Regarding the tracing task, the story takes a different turn as laser pointer is no longer the leading technique. Instead, image-plane and arrow (the two parallax-free techniques) perform best, which suggests that parallax is a critical factor for such task. Therefore, designers of large-scale interactive systems should consider which ray pointing techniques fits suits better to the user task (e.g.: steering *vs* selecting).

As observed, the same technique can produce different outcomes, which brings up interesting research questions: will users naturally adopt the most advantageous use mode of the device? Alternately, how can we design a pointer that encourages the best use of the device? Overall, our results suggest that arrow is a good choice for both task types; it performs close to laser pointing in selection tasks, and better in steering tasks. Arrow is also relatively straightforward to implement compared to image-plane (which requires some degree of head-tracking). Although participants found arrow pointing tiring for the horizontal targeting task, they did not for experiments 2 and 3. Longer term studies should test whether physical effort is really a serious issue; however, the current data suggests that, when comparing ray-pointing techniques, user perception of effort is linked to their ability to perform up to their expectations with that technique rather than to the differences on how the device is held. Furthermore, we anticipate that the magnitude of differences found in our experiments will be relevant to other research questions, e.g., when performance is very important or errors are very costly. More extreme effects are also possible for larger displays and smaller targets.

Models for Ray Pointing in Very Large Displays

We are not the first to suggest an angular adaptation of ID calculation for ray-pointing tasks. Kopper *et al.* [Kopper *et al.* , 2010] proposed a number of alternative models for ray pointing, some of which use angular measures. Although their data and our results both combine to support the use of angular formulations of Fitts's law for ray pointing, their focus was specifically on comparing (slightly different) models for what we call laser pointing. Their fit calculations are performed after averaging all participants' trials for each ID, and result in very high fit values ($R^2 > .75$), even for models using linear distance and width. This approach is problematic because it eliminates most within-participant variance, and only highlights small differences in fit between models that do not strongly advocate for any particular model. Instead, we decided to compare the linear formulation against the simplest angular model, without artificially eliminating variance. This resulted in a more accurate estimation of the real variance explained by the models and, most importantly, on very large differences in fit between the two models (up to 84% increase in fit) that strongly advocate for the use of angular models. Additionally, the differences that we found in model fit were consistent across all techniques (not only ray pointing) and across tasks (not only targeting). This suggests that the better fit of the angular model is not only due to the rotational nature of input, but also due to the differences in perception of objects that are at different angles and are seen from different perspectives. Having this strong evidence to support angular models is critical for research and design of large display interaction; using the linear model in large-display experiments (the current approach) will introduce a large amount of noise, which can dramatically reduce the power of statistical tests. Similarly, when using targeting and menu activation estimations for the design of large display interfaces, it is important to know that different locations in the display will be affected differently depending on the position of the user.

Limitations

Any experiment is necessarily limited in the amount of conditions and factors that it can test, and ours is not an exception. Most notably, our experiment only tested one distance from the display. It is possible that other models better reflect distance variability, but this remains an open question (see also the discussion in [Kopper *et al.* , 2010]). Our evidence on the factors that cause technique performance differences is also not definitive. For example, establishing a strong causal link between parallax and poor tracing performance requires further research. Furthermore, targeting results for the laser variants might exhibit an exaggeration effect due to high familiarity of the device in comparison with the positional techniques. We added significant amounts of training to the experiment to avoid this biasing, but this issue also requires further research. Finally, other slight variants to the test design can be addressed in order to perceive the bigger picture of ray pointing techniques, e.g., do users perform differently if we change the location of the click button or handedness?

Lessons for Practitioners

We summarize the implications for researchers and designers in four main points:

- If targeting is important, choose laser-style ray pointing;

- For modeling targeting and tracing tasks that span a large angle from the point of view of the user, use a performance model that takes angles into account;
- For tracing tasks choose parallax-free pointing;
- Training people to use a pointing device from a location close to the eye (arrow pointing) provides good performance for both targeting and tracing tasks.

3.3 Overall Discussion

In this section we discuss the combined result of both user tests. The following section provides insights on findings related to: Large-Scale displays, interaction techniques, and user position in Large-Scale Display scenarios.

3.3.1 Large-scale Display

Users are comfortable with pointing as an interaction technique, even though they had to maintain an upright position during the interaction. On all test sessions, users were able to finish the required tasks without major problems. The only issue detected through the tests relates directly to fatigue during prolonged pointing interaction without no breaks (some tests amounted up to 60 minutes).

Interacting while standing can bring fatigue, more so if users have to gesticulate to issue commands. Pointing requires gestures for interaction, but some variants do not require users to keep their arms in a constant strain. The laser technique is ideal for fatigue reduction. During out tests we specifically asked users to hold the laser close to their waist, but the technique allows for a number of ways to grab the device and point that can help reduce fatigue. Moreover, between pointing actions users can relax their arm and further reduce fatigue issues.

3.3.2 Distance

Our results suggest that distance does not have a direct correlation with pointing performance. Pointing is affected by jitter, the effects of which increase

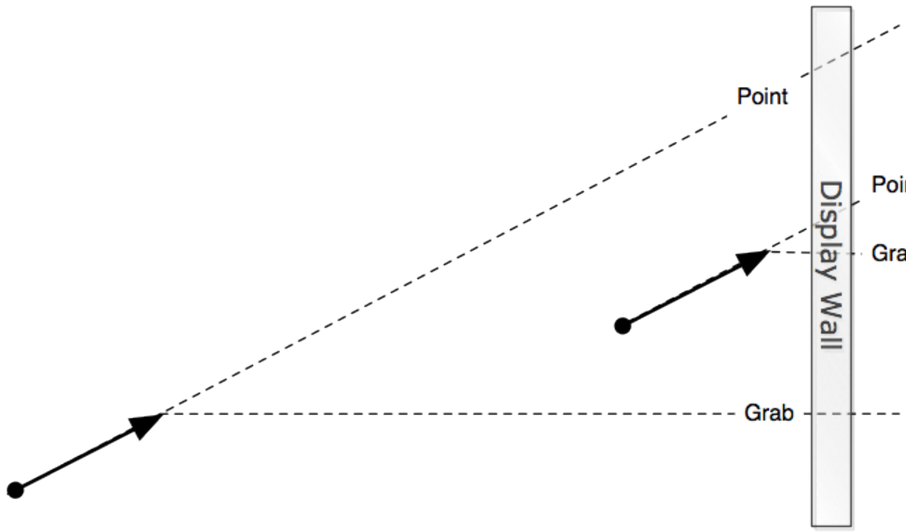


Figure 3.24: Distance exaggerate the effect or rotational control type. For example, for the same user position, a pointing metaphor is very similar to grab for close interaction. For distant interaction, the grab and the point metaphor show cursors with distinct screen positions.

with distance [Myers *et al.* , 2002], and control type (see figure 3.24). Moreover, results for the tracing task show that subjects found it easier to interact with objects in front of them (closer) that with objects in the peripheral view (slightly distant). Users naturally dealt with all these factors and showed no visible signs of being significantly affected by them.

Our conclusion is that distance does not directly affect interaction, but there are other variables that might depend on user position (such as jitter or parallax). Therefore, to reduce the effect of these variables, it is important that users have a coherent interaction technique for all distances, and not force the users to interact at a specific position.

3.3.3 Interaction Techniques

We tested five different pointing metaphor variants. On both tests, users selected the closest variant to the pointing gesture used in human to human dialogs. Moreover, factors such as control type and parallax do seem to affect interaction even if only for a select set of tasks. Overall, there is one set of techniques that seems to be more useful in everyday interaction: laser-like interaction (vectorized control type with parallax). It provides enough freedom to reduce fatigue and allows users to move freely throughout the interaction zone. The first test suggests that grab-like metaphors are useful for close distance interaction with objects that are within arms reach. However, pointing with a laser-like technique achieves similar results when close enough to the display (see figure 3.24). Therefore, if the implementation provides enough freedom to the users, pointing can replace the grab metaphor when interacting closer to the display.

3.3.4 Practical relevance of results

While most of the results presented should be taken into account, it is important to understand the results and where to apply them. Although we see a difference of 10% in ray pointing variants, these will not be relevant in situations where speed is not of the utmost importance. Living room scenarios, where people interact to view photos or select a movie are not bottlenecked by speed and, in this cases, fatigue or comfort should overcome performance. On the other hand, there are scenarios where 10% is needed. Scenarios such as airport control towers, control rooms for armed forces or firefighters headquarters need to take into account time to response and, at the same time, maintain a large number of elements on-screen, and might welcome the improvement in performance.

4

Evaluating Pointing on a real world scenario

This chapter explores the lessons learned on a real world scenario. We present how a pointing metaphor was applied to a large-scale display application for design review meetings. We start by describing the context in with this work was conducted, presenting the system requirements and providing an overview of the system architecture. We then focus on the user interface and describe how the user interface was influenced by applying the guidelines obtained in the previous chapter. From previous experience, pointing is important to user interaction on large-scale displays. To understand how a pointing interface influences user experience, we conducted user tests with twenty-two subjects. Results suggest that our approach to pointing made it adequate both to support design review and comfortable to learn. Based on these results, we conclude with observations on how users applied and adopted the interaction metaphors during design review meetings.

4.1 A design review application for large-scale displays

To illustrate the results obtained in the previous chapter we developed a multi-modal multi-user system for real-time design review meetings that use a large-scale display. The interface was designed to support pointing devices, and followed a stroke-based model, instead of the conventional point & click metaphor. The main modality was a laser pointing device. Secondary modalities included arm gestures, speech and hand-held devices, all of which can be combined with pointing to produce multi-modal dialogs. This application was developed within the context of a European project[1] targeted at the creative industries, in particular architectural firms. By applying user-centered design, we were able to observe how pointing techniques can support complex tasks such as navigation, annotation and object editing.

4.1.1 System Requirements

System requirements were obtained through a user-centered design method. To understand how people would likely operate the system, we conducted interviews with expert users (architects). Each interview was conducted by an HCI expert and two architects. We asked them to describe their current methodology, in particular, how they present their ideas to the end customer. This procedure allowed us to understand how architects execute a project. We then were able to and extract system requirements.

During a project life-cycle, customer review of architectural 3D models is one of the major tasks performed by architects. Architects described a typical scenario as follows: *"The project review usually takes place at the office, where customers review the design alternatives for the project at hand. Taking into account information collected during on-site visits, we present different design alternatives to support discussion between the architects and the customer."* Additionally, architectural projects include complex information such as plans, scale models, and hand sketches, which can be annotated using conventional pen and paper. Moreover, digital support such as CAD 3D Models or renderings are also employed.

When confronted with the possibility of using a large-scale displays, expert users suggested new functionalities that are not yet available in design review tools. In particular, architects commented that using a large-scale display could

[1] IMPROVE, european project, FP6: IST-2003-004785

allow them to show models at scales not possible with their current tool. Indeed, physical models are built at a much smaller scale, starting from 1:200. They also commented that it would create an immersive environment where clients could get a better feeling of the concept being presented. When specifically asked about possible interactions with the model, architects stated that mobility was a crucial aspect, since current customer meetings often take place in a meeting room using the aid of paper plans and models, and a projector powered by a single laptop to display renderings of the proposed design. When projecting a 3D model, architects often use deictic aids, including laser pointers, to highlight design features. When changing renderings, they are required to operate the computer, using keyboard and mouse. This interrupts the workflow of the presentation and creates disruptions in the dialog with the customer. After being presented with the available technology (described in Section 2.7) architects commented that an interface that would not disrupt their workflow during presentation would be of interest.

