Universität Konstanz
Fachbereich Informatik & Informationswissenschaft
Master-Studiengang Information Engineering

Hand Gesture Interaction for Large High-Resolution Displays
Design and Evaluation

Stephanie Föhrenbach

Konstanz, 09. Februar 2009

Masterarbeit

Masterarbeit

zur Erlangung des akademischen Grades eines

Master of Science (M.Sc.)

an der

Universität Konstanz
Fachbereich Informatik & Informationswissenschaft
Master-Studiengang Information Engineering

Thema:	Hand Gesture Interaction For Large High-Resolution Displays Design and Evaluation
Studentin:	Stephanie Föhrenbach Tulengasse 9 78462 Konstanz
Matrikel-Nr.:	01/632621
Studienfach:	Information Engineering
Schwerpunkt:	Computer Science
Themengebiet:	Angewandte Informatik
Erstgutachter:	Prof. Dr. Harald Reiterer
Zweitgutachter:	Prof. Dr. Daniel A. Keim
Abgabetermin:	09. Februar 2009

To

Helga Weniger

My beloved godmother

who departed from us too early.

Abstract

Large high-resolution displays are widely used in academic and business contexts. Those kinds of displays offer great advantages for information visualization and can improve user performance. However, they are challenging for human-computer interaction, as they lead to more physical user movements. Therefore flexible input devices and interaction techniques are needed which allow interaction from any point and distance. This thesis investigates hand gesture interaction techniques, which meet the mobility requirement, but also provide a natural and expressive way of interaction.

For the two most basic tasks in today's graphical user interfaces "pointing" and "selecting", we identified suitable hand gesture interaction techniques, based on hand gestures used in human-human communication and previous research. To underline the analogy to real-world interaction, we provided additional tactile feedback to the users' fingertips for target crossing to enhance the selection task. Previous research suggests that different movement directions of input devices, achieved with different physical user movements can influence user performance. With our hand gestures performed in mid-air no external physical support can be given, to guide user movements and compensate for irregularities. Different directions for cursor movements may therefore reveal different user performances.

To assess the performance and acceptance of our proposed hand gesture techniques for pointing and selecting, and the influence of additional tactile feedback and movement direction we conducted a comparative evaluation study based on the ISO 9241-9. The 20 participants performed horizontal and vertical one-directional tapping tasks with hand gesture input with and without tactile feedback in front of the Powerwall of the University of Konstanz, a large high-resolution display (5.20x 2.15 m). To track hand and finger movements and provide tactile feedback, our participants were equipped with a commercial data glove. For fast and robust gesture classification we applied an algorithm based on geometrical gesture models and state dependent threshold comparison.

In contrast to previous research we cannot confirm a benefit of tactile feedback on user performance. Furthermore we found a significant difference in favour of the horizontal target alignment compared to the vertical one in terms of the effective index of performance. The non-tactile version of our hand gesture interaction techniques was very well received by our participants, and the effective index of performance with a mean of 2.53 bits/s for vertical and 3 bits/s for horizontal target alignment is promising and suggests that our hand gesture interaction techniques provide an adequate and valuable interaction technique for large high-resolution displays.

To navigate within the presented information space on a large-high resolution display and explore it, users can physically move. From a distant position they can gain overview information, while moving closer reveals more details. However, physical navigation may not always be sufficient. Some parts of the display may always stay distant to the user, such as the upper part. To complement physical navigation and compensate for its limitations, additional interaction techniques are needed for virtual navigation.

Therefore, we extended our set of gesture techniques to support "panning", "dragging" and "zooming" tasks too, as those tasks are commonly used for virtual navigation. Based on interaction with physical objects and human-human communication, we identified suitable hand gesture interaction techniques, which fit seamlessly in with our existing gesture set.

Limitations of commercially available data glove solutions, such as arising issues of hygienic or fit, and the observed restriction of user movements, motivated the design of Whitey, a novel data glove solution. Whitey combines an optical tracking system with a modified textile glove and a finger classification algorithm. For the classification of fingers this algorithm takes advantage of biomechanical constraints and heuristic knowledge on finger movements. Whitey can be adapted to different hand sizes, and used almost instantly by different users without the need for an individual calibration session after an initial set up.

We conducted informal user studies to get initial feedback and first experience on the usability of our extended hand gesture interaction set and Whitey. We found that users could easily learn and apply our techniques. We could further gain valuable insides for fine tuning Whitey and our hand gesture interaction techniques to improve user interaction.

For future work we plan to further extend our set of hand gesture interaction techniques to utilize more of the potential hand gesture interaction holds for human-computer interaction, in terms of naturalness and expressiveness.