We set to develop a design review application with large-scale displays in mind, where we would devise novel interaction techniques adequate to the task on hand. Based on the user feedback, we identified several functional requirements for the design review application. These can be grouped in four major categories: navigation, annotation, object editing and collaborative support. As for interaction requirements, we set to take advantage of user familiarity with pen based devices such as stylus tablets, interactive pen displays, laser pointers and other pointing devices. Moreover, architects were interested in new developments such as speech commands and gestures, therefore these were included in the interface design as secondary modalities. As can be intuited, these user requirements indicated the need for a multimodal input lexicon combining a pointing devices, speech, and body gestures.

Navigation

To explore virtual scenes, we devised three different metaphors for 3D spatial interaction: flying, walking and examining.

When flying or examining, the observer is detached from his/her physical location. This detachment provides means to reach locations without regard to "real" physical constrains. The flying mode offers a better perspective than walking with a maximum degree of freedom. The examining mode provides navigation techniques that constrain spatial references to a single object, providing a focused view of a target through zooming and rotations around the object center.

Third, walking provides a natural way for the user to explore the model. By using this modality, users experience architectural model from a first person perspective. While being more realistic and representing everyday exploration, a walkthrough is subject to obstacles such as stairs, walls, furniture or doors.

Annotations

Architects and designers often take notes (audio, visual or documentary). Over the course of a project these might be helpful at later design stages. Indeed, annotations allow users to attach comments and thoughts to a model entity as the design progresses. Thus, notes possess the character of an addendum to capture what cannot be expressed in other ways. By capturing design intentions, modifications, or suggestions users are able to identify areas of interest and create annotations, either in visual (drawings and post-it) or in multimedia formats (audio and video). The user interface should provide the different functionalities to support annotations: hide and unhide, create, filter, and delete.

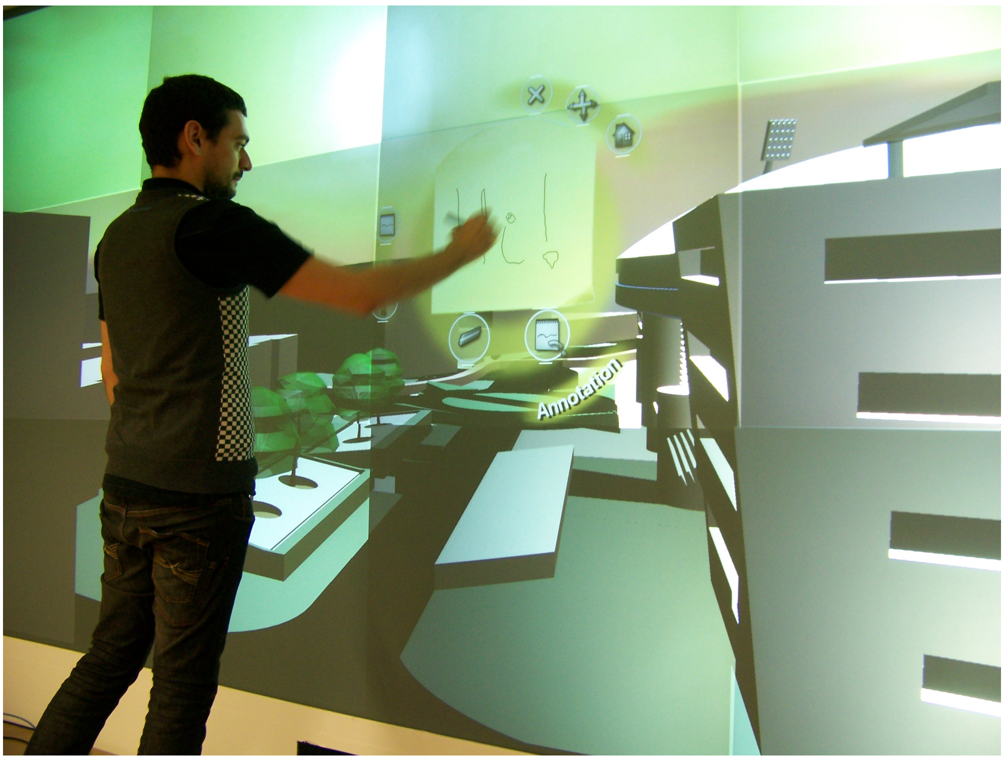


Figure 4.1: User annotating a 3D object by means of direct sketch based input, via the laser pen device.

Object editing

3D scenes are one of the main deliverables of an architectural project. Thus, the system must provide efficient means to edit 3D objects. In particular, during design review, minor modifications are often performed by experts. We identified a strong need to support simple object geometry editing and to rapidly insert graphics primitives such as cube, sphere, cone, cylinder, and plane. Moreover, basic geometrical transformations found in CAD applications are expected, such as translate, scale, and rotate.

Collaborative Review

Design review meetings are rarely single user scenarios. Therefore, we must address multi-user collaboration via tools and modalities that provide interactive capabilities to a group of architects or clients. The large-scale display serves as the natural output device for multiple users, due to its large form-factor and high-resolution capabilities. Thus architects or clients naturally gather around the display in the course of a review session. During the review session, users tend to collaborate via annotation. Although navigation and editing are also relevant, there were identified as less important for collaboration during design reviews.

To support freedom of movement, the system should be flexible to allow different input modalities. Thus, a user interacting close to the screen can opt for sketch-based input via the pointing device, whereas a second user, at a larger distance to the screen, can choose other modalities, such as gestures or speech, that may seem more appropriate to his/her situation.

4.1.2 Design Goals

Based on the user requirements briefly outlined above, we set out to design a real-world application that would allow us to assess the guidelines proposed in Chapter 3. In particular our design goals address distinct concerns:

User Domain functionality Provide a system that caters to architects' needs during a design review meeting. The system should take into account freedom of movement and the size of the display when designing the interface.

Pointing Develop devices and techniques that support the pointing variants that provide the best performance. If possible, the device is both designed to support multiple variants and naturally provide the user with the options to switch between them.

Multi-Modal Provide a multi-modal interface where different modalities provide natural alternatives to the pointing device. Along with pointing, gestures and speech are often used either as stand-alone controls or as part of multi-modal dialogues designed for large-scale displays.

Models for Ray-Pointing on Large-scale displays To propose a specialized interface for pointing, take into account both viewing angles and user position. The interface would favor the display space directly in front of the user and treat peripheral display space as secondary.

User distance Provide an interactive experience where users could move freely within a large space, without user distance affecting the choice of appropriate metaphor.

Without going into further details, we will now describe the system architecture and explain how it supports these goals.

4.1.3 System Architecture

Our application provides innovative multimodal interactions for 3D content visualization using a bus architecture as depicted by Figure 4.2. The system relies on two frameworks: AICI [Hur *et al.* , 2006], for visualization and tracking, and IMMIView, for interaction. AICI is responsible for the 3D rendering, based on [OpenSG, 2009], and was extended to support advanced lighting through High Dynamic Range rendering. Furthermore, input streams are managed with OpenTracker [Reitmayr & Schmalstieg, 2005] which provides multimodal information including marker tracking (body gestures), laser input, and traditional keyboard and mice.

Communications and input are routed through the IMMIView framework, built around an event-based bus that allows other modules (tracking, keyboard, graphics) to publish or subscribe to events. The choice of a publish-subscribe method-

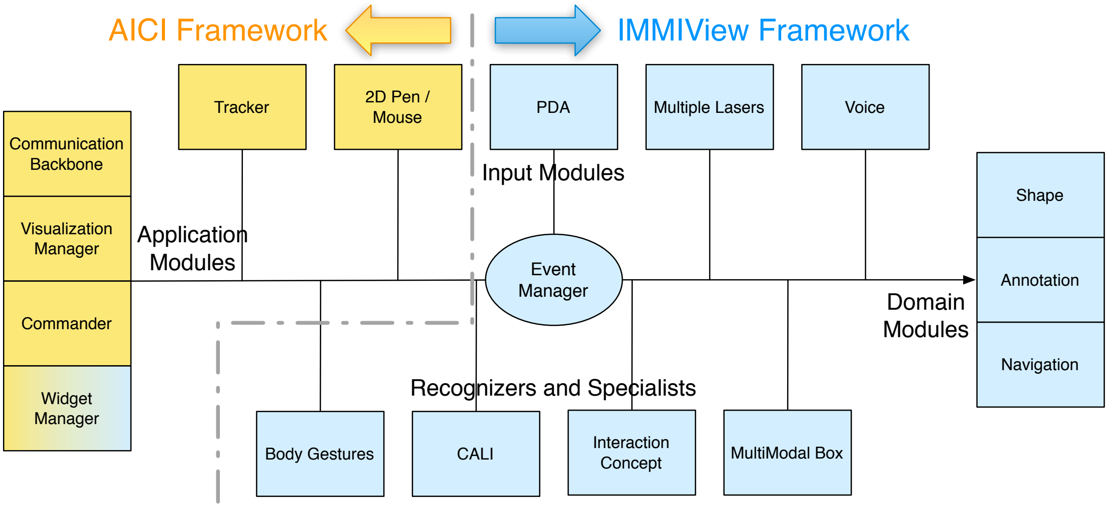


Figure 4.2: The IMMIView System Architecture

ology facilitates handling multimodal streams and provides extensibility to allow the inclusion of new modules and features without requiring any core changes, easing fusion of modalities. In IMMIView logic, the event manager registers the modules that are interested in a target event type, and once an event is triggered, all the interested parties are forward the event notification. IMMIView modules are organized into three different classes: publishers, consumers and converters.

Publishers are entities that do not require further information from the IMMIView system to update their status. Each publisher informs the event manager what type of events they produce. New information is published onto a waiting line of the event manager. This mechanism provides a high level of decoupling to allow object oriented prototyping that it is resistant to changes in modules. Modules such as the multi-user laser handler, data proxies from hand-held devices, and body tracking modules belong to this category, since they only input data to the event manager.

Consumers require information to change their state. To this end, they subscribe to callbacks for a particular event type. For example, the visualization module subscribes to *navigation type* events in order to change the camera parameters. Typical consumer modules include: annotation manager, shape creation and manipulation, and the widgets that comprise the menu interface.

Converters act as both consumers and publishers. They subscribe to multiple events such as laser input and speech commands and compose those commands to generate higher level events such as object selection or navigation actions. Modules included in this class are detailed in the next section.

User Interaction.

User interaction is supported by converter type modules that listen to events, process them and as a result, publish higher level events to the event manager. We have defined the following converter modules:

Body Gesture Recognizer. Analyzes tracking data obtained from real time marker based motion capture and publishes body gestures. To obtain such user data, we track the user's head and arms and send the information to the body gesture recognizer.

Cali Gesture Recognizer. Cali is a 2D symbol recognizer [Fonseca *et al.* , 2005]. It receives data from 2D input devices such as pen, mouse or lasers and generates events, once stroke gestures are recognized. For example, the main menu can be opened by drawing a triangle gesture.

Multimodal Box. This module provides support for multimodal dialogues via a rule-based inference mechanism. Using the information available on the event bus and a predefined grammar, the multimodal box is able to compose interaction events by combining mixed-input modalities such as body gestures with speech, or mixing pointing interaction with actions performed using mobile devices to create new annotations.

4.2 Input Devices

IMMIView offers several input devices to support interaction. From task analysis, we selected a laser pointer as the main interaction device and speech and gestures as secondary metaphors, thus providing a multi-modal interface. For example, a combination of laser and speech based interaction can activate menu options. Additionally, combining input surfaces is also supported for annotation. Thus, the user can insert multimedia content and notes using both handheld devices and laser pointer. Furthermore, the whole body can be used as an input device, allowing both to navigate and edit objects via body gestures, which can also be combined with speech to trigger complex actions.

4.2.1 Pointing Device

Following on the lessons learned in the previous chapter, distance does not seem to affect the interaction metaphor. Moreover, for targeting tasks, the pointing device should provide laser-style ray pointing. Additionally, to increase precision the device should be used in an arrow pointing style. Thus, for the device adopted, the way users carry out tasks should not be affected by distance (either by changing behavior or just working from a specific distance) and the device should support both *arrow* and *laser* stances, with the user switching between them according to the task at hand.

Given these requirements, we adopted laser pens for user interaction (hardware described in Section 2.7.1). The laser position is captured using computer vision techniques, instead of resorting to marker based tracking system. This choice enables both stances without additional recognition, as the capturing algorithm relies on the laser position on the large-scale display. We use an IR sensitive camera to reduce image noise and simplify laser detection. Once a frame is captured, it is filtered in order to identify high intensity pixel clusters, which allows us to determine the laser position on-screen. Afterwards, each laser position gets translated into cursor events. Figure 4.3 depicts the three main steps of the laser recognition algorithm. The algorithm provides an accurate position (4x4 pixel wide) and allows interaction up to eight meters. Apart from infra-red light, the laser also shows a narrow visible red light beam that shows the current position. Once a sketch has been started, the application draws a dotted line (stroke) to provide additional feedback. This implementation is not affected by distance (apart from hand jitter) and enables pointing from distances ranging from very close (arm reach) to further away, limited only by the laser light intensity.

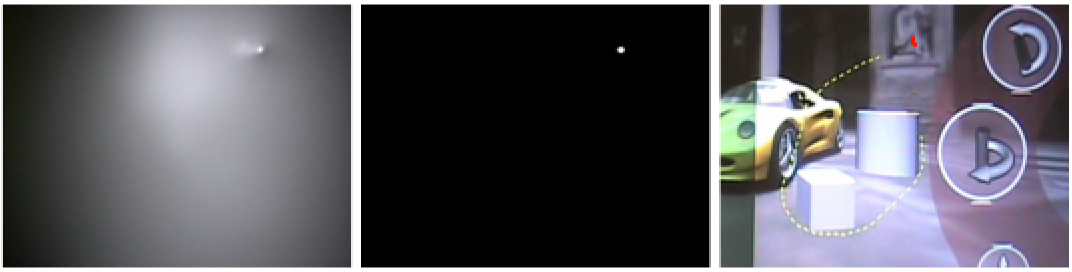


Figure 4.3: Laser detection algorithm steps: (left) acquisition, (middle) filtering and (right) stroke.

Disambiguating Laser Input

To support multiple users, the system needs to distinguish each laser signature. Because we use standard red laser pens, their light signature is similar and cannot be distinguished by computer vision algorithms alone. Using a Kalman filter, we are able to detect how many users are interacting and maintain their individual interaction state. The Kalman filter is a known method for stochastic estimation which combines deterministic models and statistical approaches in order to estimate the variable values of a linear system [Welch & Bishop, 2006]. In our system, we use this technique to estimate and predict possible laser positions (Figure 4.4 depicts this workflow). Because of screens size and camera resolution, our system integrates two cameras for laser detection. Cameras are calibrated according to the system and their position is known, thus we can translate between each camera coordinate system and a global position, in which we will be handing the next steps. Laser positions are sent to a single server, responsible for collecting the information of all cameras and matching the input to active strokes. Using the Kalman filter's predictive behavior, it is possible to match points of the same laser. In case of a point not being matched to any estimation, we assume that a new stroke was started and use the location coordinates as the stroke's first position. If there is a match between a point and its estimation, the corresponding stroke is updated with a new point and remains active. Thus, using simple laser pointers it is possible, without extra hardware, that input from two users using laser pens can be disambiguated and independently draw on the same surface.

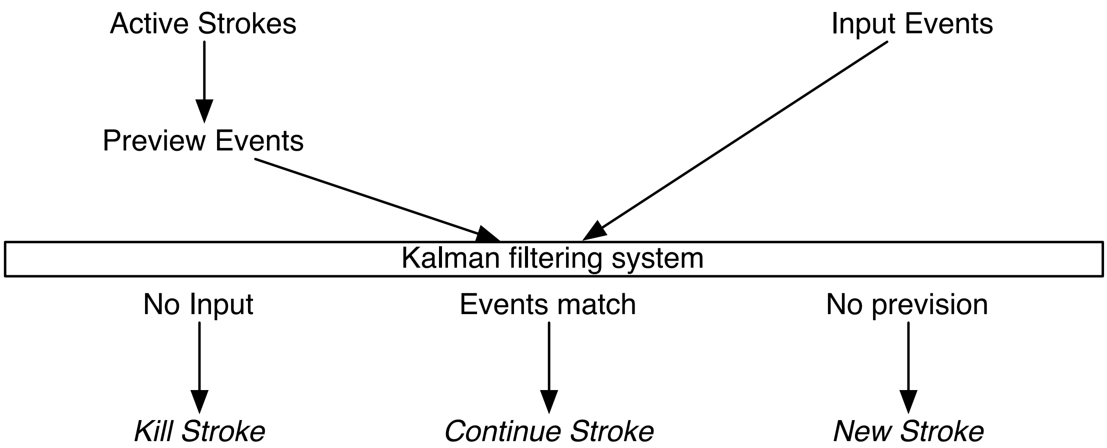


Figure 4.4: Stroke estimation workflow

4.2.2 Speech

Speech recognition uses Microsoft Speech SDK, trained for a number of commands to enhance laser interaction with direct commands such as *"Hide This"* and in combination with body gestures. Recognition rate was around 90% for British native speakers. The recognizer used was configured for british accent which posed problems for other accents, for example, a Scottish accent reduced the speech recognition rate down to around 75%.

4.2.3 Handheld Device

The post-it metaphor was largely mentioned whenever the annotations were discussed during user interviews. This metaphor is available on the GUI – by writing an annotation and dragging it to the desired location. To increase annotation drawing resolution, a handheld device was used to simulate physical post-it notes. Users can sketch on a handheld device, or choose a picture, and then using the laser point to the large-scale display position where they want an annotation to appear (see Figure 4.5).

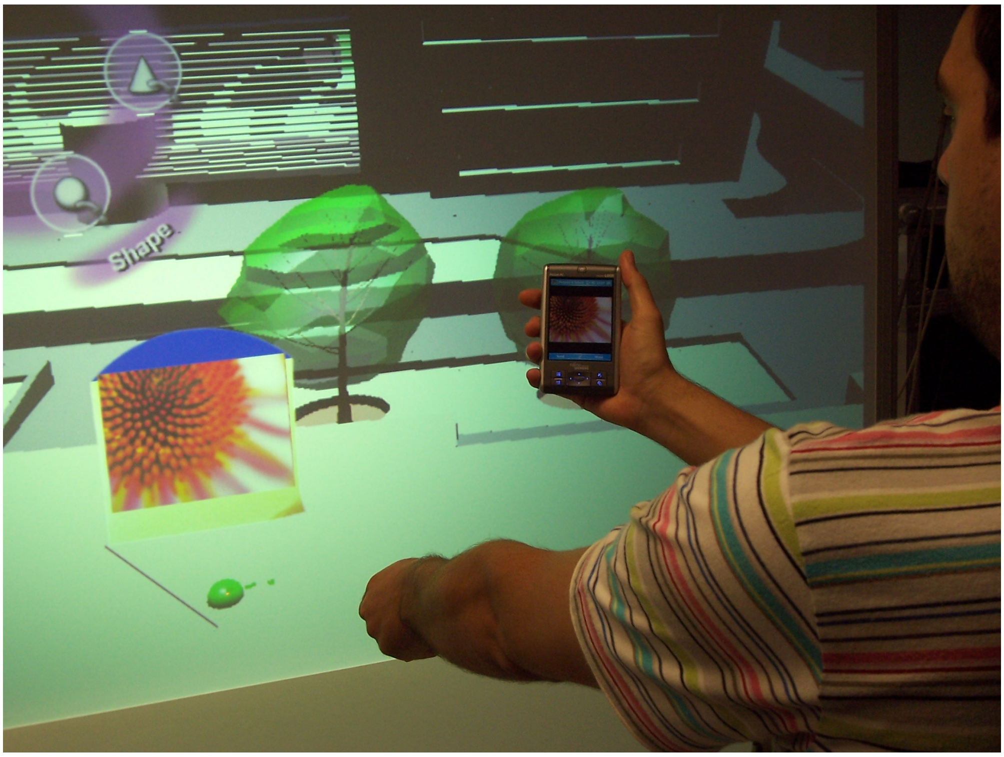


Figure 4.5: Handheld device metaphor example.

4.2.4 Body Tracking

This component resorts to IR cameras which track parts of the body using markers. For instance, using arm gestures users can navigate and edit objects. These different modes are activated using a speech command, after which the user can control the action using motions depicted on Figure 4.6. Our tracking setup uses four cameras with infra-red sources to capture reflective markers. To capture arm gestures, users wear markers on each wrist and shoulder. The information provided by each maker position allows us to compute the geometrical relationship between hands and shoulders. In this way, we are able to recognize different postures and the corresponding gestures.

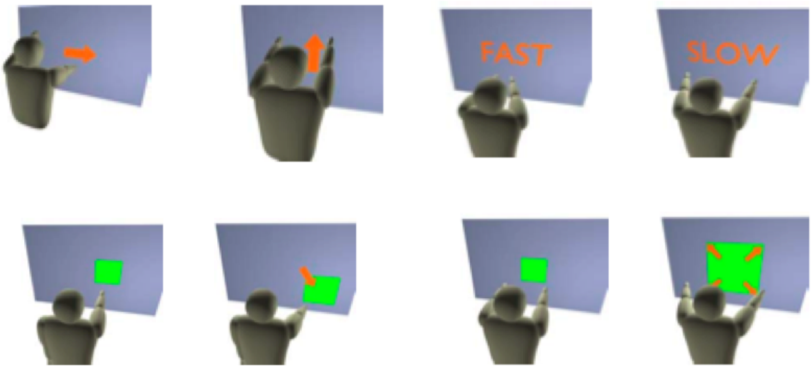


Figure 4.6: (Top row) Controlling fly mode. (Bottom row) Moving or scaling objects.

4.3 User Interface

Our approach uses laser pens to interact with the large-scale display, using strokes to activate interface elements, select objects or invoke menus. A stroke is a sequence of points that define a geometric shape such as lines, curves, circles, laces, as depicted on Figure 4.7. These shapes can represent any number of interaction functionalities, such as selecting objects inside the circle or deleting an object crossed by a line.

4.3.1 Activating interface elements using strokes

Clickable desktop interface widgets such as buttons and menus assume that users can *click* to activate them. However, our laser pointers have no extra hardware for this function. We devised a basic set of widgets that are activated

Figure 4.7: Stroke Examples. Left to Right, Top: closed curve, straight line, open curve. Bottom: Complex line, triangle, gesture

using strokes, instead of *clicks* (see Figure 4.8).