Parts of this thesis were published in:

S. Foehrenbach, W. A. König, J. Gerken, and H. Reiterer. Natural interaction with hand gestures and tactile feedback for large, high-res displays. In *MITH'08: Workshop on Multimodal Interaction Through Haptic Feedback, held in conjunction with AVI'08: International Working Conference on Advanced Visual Interfaces*, May 2008.

S. Foehrenbach, W. A. König, J. Gerken, and H. Reiterer. Tactile Feedback enhanced Hand Gesture Interaction at Large, High-Resolution Displays. *JVLC Special Issue on Multimodal Interaction through Haptic Feedback (selected papers from the International Workshop MITH'08),* 2009. (submitted)

Contents

List of Abbreviations

3dof : Three degrees of freedom
6dof : Six degrees of freedom
dof, DOF : degree of freedom
HCI : Human-Computer Interaction
LHR : Large high-resolution
LHRD : Large high-resolution display
UFOV : Useful field of view
WIMP : Windows, Icons, Menus, and Pointing

1. Introduction

“Among different body parts, the hand is the most effective, general-purpose interaction tool due to its dexterous functionality in communication and manipulation”

[Erol et al. 2007]

1.1 Motivation

Parts of this chapter have been published in [Foehrenbach et al. 2008], but have been further enhanced for the purpose of this thesis.

In application domains where collaboration, presentation or exploration and analysis of large information spaces are predominant tasks large high-resolution displays are widely used. These wall-sized displays offer great opportunities for information visualization [Yost et al. 2007] and improve user orientation and search performance [Ball et al. 2005], but also lead to more physical navigation [Ball et al. 2007]. Physical navigation describes the use of bodily movements, e.g. walking, to navigate within the displayed information space. From a distant position of the display, users can gain overview information while moving closer to the display reveals more details and user can gain in-depth knowledge. To not impede user interaction, input devices and interaction techniques are needed which allow flexible interaction from any point and distance. Hand gesture input as an interaction technique can meet this mobility requirement.

Moreover, hands are one of our main tools when interacting with “real-world” (referring to physical non-digital) surroundings. Hands are used to manipulate physical objects (e.g. grab and move items) and to complement spoken language in human-human communication (e.g. “the fish was *this* big” or “look *there*”). The ability of the hands to take on a broad variety of shapes and functions makes them highly valuable to humans. If hands are so valuable for interacting with “real-world” surroundings, why not use them for human-computer interaction in a more direct manner than they already are?

Instead of using the hand to operate an intermediary input device, hand gestures could be used by the user to interact. Thereby, human-computer interaction designer can take advantage of the capacities of the hand, pre-acquired motor skills and experience of users in manipulating and navigating within their “real-world” surroundings. If hand gesture interaction mimics interaction with the “real-world”, users are familiar with the manual skills needed to accomplish a task, and instead of concentrating on how to operate the input device, they can focus on the task at hand.

Furthermore, each different aspect of hand movements, such as the shape of the hand and its finger, wrist rotation or movement speed can be used to convey meaning to the computer. Therefore hand gestures can easily specify multiple input dimensions, which can be used to create a terse and powerful interaction.

Hand gesture interaction techniques can meet the mobility requirement and provide users with an input device that not only allows interaction from any point and distance but can also lead to a natural and expressive way of interaction. Previous research has proposed hand gesture interaction techniques for distant pointing and clicking [Vogel & Balakrishnan 2005] and object manipulation [Baudel & & Beaudouin-Lafon 1993] at large displays. Inspired by this work, and building on the insights provided, we are going to design a set of hand gesture interaction techniques that not only provide techniques for distant point and selection, but also for virtual navigation techniques such as panning, dragging and zooming to address limitations of physical navigation.

1.2 Outline

In **Chapter 2** we will introduce large high-resolution displays. We will describe their features, the application areas and the benefits for the user. We will introduce physical and vertical navigation as strategies to compensate for the limits of the human visual system, which may be exceeded by large high-resolution displays, and substantiate the need for a mobile input device.

Chapter 3 deals with foundations and related work on hand gesture interaction. The definition of hand gestures is followed by an outline of potential and prospects of hand gesture interaction. Biomechanical constrains on hand movements, the highly discriminative sense of touch in the fingertips, and techniques for tracking hand movements are the subsequent topic. Related work on hand gesture interaction will be then presented, focusing on the use of hand gestures for distant interaction on large displays.