Buttons are replaced with gates which are activated when crossed by a stroke (see Figure 4.9). The first concept is the activation gate [Apitz & Guimbretière, 2005]. To select objects, users must surround the desired object using a closed stroke, instead of a selection box as is found in desktops (Figure 4.10 left). Any object that is inside a closed curve is considered selected (Figure 4.10 middle). To discard a selection the user creates a closed curve, with no objects inside (Figure 4.10 right).

Finally, desktop mice usually feature a second button that allows one to easily switch modalities – and often triggers context menus, when performed on a target object. As laser pointers do not posses this explicit physical mode switch, we opted for automatic context menus. Once an object is selected, a context menu is displayed. For global menus, because there is no object to select, a triangle gesture is required, as depicted in Figure 4.7. Furthermore, initial testing showed us that this gesture is rarely recognized by accident, since it includes three acute angles and therefore robust to recognition errors.

Figure 4.8: Text or icon based activation gate.

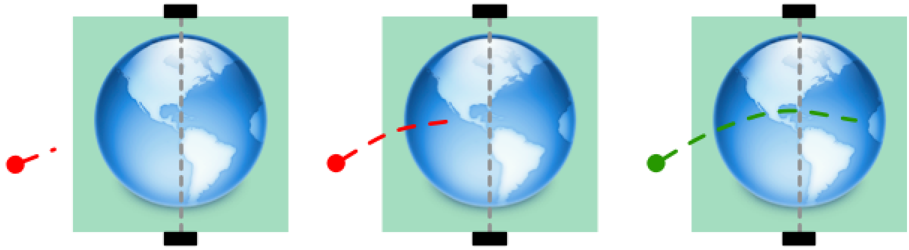


Figure 4.9: Example of gate activation: The gate is only activated in when the stroke (color dashed line) crosses the gate threshold (gray dashed line).

Figure 4.10: (left) Desktop selection, (middle) stroke selection, (right) example of empty selection

4.3.2 Menus

Designing menus for pointing devices without a button required us to re-invent menus to take advantage of strokes the calligraphic input. We opted for circular layout menus, similar to a torus shape (Figure 4.11). Feedback is provided both through iconic and textual cues. Their circular shape minimizes sketch motion and avoids occluding the selected object, since the menu options appear around the object, rather that on top of it.

Menus are labelled using captions. However, the background color enabled us to easily identify the scope of the menu. For example annotations, navigation, object creation or transformation and system configuration all have different background colors. Finally, to support multiple user interaction on collaborative scenarios, multiple menus can be opened, moved and controlled using the peripheral options located on the top right of each circular menu.

Annotation Menu

Annotations use a yellow canvas, evoking the post-it metaphor, placed in the central area of the *Annotation Menu* (Figure 4.12, top left). To place a note, the user sketches a path from the placing button to the desired 3D location (Fig-

Figure 4.11: Left: Main menu. Right: Notes Menu including free sketching area

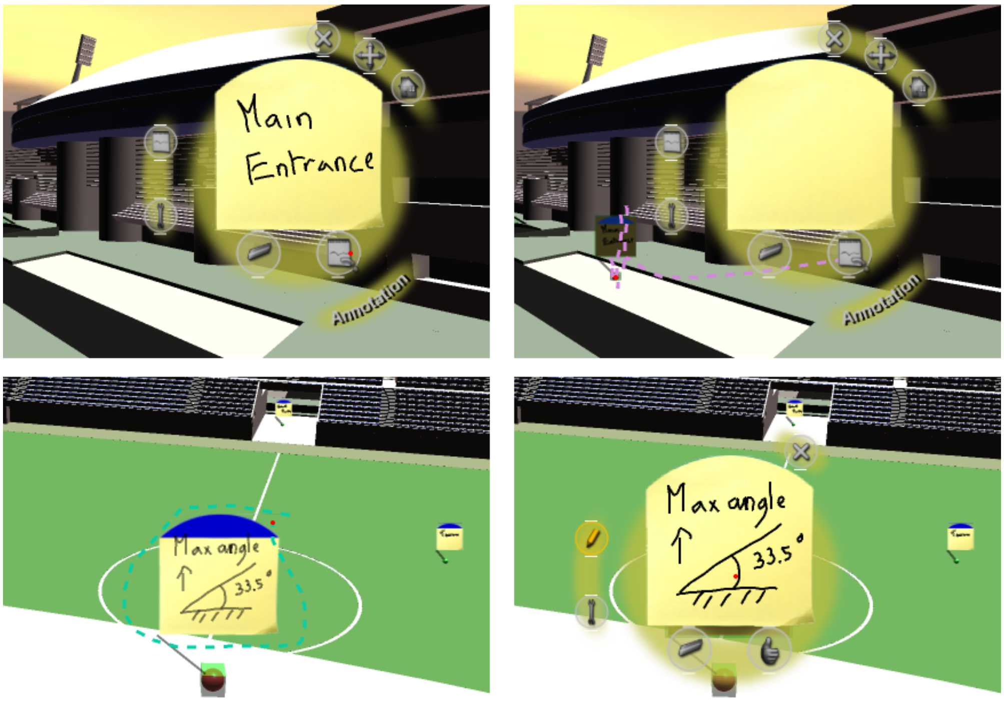


Figure 4.12: Creating and editing annotations using the IMMIView GUI

ure 4.12, top right). The annotation will be snapped automatically to an object on the scene. Notes are represented on the scene as floating post-its with anchors (Figure 4.12, bottom left). They can be edited, deleted or hidden by selecting them with a lasso. This action brings the *Annotation Menu* dedicated to the selected note (Figure 4.12, bottom right).

Navigation Menu

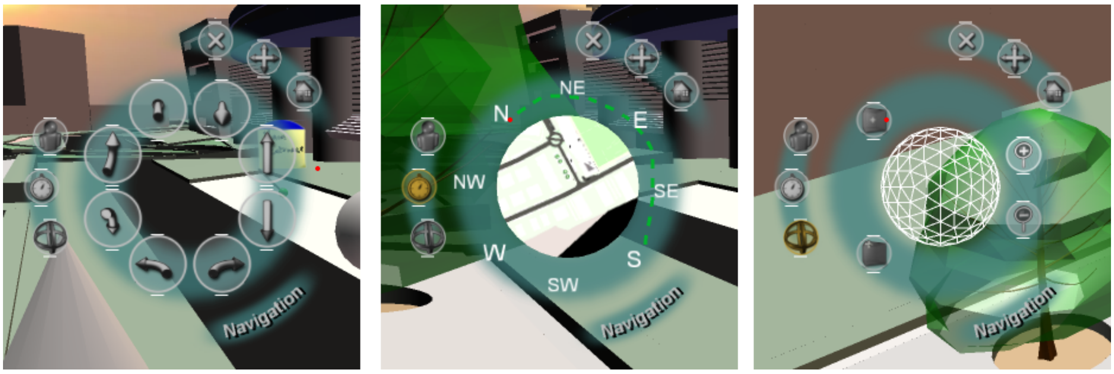


Figure 4.13: Three mode navigation menu: 1st Person, Mini-Map, Explore.

The *Navigation Menu* affords three different ways to explore a 3D scene (Figure 4.13): first-person, map-based and examine. First-person allows the user to issue direction commands and is more suitable to accurately adjust the user's position. The map-based mode provides a map and a compass. Users point directly to the map position where they want to navigate to and use the compass to control orientation. Finally, the examine mode allows users to rotate around a target object. Using a track-ball analogue, the user can control rotation and zoom of the scene centered on object.

Object editing Menu

Simple object edition functionality allows user to add simple shapes to the 3D scene via the *Shape Menu*. Spheres, cubes, cones, cylinders and planes can be created and placed anywhere by sketching a path from the menu icon to the target location. Moreover, these shapes can be deleted or geometrically transformed by selecting the object and accessing the corresponding option on the context menu.

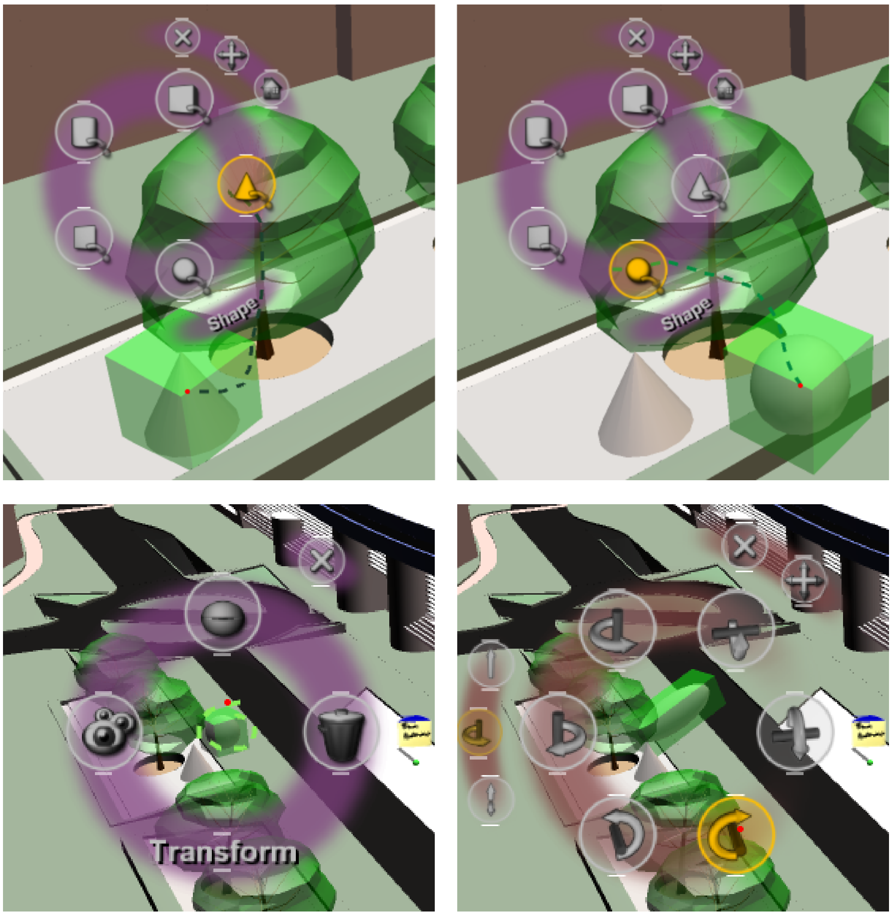


Figure 4.14: Creating simple shapes with menu and editing their attributes

4.4 User Tests

To validate our approach, the prototype was evaluated by expert users (architects). In this section we describe the user tests and discuss how the pointing interaction techniques affected user performance in this real world scenario.

4.4.1 Apparatus

The tests were conducted in a controlled environment. The large-scale display, depicted in Figure 4.15, is comprised of a 4x3 projector array with a total resolution of 4096x2304 pixels over a physical area measuring 4 x 2.25 square meters and located 30 cm above the floor plane. Subjects were given a starting position two meters orthogonal front, and facing, the center of the display. Participants were free to physically move in a 4x5 square meters interaction area, which allowed participants to select between interaction distances ranging from close (touching the display with the laser pen) to up to five meters from the screen (long distance pointing). Gestural input was captured in a 2x2 square meter area, due to marker tracking limitations. The area were body gestures were recognized was aligned with the display and located at one meter apart from the display plane.

Figure 4.15: Two users collaborating on a large-scale Display.

As described in Section 4.2, we implemented pointing with a laser pen, with an on/off button. Speech was provided using a microphone earpiece, and gestures were recognized using a OptiTrack maker tracking system. Subjects were tagged with reflective markers in wrists and shoulders and gestures were captured at a 60 hz rate. Subjects were instructed that they could remove the microphone and infra-red markers, if they felt discomfort or fatigue. Software ran on a Pentium 4 running Windows XP. Both image and input were refreshed at a rate above 50Hz.

4.4.2 Participants

The tests included 22 subjects, nine female and 13 male, ranging from 22 years up to 45 years old with an average of 33 years. Subjects were all expert domain users, architects whose job description include client presentations and design review meetings with fellow architects. All subjects, except two, had no previous experience with interactive large-scale displays. All of the subjects were comfortable with laser pens, and had previously used those in presentations. While subjects were aware of speech interfaces, none had experience with multimodal speech and pointing devices. Four users were familiar with gestural controllers such as Wiimote, but not within a large-scale display scenario.

4.4.3 Procedure and Design

Subjects were presented with a document that included a user manual explaining the interface, and a consent form. Before executing any task, subjects filled out a initial questionnaire to capture their profile. Afterwards, subjects were shown the interface by an HCI expert and experimented with the system for up to 15 minutes. They were then presented with the three input methods: pointing, speech, and gestures and instructed that they could opt for whatever method they felt was better suited for the task on hand. Users were not instructed regarding ideal interaction distance or optimal pointing stance.

When subjects felt comfortable with the system, evaluation tasks would begin. Each session included three single-user tasks and one multi-user collaborative task, to be executed with a HCI expert. Single user tasks were designed with different degrees of difficulty. The first one comprised two easy steps related to navigation, creating and editing annotations. The second one included four medium difficulty subtasks including navigation, creating, selecting and manipulating notes and 3D objects. Finally, the third task added seven more specific steps such as geometric transformations to 3D objects including scaling, rotation and translation. For the collaborative task, subjects were required to execute four medium subtasks. It was up to the subject to decide how they would split the actions between the two. To conclude, we asked subjects to revisit the third task and requested that they used a different interface (*e.g.,* if they used laser to navigate, then they should use gestures). Next, after conducting the tasks, users were debriefed and asked to rate the interaction techniques through a final questionnaire.

4.4.4 Measures and Analysis Methodology

The data collected during the user tests come from different sources, these include a usability questionnaire given to each user based on the standardized *ISONORM 9241 Part 10*, technical logs representing the laser position and movement during tasks, feedback from user comments, experts observation task notes and video analysis.

The standardized *ISONORM 9241 - Part 10* (Ergonomic requirements for office work with visual display terminals) provides requirements and recommendations related to the hardware, software and environment attributes that contribute to usability and the ergonomic principles underlying them. Through the usage of questionnaires, the goal was to obtain feedback from the users experience related to the following seven principals of this standard: (1) Suitability for the task, (2) Self descriptiveness, (3) Controllability, (4) Conformity with user expectations, (5) Error tolerance, (6) Suitability for individualization and (7) Suitability for learning. Users were asked to rate each question from 1 (the least favorable case) to 6 (the most favorable case). The results are presented in the Table 4.1.

4.4.5 Results

Except for error tolerance, all other principles are above the mean value of 3.5. Even though standard deviation values are relatively high, in general the system seemed suited to the tasks tested (Suitability for the task: 4.49, Conformity with user expectations: 4.27). Users found the system easy to understand (Self

	Average	*Std. Deviation*
Suitability for the task	4.49	0.79
Self Descriptiveness	4.56	0.63
Controllability	4.26	0.68
Conformity with User Expectations	4.27	0.83
Error Tolerance	3.32	0.89
Suitability for individualization	4.50	0.39
Suitability for Learning	4.62	0.51

Table 4.1: Results of ISONORM questionnaire on the scale 1 to 6.

descriptiveness: 4.56) and control (Controllability: 4.26). The controllability result suggests that users found the pointing metaphor an adequate solution for a large-scale interactive application. However, in the questionnaires, subjects ask for performance improvements to speech recognizer, gestures and laser pointer accuracy. Indeed, some subjects had a thick scottish accent lowering the speech recognition rate to 75%. Regarding laser accuracy, the laser pen projects a red dot that users perceive as the cursor. The algorithm is accurate up to four pixels. However users expected one pixel accuracy. This can be easily achieved by adding infra-red cameras to the setup and fine-tuning the calibration algorithm. On a positive note, users found that the different modalities offered multiple solutions. This is reflected in the "Suitability for individualization" result of 4.50. Moreover, users found it easy to learn the system (4.62), although they required some assistance remembering speech commands and gestures associated with geometric transformations.

4.4.6 Users Task Performance

We collected quantitative data from tests related to navigation and annotations to assess the performance of each multi-modal metaphor. Table 4.2 provides the error rate and the time per task when using three different modality combinations.

	Error/Usage	*Time/Usage*
Pointing/Menus	0.27	1:37
Speech/Menus	1.80	0:59
Gestures/Speech	0.89	1:50

Table 4.2: Navigation performance data by modality.

We found that error rates increased when subjects exercised the speech modality, which can be explained by the failures of the speech recognizer system to identify commands. On the other hand, speech interaction shows a significant speed up when opening menus; when the speech command was correctly recognized, the time to activate the corresponding commands was very short.

Compared to pointing, subjects needed slightly more time to accomplish the same task with gestures. Moreover, gestures exhibited a higher error rate. This suggests that the users spent more cognitive resources to adjust their position and orientation when using gestures for navigation. The results show that there were less errors using the pointing modality. This might be because users have more experience with pointing devices.

	Error/Usage	*Time/Usage*
Pointing/Menus	0.00	00:30
Pointing/Speech	0.84	0:51

Table 4.3: Annotation performance data by modes.

For annotation tasks, Table 4.3 provides the error rate and the time required when using each multimodal combination. Although the speech modality was chosen primarily by all users (see Table 4.5), the error rate and time required were higher than pointing. In fact, pointing exhibited no errors during our tests. The errors incurred when using speech commands were due faulty recognitions.

4.4.7 Multi-modal Preferences for user tasks

Throughout the experiment, subjects could interact with the system using different modalities and were free to select the modality they felt was better suited to the task. Next, we present user preferences for navigation, annotations and geometric transformations.

Navigation

Table 4.4 shows modalities sorted by user preference. From it, we conclude that participants used different combinations of modalities to perform the same task. The first choice is balanced among the three different combinations, which illustrates the system flexibility in accommodating user preferences. When users chose a second combination of modalities (40% of users chose a second option), a slight majority (54.55%) opted for the pointing modality.

	1st Choice	2nd Choice	3rd Choice
Pointing/Menus	32.73%	54.55%	100.00%
Speech/Menus	27.27%	27.27%	0.00%
Gestures/Speech	40.00%	18.18%	0.00%
% from previous	-	40.00%	4.55%

Table 4.4: User modality preference in navigation tasks (in percentage).

Annotations

Table 4.5 shows modalities sorted by user preference, in annotation tasks.

	1st Choice	2nd Choice
Pointing/Menus	0.00%	100.00%
Pointing/Speech	100.00%	0.00%
% from previous	-	24.49%

Table 4.5: User modality preference in annotation tasks (in percentage).

In spite of of speech recognizer mistakes, the speech modality was unanimously preferred as the first choice to perform annotation tasks. As a second choice, 24% of the subjects selected pointing as a fallback modality, in case the speech recognizer failed to understand a command. These results indicate that subjects welcome secondary modalities, in addition to pointing, for creation and edition of annotations.

Object Editing

Users were able to create and edit objects via menus or speech activated gestures. Table 4.6 shows modalities sorted by user preference in object related tasks. Similar to navigation, subjects preferred gestures over pointing (88.46%). When inquired about this choice, they explained that gestures felt more natural and direct, albeit slightly less precise.

	1st Choice	2nd Choice
Pointing/Menus	11.54%	100.00%
Speech/Gestures	88.46%	0.00%
% from previous	-	30.77%

Table 4.6: Choice of modality combinations when editing objects.

4.5 Observations

Questionnaire results suggest that users are comfortable with pointing as an interaction metaphor and like a multi-modal interface which provides redundant techniques for the same tasks. Indeed, despite admitting a strong preference for the first and second modalities, users switched between modalities accordingly to the task at hand. Throughout the remainder of this section we discuss our observations and draw conclusions based on the test results.

Pointing

Although not being the primary choice for all tasks, pointing showed satisfactory results as it yields less errors than the other modalities tested. From responses to questionnaires, we understand that pointing is a suitable modality to operate large-scale displays, since it fits better within the expectations of architects for a presentation/meeting scenario, as pointing devices are familiar tools in such contexts.

Error tolerance is an important metric for real world applications. When we analyze our results from this perspective we realize that further work is required to improve the system robustness. In particular, pointing shows promising results, since it exhibits virtually no errors and provides a high accuracy in both navigation and annotation tasks. Moreover, pointing suited users in need of a calligraphic tool for collaboratively annotating content on 3D scenes displayed on a large-scale display. On the other hand, speech commands are still a source of errors, due to a recognition rate of the recognizer. However, spoken commands are valuable shortcuts for navigation tasks, as tests revealed.

For navigation and object editing tasks, users selected other modalities than pointing as first choices. However, users still manifested appreciation for pointing, as a relevant fallback metaphor. Indeed, during tests we observed that subjects used both modalities to carry out tasks. For example, several subjects used gestures, to *fly* to a position close to the target, and switched to pointing to fine-tune both position and camera orientation. When pointing was chosen as the primary interaction technique, as in annotation tasks, the majority of users only expressed a need for an alternative technique if the first would fail. This clearly denotes how comfortable users felt with pointing devices.

User Preference for Gestures over Pointing

For navigation and object editing, it is clear that results show a tendency towards more direct metaphors, such as gestures accompanied by speech commands, instead of pointing. This is somewhat congruent with the benefits of direct versus indirect control [Schmidt *et al.* , 2009], as users believe that gestures seem more natural if they are executed directly on 3D objects – e.g., a user would prefer to scale an object by issuing a spoken speech command and controlling the intensity with a continuous two-arm gesture, rather than pointing to activate a menu option.

User Position, Control Type, and Parallax

Through the laser pen, our implementation provided a simple control device to support interaction across multiple distances. Based on the results from Chapter 3, the pointing device should support multiple stances. Indeed, during user tests, we observed three main stances that corroborate these results. We were able to observe *laser* stance during object selection, *arrow* during gate activation, and users holding the laser pen as a whiteboard marker (similar to the *grab* metaphor) while writing annotations.

Pointing stances were strongly linked to changes in task and, in the case of annotations, user position. That is, when we asked users to navigate to a target position and create annotations, they would navigate using a *laser* stance from their current position, approach the display, switch to a whiteboard stance, and create the annotation. When asked why they changed their behavior, users claim they were not comfortable either drawing or writing annotations at a distance. This seems to indicate that users naturally adopt the most advantageous mode afforded by the device, suggesting that for annotation tasks there is an optimal interaction distance (close to the display).

Taking into account angles when modeling targeting

The interface design supports menus that open and can manually be moved to any position. When performing tasks, the majority of subjects opened or moved menus to a position in front of them. Moreover, subjects would only move the menu to peripheral positions when they required an obstructed view (for example when selecting multiple objects). Whenever a menu was no longed required, most users closed it, instead of moving it to another position. During collaborative review, this behavior can be attributed to a sense of possession. However during single user tasks, such is not the case and therefore we conclude that users were aware that interacting with objects at an angle was less convenient than interacting with objects placed in front of them.