Chapter 4 opens with a distinction between dynamic and static hand gestures. It then illustrates the approach we applied for recognizing static hand gestures that we used for our hand gesture interaction techniques described in chapter 5 and 7.

We will present our hand gesture interaction techniques for distant pointing and selecting in **chapter 5**. We although analyze factors which might influence the performance of those two techniques, namely movement direction and additional tactile feedback. A controlled experiment we conducted to investigate on the usability of our hand gesture interaction techniques for pointing and selecting, and potential influencing factors is also described. At the end of the chapter, we present our conclusions and possible implications for interaction design.

In **Chapter 6** we present Whitey, a data glove solution we have developed for tracking hand movements. We will then describe its components and the algorithms we applied. Thereafter, we will compare Whitey with other commercially available data glove solutions. The chapter concludes with a description of how Whitey can be adapted to settings other than the one we used it for.

In **Chapter 7**, we describe further hand gesture interaction techniques, widening the range of gesture techniques we employed to support panning, dragging and zooming tasks too.

Chapter 8 describes an informal user study we conducted to gain initial user feedback and first experience on the usability of our extended set of hand gesture interaction techniques and Whitey. We present our observations, conclusions and derived suggestions for fine-tuning our techniques.

Finally, in **Chapter 9**, we summaries our main results and give an outlook on future work.

2. Large High-Resolution Displays

In this chapter, we will introduce large high-resolution displays. We will describe their features, the application areas and the benefits for the user. We will then focus on the human visual system, whose capacities in terms of visual field of view and spatial resolution may be exceeded by the physical size and resolution of large high-resolution displays. Following this, we will then introduce physical and vertical navigation as strategies to compensate for the limits of the human visual system and substantiate the need for a mobile input device.

2.1 Features

Large high-resolution displays (= LHRDs) are created using various hardware configurations, ranging from combining multiple monitors, to tiled LCD (= liquid crystal display) panels to back projection-based seamless displays. Two common features of those displays are increased physical size and high resolution [Ni et al. 2006]. They make it possible to display large amounts of data, to display large objects, for instance concept sketches of cars, with a 1-1 scale [Buxton et al. 2000] and further multiple users to view and interact simultaneously [Cao et al. 2008].

2.2 Application areas and benefits for the user

Their features and the corresponding abilities to (1) display large amounts of data, (2) display large objects in full scale and (3) support multiple user, makes LHRD suitable for various domains and tasks. They are widely used for monitoring large amounts of data in command and control center, for example in traffic management or utility monitoring [Barco] [eyevis 2008]. LHRD are used for scientific visualizations, in particular for exploration and analysis of large data sets [Spiegel 2005], such as geo-spatial data sets [Sips et al. 2006]. Companies in the automotive industry, e.g. General Motors [TechnologyReview 2002], Ford [Dexigner 2007] or Nissan [Nissan 2007], apply LHRDs in the vehicle design process to create, verify and modify design models in full 1-1 scale. In applying LHRDs, they are able to accelerate the product design process and increase the design quality, to bring better products to market faster. Besides supporting collaborative design processes, LHRDs can also be utilized as an electronic whiteboard for brainstorming sessions, another task performed in collaborative group work [Guimbretière et al. 2001]. LHRDs can further be found in public spaces for presentation of information [Barco 2008] [ZKM 2008], or to support TV coverage [Wired 2008] with their ability to present interactive visualizations of changing information. With the current trend to increase size and resolution of regular displays, in combination with decreasing prices, large high-resolution displays may become more easily available and we can easily imagine seeing them more often in public and also in private spaces in the near future [Vogel & Balakrishnan 2005].

LHRDs provide unique advantages for presentation of data, but how do users benefit from their increased size and resolution? It was found that when compared to smaller displays, large high resolution displays lead to less user frustration [Ball & North 2005] [Shupp et al. 2006], less window management [Ball et al. 2005] and improve user performance in geospatial search and navigation tasks [Ball & North 2005] [Ball et al. 2005].

Figure 1: Large high-resolution displays in use. Left: Monitoring information in a control center, taken from [ict]. Right: Explorating and analyzing the scientific visualization of a large geo-spatial data set, taken from [Sips et al. 2006].

However, the increased physical size and resolution of LHRD can exceed the capacities of the human visual system in terms of visual field of view or spatial resolution [König et al. 2008].

2.3 Visual acuity and Visual Field of View

The **visual angle** is a key concept to describe and understand the capacities of the human visual system in terms of visual field of view and resolution. The *"[...] visual angle is the angle subtended by an object at the eye of an observer"* [Ware 2004, p. 40]. Figure 3 illustrates the visual angle. The visual axis originates from the fovea and extends to the point that is being looked at directly. The fovea is a small area in the center of the retina, specialized for fine pattern discrimination and color perception [Rosenbaum 1991, p. 164].

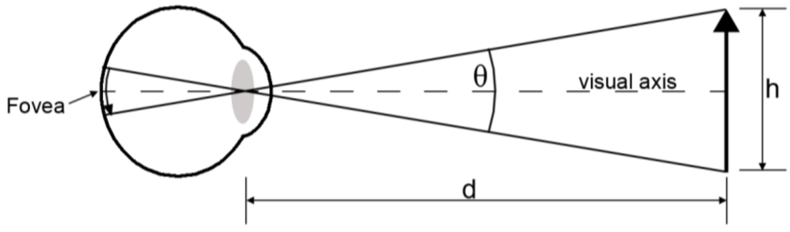


Figure 2: Visual angle of an object (adapted from [Ware 2004, p. 40]). The visual axis originates from the fovea to the location which is directly looked at.

Visual angles are defined in degrees, minutes and seconds (1° degree = 60' minutes = 360'' seconds). The visual angle of an object can be calculated by using equation (1) [Ware 2004, p. 40].

$$\theta = 2\arctan\left(\frac{h}{2d}\right) \quad (1)$$

As the equation shows, the visual angle depends on the size of the object (h) and the distance between the object and the observer, the viewing distance (d) that is, the smaller the object and the larger the viewing distance, the smaller the visual angle. The visual angle is used to describe visual acuity (or more technically, spatial resolution) and the visual field of view.

Figure 3: The Landolt C

Visual acuity measures our ability to resolve spatial patterns. Normal visual acuity is defined as *"the ability to resolve a spatial pattern separated by a visual angle of one minute of arc"* Hermann Snellen, 1862, cited from [König 2006]. Normal visual acuity can be illustrated at the example of the Landolt C, a symbol shaped as a circle with an opening, resembling the letter C (see Figure 3). If the visual angle of its opening falls below 1' the visual system of the observer cannot detect the spatial pattern, formed by the circle and the opening, and the Landolt C seems to resemble an O. However, if the visual angle of the opening matches or exceeds 1' the spatial pattern can be detected and the observer perceives a C.

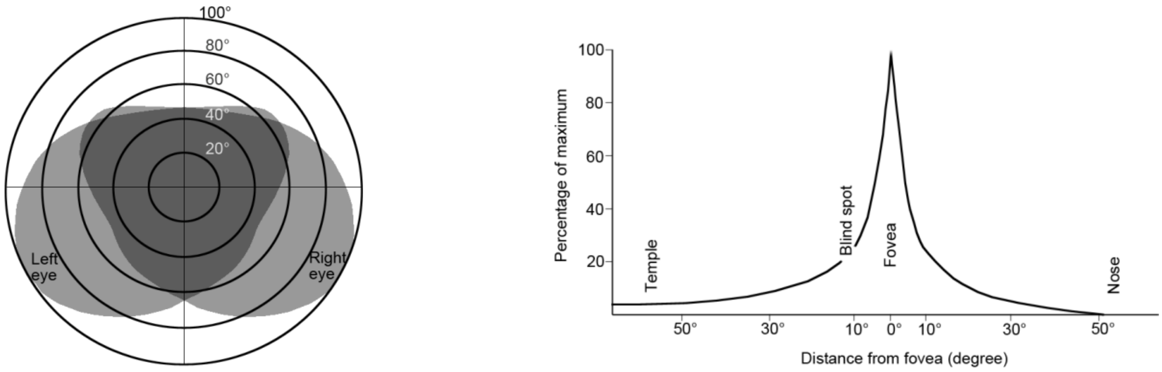


Figure 4: Left: Visual field of view of a person gazing straight ahead, Right: Distribution of visual acuity. Both adapted from [Ware 2004, p50-51]

The visual **field of view** is the number of degrees of visual angle that can be seen. Figure 4, left illustrates the visual field of view with combined input of both eyes. In this visual field of view, visual acuity is distributed in a non-uniform manner. For each eye visual acuity is highest in the fovea, and drops rapidly with increasing distance from the fovea (see Figure 4, right) [Ware 2004, p. 50].

Only a small part of the visual field of view falls within a "useful field of view (UFOV)". The UFOV is a concept to describe the size of the region where information can be rapidly taken in. The size of the UFOV varies, depending on the information presented and the task, and can range from 1° up to 15° [Ware 2004, p. 147], where *"The central two degrees of visual angle is the most useful […]"* [Ware 2004, p. 364]. The UFOV and the accompanied area with sufficient visual acuity can be experienced when fixing a written word within a sentence. The other words in the sentence located only few centimeters away cannot be read unless the eye is moved towards them.