Summary

In this chapter we apply our lessons learnt and guidelines of pointing, as an interaction technique, to a collaborative design review application in a multimodal framework, and include secondary modalities, such as speech and gestures. We support passive, *clickless* interaction devices such as laser pointers, to make the setup simple and accessible from the non-technical user side. Based on the guidelines elaborated in chapter 3, we developed a graphical user interface centered around pointing as an interaction enabler for large scale displays. To this end, the whole graphical user interface framework resorted to strokes as an activation method instead of clicks. We strove to make all modalities almost equally accessible to users so they could write annotations, select objects, draw and open menus using either pointing, speech, or gestures. We conducted user studies to observe how architects would use pointing to execute design review tasks and we were able to assess how our guidelines could positively affect the architecture of a less contrived application that pure laboratory settings would allow.

5
Conclusions

Our work focused on interaction with large-scale displays, in particular in understanding the effect of user position on ray-pointing techniques for large-scale displays. Towards that end, we conducted two studies. In the first study, we tested three metaphors at three different distances and found that distance to the display does not seem to directly affect interaction. However, we found that the metaphor used has a correlation with the interaction performance. Having obtained these results, we conducted a second user study to understand what factors affect the interaction metaphor. Moreover, we now focused on ray-pointing, the technique that showed the best promise from the first study. Also, given the initial results that distance does not seem to affect interaction, instead of testing multiple distances we position users to the left of a large-scale display. In this second study, our goal was to better understand how metaphors are affected and to see if target position, as related to user position, affects interaction. We found that control type, one of the factors tested, affects targeting and that target position affects tracing tasks.

We followed the user studies with an application of pointing, as an interaction technique, for a design review application. The prototype included pointing as a the primary modality and secondary modalities such as speech and gestures in a multi-modal setting. Based on the guidelines presented in Chapter 3, we developed a graphical user interface based on pointing interaction for large-scale displays. The graphical user interface used *strokes*, instead of *clicks*, which made the prototype easier to use with passive pointing devices by allowing users to select, write, draw and open menus using pointing, speech, or gestures. Further user studies allowed us to observe how architects use pointing to perform design review tasks, and we were able to observe how the guidelines would affect a real-word application. The experimental results gathered provided us with valuable feedback regarding the validity of our approach. From these results we were able not only to confirm the value of our contributions, but also to identify the limitations of our approach. In this section we will present and discuss both.

5.1 Contributions

Human factors research that focuses on large-scale displays is still in its infancy. We contribute to the community by presenting user studies on large-scale display interaction that add to the knowledge base and by evaluating that contribution on a real-world scenario that took into consideration the previous findings. This allowed us to make educated choices when adopting an input device, devising interaction metaphors, and designing the user interface. This will provide practitioners with theoretical contributions that state what factors should be taken into account when designing interfaces for large-scale displays and practical examples of how those lessons can be applied.

5.2 Limitations

The narrow focus of a Phd dissertation introduced natural limitations. The work presented aims at a single, very specific objective. Without affecting the validity of the research presented, it is important not only to identify our contributions but also to frame the work presented and identify some of its limitations.

Our research provided more depth than breath. Although we set out to study interaction on large scale displays, our results hinge on vertical large-scale displays. Although there will be some similarities, nothing can be said about the applicability of our results to tabletops or even CAVE displays.

Another limitation is that, although pointing is a popular interaction technique, other metaphors exist and should be examined from a human factors point of view. It remains to be seen whether the human factors tested here apply to other metaphors such as gestures or handheld device interaction.

We tested distances according to proxemics spaces and our results show that distance does not affect interaction. However, we have no proof that the results still hold, if users interact at distances further away than the ones tested. However, based on the literate surveyed [Greenberg *et al.* , 2011; Myers *et al.* , 2002; Vogel & Balakrishnan, 2004], this is not a likely scenario and rarely will users interact, using pointing devices, from distances farther that the ones tested.

Although multiple users are supported by the real-world application, our studies did not take into account variables that multiple users introduce. Indeed, parameters such as occlusion, personal space versus public spaces, ownership of information, to name a few, will surely affect the way people interact and how information is displayed. Although we do not foresee that our results become invalid in multiple user scenarios, these bring forth a complexity that is beyond the scope of this dissertation.

5.3 Other work produced

During the execution of the body of work presented, other works were also developed and published. For various reasons, they do not fit the flow of this document but are an important part of the process that concluded with this document. Therefore, we present a brief abstract of three main contributions that were not detailed here.

5.3.1 MAXIMUS project

MAXMUS is a FP7 European research project which aims to improve the design review process within automotive and architectural design through dramatically improved rendering and interaction technologies[1]. Over three years (between 2008 and 2011) the consortium, lead by the Fraunhofer institute, developed a system that compliments the way designers work and collaborate,

[1]http://maximus-fp7.eu/

by taking advantage of high dynamic range based rendering and novel interaction techniques. I was main technical researcher and advised the project during the three years of execution in which we developed an interactive environment for creative industries. In particular, we constructed a multi-touch tabletop (A0 size, not available on market), developed a prototype that integrated both vertical and horizontal surfaces, and evaluated a 3D interaction device to be utilized during design meetings. At the time of the writing a publication was submitted to CHI'11 regarding these advancements.

5.3.2 The continuous interaction space

During my internship in Calgary, in collaboration with Nicolai Marquadt, the *Continuous interaction space* was researched. This project looks at two modalities: 1) direct touch and multi-touch (by hand and by tangibles) directly on the surface, and 2) hand gestures above the surface, but also at the rich interaction space between them. The idea is that many interaction techniques can be developed that go beyond these two modalities, where they can leverage the space between them. That is, we believe that the underlying system should treat the space on and above the surface as a continuum, where a person can use touch, gestures, and tangibles anywhere in the space and naturally move between them. The project resulted in the following publication:

The continuous interaction space: Interaction techniques unifying touch and gesture on and above a digital surface. N. Marquardt, **R. Jota**, S. Greenberg, and J. A. Jorge. In Proceedings of the 13th IFIP INTERACT Conference, Lisbon, Portugal, September 2011.

5.3.3 StereoBlocks

The internship at Microsoft Research produced a tabletop side project. The immediate availability of depth cameras - not available until 2010 - allows us to capture object information without the need to resort to simplified object representations. We combined a depth camera with a 3D projection to implement a modeling application. By treating the camera information in an analog fashion, instead of treating it as marker information common in VR applications, we are able to digitize all the details of the real world and re-project objects side-by-side with real objects. We tried to re-imagine if architects had the capability to construct simple digital models out existing objects lying around, with no trackers, by simply adding their representation to a virtual scene. The following publication was the result of such project:

Constructing virtual 3D models with physical building blocks. **Ricardo Jota** and Hrvoje Benko. In Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems (CHI EA '11). ACM, New York, NY, USA, 2173-2178.

5.4 Future Work

The limitations described are good indicators of possible next steps. Although it is not within the present research to find an ultimate solution, we believe that our approach can evolve in those directions. The next paragraphs summarize some of the possible paths for future work.

5.4.1 Distance

We would like to expand the distances tested, both closer to and farther away from the screen. In particular, studying the distance that blends pointing to touch interfaces seems particularly relevant given the recent interest in multi-touch devices. On the other hand, even though our results conclude that distance does not seem to affect interaction, we would like to understand at what distance does pointing stop to be an effective interaction technique.

Other Large-scale Displays Tabletops, in particular, are very in popular within the interaction community. Therefore, we could argue that the concepts of distance, or ray-pointing as a technique, are relevant for horizontal surfaces [Parker *et al.* , 2005]. Moreover, the physical affordances of vertical surfaces are very different from those of horizontal surfaces. For example, in horizontal surfaces users can position tangibles, etc. This might indicate that the lessons learned on our study do not totally transpose to horizontal surfaces, thus making future studies even more important.

Non-pointing Metaphors

Pointing is one technique amongst many. We have no information on how gestures are affected by field of view, for example. Extending this work to another class of interaction metaphors might present opportunities for interesting findings.

Multiple Users

Because of their size, large-scale displays are often used in collaborative work. Indeed, our real world scenario supported multiple users in a collaborative setting. This path was not exhausted with this dissertation and further studies are surely warranted. Pointing is a natural metaphor for human-to-human communication and therefore a good candidate for collaborative work studies. One example is how people might use pointing to agree on how to execute a task, such as pin-point a target location on a map visualized on a large-scale display.

Applying lessons learned

Large-scale displays are exiting the academic labs into the real world. Indeed, they see growing use in applications such as targeted interactive advertisement, control rooms, indoor navigation for public spaces, gaming consoles, and others. Given the wide-spread use of large-scale displays, and new technologies that enable pointing without complex gadgets or even hands free methods[2], it would be interesting to conduct a field test for an extended period of time in a non-controlled, public environment, with almost no academic supervision, and compare the performance of pointing with user tests conducted in a controlled environment. With the introduction of depth sensing as an input device, and could possibly target thousands of users and provide a exciting avenue for further research.

[2] As of 2011, depth cameras are only recently available as a commodity technology

References

Andersen, P. B. 1990. *A theory of computer semiotics: semiotic approaches to construction and assessment of computer systems*. New York, NY, USA: Cambridge University Press.

Apitz, Georg, & Guimbretière, François. 2005. CrossY: a crossing-based drawing application. *Pages 930–930 of: ACM SIGGRAPH 2005 Papers*. SIGGRAPH '05. New York, NY, USA: ACM.

Argelaguet, Ferran, & Andujar, Carlos. 2009. Visual feedback techniques for virtual pointing on stereoscopic displays. *Pages 163–170 of: Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*. VRST '09. New York, NY, USA: ACM.

Ball, Robert, North, Chris, & Bowman, Doug A. 2007. Move to improve: promoting physical navigation to increase user performance with large displays. *Pages 191–200 of: CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM Press.

Baudisch, Patrick, Sinclair, Mike, & Wilson, Andrew. 2006. Soap: a pointing device that works in mid-air. *Pages 43–46 of: Proceedings of the 19th annual ACM symposium on User interface software and technology*. UIST '06. New York, NY, USA: ACM.

Bodker, Susanne. 1990. *Through the Interface: A Human Activity Approach to User Interface Design*. Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc.

Bolt, Richard A. 1980. Put-that-there: Voice and gesture at the graphics interface. *Pages 262–270 of: SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press.

Bomsdorf, Birgit, & Szwillus, Gerd. 1999. Tool support for task-based user interface design - A CHI'99 workshop. *SIGCHI Bull.*, **31**(4), 27–29.

Bowman, Doug A., & Hodges, Larry F. 1997. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. *Pages 35–ff. of: Proceedings of the 1997 symposium on Interactive 3D graphics*. I3D '97. New York, NY, USA: ACM.

Bowman, Doug A., Johnson, Donald B., & Hodges, Larry F. 2001. Testbed Evaluation of Virtual Environment Interaction Techniques. *Presence: Teleoper. Virtual Environ.*, **10**(February), 75–95.

Buxton, William, Fitzmaurice, George, Balakrishnan, Ravin, & Kurtenbach, Gordon. 2000. Large Displays in Automotive Design. *IEEE Computer Graphics and Applications*, **20**, 68–75.

Byrne, M. 2003. *Cognitive Architecture*.

Cao, Xiang, & Balakrishnan, Ravin. 2003. VisionWand: Interaction Techniques for Large Displays using a Passive Wand TRacked in 3D. *UIST*.

Card, Stuart K., Newell, Allen, & Moran, Thomas P. 1983. *The Psychology of Human-Computer Interaction*. Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc.

Cavens, Duncan, Vogt, Florian, Fels, Sidney, & Meitner, Michael. 2002. Interacting with the big screen: pointers to ponder. *Pages 678–679 of: CHI '02 extended abstracts on Human factors in computing systems*. CHI EA '02. New York, NY, USA: ACM.

Cockburn, A., Quinn, P., Gutwin, C., Ramos, G., & Looser, J. 2011. Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback. *International Journal of Human-Computer Studies*, **69**(6), 401 – 414.

Cohé, Aurélie, Dècle, Fabrice, & Hachet, Martin. 2011. tBox: a 3d transformation widget designed for touch-screens. *Pages 3005–3008 of: Proceedings of the 2011 annual conference on Human factors in computing systems*. CHI '11. New York, NY, USA: ACM.

Davis, J., & Chen, X. 2002. LumiPoint: Multi-User Laser-Based Interaction on Large Tiled Displays. *Displays*, **23**(5).

de la Hamette, P., Lukowicz, P., G.Tröster, & Svoboda, T. 2002. FingerMouse: A Wearable Hand Tracking System. *UBICOMP*.

de la Rivière, Jean-Baptiste, & Guitton, Pascal. 2003. Hand posture recognition in large display VR environments. *In: Proceedings of the 5th International Gesture Workshop*. Springer.

de la Rivière, Jean-Baptiste, & Guitton, Pascal. 2005. Image-based analysis for model-based tracking. *Mirage*, March.

Descartes, René. 1641. *Meditationes de prima philosophia, in qua Dei existentia et animæ immortalitas demonstratur*. René Descartes.

Dietz, Paul, & Leigh, Darren. 2001. DiamondTouch: a multi-user touch technology. *Pages 219–226 of: UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM Press.

DMX. 2011. *webpage: http://dmx.sourceforge.net/*.

Fonseca, Manuel J., Pimentel, César, & Jorge, Joaquim A. 2002 (Mar.). CALI: An Online Scribble Recognizer for Calligraphic Interfaces. *Pages 51–58 of: Proceedings of the 2002 AAAI Spring Symposium - Sketch Understanding*.

Fonseca, Manuel J., Ferreira, Alfredo, & Jorge, Joaquim A. 2005 (August). Generic Shape Classification for Retrieval. *In: Proceedings of the 6th. IAPR International Workshop on Graphics Recognition(GREC 2005)*.

Fu, Chi-Wing, Goh, Wooi-Boon, & Ng, Junxiang Allen. 2010. Multi-touch techniques for exploring large-scale 3D astrophysical simulations. *Pages 2213–2222 of: Proceedings of the 28th international conference on Human factors in computing systems*. CHI '10. New York, NY, USA: ACM.

Greenberg, Saul, Marquardt, Nicolai, Ballendat, Till, Diaz-Marino, Rob, & Wang, Miaosen. 2011. Proxemic interactions: the new ubicomp? *interactions*, **18**(January), 42–50.

Grossman, Tovi, & Balakrishnan, Ravin. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *Pages 281–290 of: Proceedings of the SIGCHI conference on Human factors in computing systems*. CHI '05. New York, NY, USA: ACM.

Grubert, Jens, Hanckock, Mark, Carpendale, Sheelagh, Tse, Edward, & Isenberg, Tobias. 2007 (October). *Interacting with Stroke-Based Rendering on a Wall Display*. Tech. rept. 2007-882-34. Department of Computer Science, University of Calgary, Canada.

Hall, Edward T. 1990. *The Hidden Dimension*. Anchor.

Han, Jefferson Y. 2005. Low-cost multi-touch sensing through frustrated total internal reflection. *Pages 115–118 of: UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM Press.

Harrison, Chris, & Dey, Anind K. 2008. Lean and zoom: proximity-aware user interface and content magnification. *Pages 507–510 of: CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM.

Heidegger, Martin. 1979. *Sein und Zeit*. Tbingen: Niemeyer.

Hill, A., & Johnson, A. 2008. Withindows: A Framework for Transitional Desktop and Immersive User Interfaces. *3D User Interfaces, 2008 IEEE Symposium on*, March, 3–10.

Hur, Hyejung, Fleisch, Timo, Kim, Taek-Bong, & On, Giwon. 2006. AICI-Advanced Immersive Collaborative Interaction Framework. *Pages 20–25 of: Proceedings of the SCCE CAD/CAM Conference : Book of Abstracts & Proceedings CD-ROM.* Pyungchang, Korea: Society of CAD/CAM Engineers (SCCE).

Isenberg, Petra. 2009 (December). *Collaborative Information Visualization in Co-located Environments*. PhD Thesis, University of Calgary, Calgary, AB, Canada.

ISONorm. 2002. *ISO. Iso 9241-9:2000(e): Requirements for non-keyboard input ISO.*

Jiang, Hao, Ofek, Eyal, Moraveji, Neema, & Shi, Yuanchun. 2006. Direct Pointer: Direct Manipulation for Large-Display Interaction using Handheld Cameras. *CHI*.

Jordà, Sergi, Geiger, Günter, Alonso, Marcos, & Kaltenbrunner, Martin. 2007. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. *Pages 139–146 of: Proceedings of the 1st international conference on Tangible and embedded interaction*. TEI '07. New York, NY, USA: ACM.

Jota, Ricardo, Ferreira, Alfredo, Cerejo, Mariana, Santos, José, Fonseca, Manuel J., & Jorge, Joaquim A. 2006. Recognizing Hand Gestures with CALI. *SIACG*, July, 187–193.

Kela, Juha, Korpipää;, Panu, Mäntyjärvi, Jani, Kallio, Sanna, Savino, Giuseppe, Jozzo, Luca, & Marca, Di. 2006. Accelerometer-based gesture control for a design environment. *Personal Ubiquitous Comput.*, **10**(5), 285–299.

Kim, Hyosun, & Fellner, Dieter W. 2004. Interaction with Hand Gesture for a Back-Projection Wall. *CGI*, 395–402.

Kobayashi, Masatomo, & Igarashi, Takeo. 2008. Ninja cursors: using multiple cursors to assist target acquisition on large screens. *Pages 949–958 of: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. CHI '08. New York, NY, USA: ACM.

Kondraske, G. V. 1994. An Angular Motion Fitts' Law For Human Performance Modeling And Prediction. *Proc. Engineering in Medicine and Biology Society*, 307.

König, Werner A., Gerken, Jens, Dierdorf, Stefan, & Reiterer, Harald. 2009. Adaptive pointing: implicit gain adaptation for absolute pointing devices. *Pages 4171–4176 of: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*. CHI EA '09. New York, NY, USA: ACM.

Kopper, Regis, Bowman, Doug A., Silva, Mara G., & McMahan, Ryan P. 2010. A human motor behavior model for distant pointing tasks. *Int. J. Hum.-Comput. Stud.*, **68**(October), 603–615.

Lapointe, Jean-François, & Godin, Guy. 2005. On-Screen Laser Spot Detection for Large Display Interaction. *HAVE*.

Lawson, Ed, & Duric, Zoran. 2006. Using Deficits of convexity to recognize hand gestures from silhouettes. *VISAPP*, February.

Lee, Sangyoon, Seo, Jinseok, Kim, Gerard Jounghyun, & mo Park, Chan. 2003. Evaluation of pointing techniques for ray casting selection in virtual environments. *In: In Third International Conference on Virtual Reality and Its Application in Industry*.

Lee, SK, Buxton, William, & Smith, K. C. 1985. A multi-touch three dimensional touch-sensitive tablet. *Pages 21–25 of: Proceedings of the SIGCHI conference on Human factors in computing systems*. CHI '85. New York, NY, USA: ACM.

Leibe, Bastian, Starner, Thad, Ribarsky, Willian, Wartell, Zachary, Krum, David, Singletary, Brad, & Hodges, Larry. 2000. The Perceptive Workbench: Toward Spontaneous and Natural Interaction in Semi Immersive Virtual Environments. *IEEE Virtual Reality*.

Leubner, Chirstian, Brockmann, Chirstian, & Muller, Heinrich. 2001. Computer-vision-based human-computer interaction with a back projectioon wall using arm gestures. *Euromicro*, **27**(September), 308–314.

Lien, Cheng-Chang, & Huang, Chung-Lin. 1998. Model-based articulated hand motion tracking for gesture recognition. *Image and Vision Computing*, **Volume 16**(Number 2), 121–134.

Limbourg, Quentin, & Vanderdonckt, Jean. 2003. *Comparing Task Models for User Interface Design*.

MacKenzie, I. Scott, & Jusoh, Shaidah. 2001. An Evaluation of Two Input Devices for Remote Pointing. *Pages 235–250 of: Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction*. EHCI '01. London, UK: Springer-Verlag.

Malik, Shahzad, & Laszlo, Joe. 2004. Visual touchpad: a two-handed gestural input device. *Pages 289–296 of: ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*. New York, NY, USA: ACM Press.

Malik, Shahzad, Ranjan, Abhishek, & Balakrishnan, Ravin. 2005a. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. *Pages 43–52 of: UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM Press.

Malik, Shahzad, Ranjan, Abhishek, & Balakrishnan, Ravin. 2005b. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. *Pages 43–52 of: UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM.

Merleau-ponty, Maurice. 1976. *Phenomenologie de la perception*. French & European Pubns.

Microsoft. 2010. *Kinect webpage: http://www.xbox.com/en-US/kinect*.

Morris, M. R., Wobbrock, J. O., & Wilson, A. D. 2010. Understanding users preferences for surface gestures. *Pages 261–268 of: In Proceedings of Graphics*

Interface 2010. Ottawa,Ontario, Canada: Canadian Information Processing Society, Toronto, Ont., Canada.

Myers, Brad A., Peck, Choon Hong, Nichols, Jeffrey, Kong, Dave, & Miller, Robert. 2001. Interacting at a Distance Using Semantic Snarfing. *Pages 305–314 of: Proceedings of the 3rd international conference on Ubiquitous Computing*. UbiComp '01. London, UK, UK: Springer-Verlag.

Myers, Brad A., Bhatnagar, Rishi, Nichols, Jeffrey, Peck, Choon Hong, Kong, Dave, Miller, Robert, & Long, A. Chris. 2002. Interacting at a distance: measuring the performance of laser pointers and other devices. *Pages 33–40 of: Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*. CHI '02. New York, NY, USA: ACM.

Nacenta, Miguel A., Sallam, Samer, Champoux, Bernard, Subramanian, Sriram, & Gutwin, Carl. 2006a. Perspective cursor: perspective-based interaction for multi-display environments. *CHI*, 289–298.

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

Nacenta, Miguel A., Pinelle, David, Stuckel, Dane, & Gutwin, Carl. 2007. The effects of interaction technique on coordination in tabletop groupware. *Pages 191–198 of: Proceedings of Graphics Interface 2007*. GI '07. New York, NY, USA: ACM.

NASA. 1987. *NASA. NASA Human Performance Research Group. Task Load Index (NASA-TLX).*

Ni, Tao, & Baudisch, Patrick. 2009. Disappearing mobile devices. *Pages 101–110 of: Proceedings of the 22nd annual ACM symposium on User interface software and technology*. UIST '09. New York, NY, USA: ACM.

Nintendo. 2011. *Wiimote webpage: wii.com*.

Nolker, C., & Ritter, H. 1996. Illumination independent recognition of deitic arm postures. *Proc. 24th Annual conf. of the IEEE Indutrial Electronics Society*.

Norman, Donald A. 1999 (September). *The Invisible Computer*. MIT Press.

Norman, Donald A. 2002. *The Design of Everyday Things*. Basic Books.

Oh, Ji-Young, & Stuerzlinger, Wolfgang. 2002. Laser Pointers as Collaborative Pointing Devices. *CHI*.

Oka, Kenji, Sato, Yoichi, & Koike, Hideki. 2002. Real-Time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems. *Page 429 of: FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*. Washington, DC, USA: IEEE Computer Society.

Olsen, Jr., Dan R., & Nielsen, Travis. 2001. Laser pointer interaction. *Pages 17–22 of: Proceedings of the SIGCHI conference on Human factors in computing systems*. CHI '01. New York, NY, USA: ACM.

OpenSG. 2009. *OpenSG: OpenSource SceneGraph*. http://www.opensg.org.

OpenSG. 2010. *OpenSG Project webpage :http://www.opensg.org/*.

Oviatt, Sharon, DeAngeli, Antonella, & Kuhn, Karen. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. *Pages 415–422 of: CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM.

Paradiso, J.A., Leo, Che King, Checka, N., & Hsiao, Kaijen. 2002. Passive acoustic sensing for tracking knocks atop large interactive displays. *Sensors, 2002. Proceedings of IEEE*, **1**, 521–527 vol.1.

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

Pavlovych, Andriy, & Stuerzlinger, Wolfgang. 2009. The tradeoff between spatial jitter and latency in pointing tasks. *Pages 187–196 of: Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*. EICS '09. New York, NY, USA: ACM.

Peck, Choon Hong. 2001. Useful parameters for the design of laser pointer interaction techniques. *Pages 461–462 of: CHI '01 extended abstracts on Human factors in computing systems*. CHI EA '01. New York, NY, USA: ACM.

Pierce, Jeffrey S., Forsberg, Andrew S., Conway, Matthew J., Hong, Seung, Zeleznik, Robert C., & Mine, Mark R. 1997. Image plane interaction techniques in 3D immersive environments. *Pages 39–ff. of: Proceedings of the 1997 symposium on Interactive 3D graphics*. I3D '97. New York, NY, USA: ACM.

Quek, F. K. H., & Zhao, M. 1996. Inductive learning in hand pose recognition. *Page 78 of: FG '96: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*. Washington, DC, USA: IEEE Computer Society.

Raja, Dheva, Bowman, Doug A., Lucas, John, & North, Chris. 2004. Exploring the Benefits of Immersion in Abstract Information Visualization. *Proc. of Immersive Projection Technology Workshop*.

Reitmayr, Gerhard, & Schmalstieg, Dieter. 2005. OpenTracker: A flexible software design for three-dimensional interaction. *Virtual Reality*, **9**(1), 79–92.

Rosenberg, Ilya, & Perlin, Ken. 2009. The UnMousePad: an interpolating multi-touch force-sensing input pad. *ACM Trans. Graph.*, **28**(July), 65:1–65:9.

Russell, Daniel M., Streitz, Norbert A., & Winograd, Terry. 2005. Building disappearing computers. *Commun. ACM*, **48**(March), 42–48.

Sato, Yoichi, Kobayashi, Yoshinori, & Koike, Hideki. 2000. Fast Tracking of Hands and Fingertips in Infrared Images for Augmented Desk Interface. *Page 462 of: FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*. Washington, DC, USA: IEEE Computer Society.

Sato, Yoichi, Saito, Makiko, & Koik, Hideki. 2001. Real-Time Input of 3D Pose and Gestures of a User's Hand and Its Applications for HCI. *Page 79 of: VR '01: Proceedings of the Virtual Reality 2001 Conference (VR'01)*. Washington, DC, USA: IEEE Computer Society.

Schmidt, Dominik, Block, Florian, & Gellersen, Hans. 2009. A Comparison of Direct and Indirect Multi-touch Input for Large Surfaces. *Pages 582–594 of:* Gross, Tom, Gulliksen, Jan, Kotzé, Paula, Oestreicher, Lars, Palanque, Philippe, Prates, Raquel, & Winckler, Marco (eds), *Human-Computer Interaction – INTERACT 2009*. Lecture Notes in Computer Science, vol. 5726. Springer Berlin / Heidelberg. 10.1007/978-3-642-03655-2$_6$5.

Shoemaker, Garth, Tang, Anthony, & Booth, Kellogg S. 2007. Shadow reaching: a new perspective on interaction for large displays. *Pages 53–56 of: Proceedings of the 20th annual ACM symposium on User interface software and technology*. UIST '07. New York, NY, USA: ACM.

Shoemaker, Garth, Findlater, Leah, Dawson, Jessica Q., & Booth, Kellogg S. 2009. Mid-air text input techniques for very large wall displays. *Pages 231–238 of: Proceedings of Graphics Interface 2009*. GI '09. Toronto, Ont., Canada, Canada: Canadian Information Processing Society.

Soares, Luciano P, Jota, Ricardo, Araújo, Bruno, & Jorge, Joaquim. 2007. Automatic Color Calibration for Commodity Multi-projection Display Walls. *Symposium on Virtual and Augmented Reality*.

Sony. 2011. *PS3 Move webpage: http://us.playstation.com/ps3/playstation-move/*.

Soukoreff, R. William, & MacKenzie, I. Scott. 2004. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *Int. J. Hum.-Comput. Stud.*, **61**(December), 751–789.

Stødle, Daniel, Hagen, Tor-Magne Stien, Bjørndalen, John Markus, & Anshus, Otto J. 2008. Gesture-Based, Touch-Free Multi-User Gaming on Wall-Sized, High-Resolution Tiled Displays. *Journal of Virtual Reality and Broadcasting*, **5**(10). `urn:nbn:de:0009-6-15001`, , ISSN 1860-2037.

Subramanian, Sriram, Aliakseyeu, Dzimitry, & Lucero, Andrés. 2006. Multi-layer interaction for digital tables. *Pages 269–272 of: UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM.

Sutherland, Ivan E. 1963. Sketchpad: a man-machine graphical communication system. *Pages 329–346 of: Proceedings of the May 21-23, 1963, spring joint computer conference*. AFIPS '63 (Spring). New York, NY, USA: ACM.

Svanæs, Dag. 2000. *Understanding Interactivity: Steps to a phenomenology of Human-Computer Interaction*. Monograph, Computer Science Department, Monograph, Computer Science Department, ISBN 82-471-5201-0, NTNU, Trondheim.

Takeoka, Yoshiki, Miyaki, Takashi, & Rekimoto, Jun. 2010. Z-touch: a multi-touch system for detecting spatial gestures near the tabletop. *Pages 57:1–57:1 of: ACM SIGGRAPH 2010 Talks*. SIGGRAPH '10. New York, NY, USA: ACM.

Tan, Desney S., Czerwinski, Mary, & Robertson, George. 2003. Women go with the (optical) flow. *Pages 209–215 of: CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM Press.

Teather, R. J., & Stuerzlinger, W. 2008. Assessing the Effects of Orientation and Device on (Constrained) 3D Movement Techniques. *Pages 43–50 of: Proceedings of the 2008 IEEE Symposium on 3D User Interfaces*. 3DUI '08. Washington, DC, USA: IEEE Computer Society.

Tech., Smart. 2010. *Smart Technologies: http://smarttech.com/*.

Tse, Edward, Hancock, Mark, & Greenberg, Saul. 2007. Speech-filtered bubble ray: improving target acquisition on display walls. *Pages 307–314 of: Proceedings of the 9th international conference on Multimodal interfaces*. ICMI '07. New York, NY, USA: ACM.

Usoh, Martin, Arthur, Kevin, Whitton, Mary C., Bastos, Rui, Steed, Anthony, Slater, Mel, & Frederick P. Brooks, Jr. 1999. Walking > walking-in-place > flying, in virtual environments. *Pages 359–364 of: SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.

Vogel, Daniel, & Balakrishnan, Ravin. 2004. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. *Pages 137–146 of: UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM.

Vogel, Daniel, & Balakrishnan, Ravin. 2005. Distant freehand pointing and clicking on very large, high resolution displays. *Pages 33–42 of: Proceedings of the 18th annual ACM symposium on User interface software and technology*. UIST '05. New York, NY, USA: ACM.

Voida, Stephen, Podlaseck, Mark, Kjeldsen, Rick, & Pinhanez, Claudio. 2005. A study on the manipulation of 2D objects in a projector/camera-based augmented reality environment. *Pages 611–620 of: Proceedings of the SIGCHI conference on Human factors in computing systems*. CHI '05. New York, NY, USA: ACM.

von Hardenberg, Christian, & Bérard, François. 2001. Bare-hand human-computer interaction. *Pages 1–8 of: PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*. New York, NY, USA: ACM Press.

Ware, Colin, & Lowther, Kathy. 1997. Selection using a one-eyed cursor in a fish tank VR environment. *ACM Trans. Comput.-Hum. Interact.*, **4**(December), 309–322.

Weiss, Malte, Voelker, Simon, & Borchers, Jan. 2010a. The BendDesk demo: multi-touch on a curved display. *Pages 317–317 of: ACM International Conference on Interactive Tabletops and Surfaces*. ITS '10. New York, NY, USA: ACM.

Weiss, Malte, Voelker, Simon, Sutter, Christine, & Borchers, Jan. 2010b. BendDesk: dragging across the curve. *Pages 1–10 of: ACM International Conference on Interactive Tabletops and Surfaces*. ITS '10. New York, NY, USA: ACM.

Welch, Greg, & Bishop, Gary. 2006. *An Introduction to the Kalman Filter*. Tech. rept. University of North Carolina.

Wigdor, Daniel, Shen, Chia, Forlines, Clifton, & Balakrishnan, Ravin. 2006. Effects of display position and control space orientation on user preference and performance. *Pages 309–318 of: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York, NY, USA: ACM Press.

Wikipedia. 2011. *Wikipedia entry on flipPhones: en.wikipedia.org/wiki/Flip$_($form*.

Wilson, Andrew D. 2004. TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction. *ICMI*.

Wilson, Andrew D. 2005. PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System. *UIST*.

Wilson, Andrew D., & Benko, Hrvoje. 2010. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. *Pages 273–282 of: Proceedings of the 23nd annual ACM symposium on User interface software and technology*. UIST '10. New York, NY, USA: ACM.

Wilson, Andrew D., & Cutrell, Edward. 2005. FlowMouse: A Computer Vision-Based Pointing and Gesture Input Device. *Interact*.

Wilson, M. 2002. Six views of embodied cognition. *Psychon Bull Rev*, **9**(4), 625–636.

Wingrave, Chadwick A., Tintner, Ryan, Walker, Bruce N., Bowman, Doug A., & Hodges, Larry F. 2005. Exploring Individual Differences in Raybased Selection; Strategies and Traits. *Pages 163–170 of: Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*. VR '05. Washington, DC, USA: IEEE Computer Society.

Winograd, Terry, & Flores, Fernando. 1987. *Understanding Computers and Cognition: A New Foundation for Design*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Wu, Mike, & Balakrishnan, Ravin. 2003. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *UIST*, 193–202.

Wu, Ying, & Huang, Thomas S. 2000. View-Independent Recognition of Hand Postures. *CVPR - IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **Volume 2**(June), 2088.