At the end of the previous section, we argued that the physical size and resolution of LHRDs can exceed the human visual system in terms of spatial resolution or visual field of view. We will illustrate this at the example of two LHRDs mentioned below.

Let us consider a LHRD build with 24 tilted 17’’ monitors and a physical size of approximately 2.90x0.81 m and a resolution of 96 DPI (see Figure 5). Shupp et al. [2006] used such a LHRD to evaluate the effect of viewport size and curvature of LHRD. The physical size taken up by one pixel is 0.264mm. Using equation (1) a visual angle of 1’, which is the smallest angle that can be detected with normal visual acuity, is subtended by the pixel at a viewing distance of 90.756 cm. This means that a user standing centred in front of the display with a distance of 90.756 cm can see a pixel when looking straight ahead. However, the user is not able to detect a pixel on the right or left side of the display from his position. This particular LHRD can exceed the human visual system in terms of spatial resolution if the viewing distance is larger than 90.756 cm.

Figure 5: LHRD build with 24 tiled 17’’ monitors, with a resolution of 96 DPI (taken from [Shupp et al. 2006])

The Powerwall of the University of Konstanz is a wall-sized display with a resolution of 4640x1920 pixels and a physical dimension of 5.20x2.15 meters. The upper border of this display is located at 2.65 m. Due to the vertical extension of the visual field of view, a user looking straight ahead with a viewing distance of 80 cm at the height of 1.50 m cannot see the upper part of the display (approximately 20 cm). The Powerwall of the University of Konstanz can therefore exceed the human visual system in terms of visual field of view when users are close to the display. Note, here we have considered only the total visual field of view with a vertical angle of approximately 45° above the visual axis (see Figure 4,left on page 5). The useful field of view, describing the screen space from which we can rapidly take information in with high visual acuity, is much narrower. Therefore only a small area of the information space presented on the display can be perceived instantly at a glance.

The area comprised within the visual field of view and the amount of details which can be resolved depend on the size of the object and the viewing distance. In the context of LHRDs, users can adjust those two parameter in (1) either physically move (=adjusting the viewing distance) or (2) use dedicated interaction techniques for virtual navigation while standing at a static position, such as panning, dragging (=adjusting the viewing distance), or zooming (=adjusting the size of objects).

2.4 Physical and virtual navigation

Physical movement of the user (e.g. walking or turning) changes the viewing distance between the user and objects on the display. From a distant position users can gain an overview of the displayed information space, while moving closer to the display reveals more details and users can gain in-depth knowledge. This physical user's movement is called physical navigation. Note that this is leads to a distinct advantage of LHRDs: taking into account the physical movement of the user, a single visualization can contain overview and detail information.

Virtual navigation describes the use of dedicated interaction techniques (e.g. panning, dragging or zooming) to adjust the viewing distance or the size of objects to gain overview or detail information, while the user itself can maintain a static position.

Considering the relation between the visual representation of data and the capacities of the human visual system, virtual navigation adjusts the visual representation of data to the capacities of the human visual system, whereas physical navigation goes the opposite way and adjusts the capacities of the human visual system to the visual representation.

While physical navigation may be a necessity for static visualizations at LHRD, to perceive different information (overview vs. detail) it is not just a necessity but also holds distinct advantages for the user. Researchers found that when given the chance to choose between physical and virtual navigation for spatial visualization tasks, users preferred physical navigation [Ball et al. 2007]. Furthermore, Ball et al. [2007] report that large displays lead to more physical navigation, which correlates with reduced virtual navigation and improved user performance for basic search and navigation tasks with spatial visualizations.

The capacities of LHRD considering the amount of information which can be displayed at a glance are tremendous and physical navigation allows a "device-less" change in the granularity. However, the possibility of additional control with a manual input device is desirable, as

> *"Even though all information could be visualized at a glance, users want to manipulate or annotate data or explore related information (e.g. details-on-demand) directly, instantly and right on the spot" [König et al. 2008].*

To not impede fluid user interaction a suitable input device should therefore allow interaction from any point and distance.

The drawbacks of physical navigation give further rise to the need of such a mobile input device. Despite the advantages of physical over virtual navigation, physical navigation may not always be sufficient [Ball et al. 2007]. Some parts of the display may always stay distant to the user, for example the upper part of a wall-sized display, which introduces a limit to the smallest viewing distance achievable and amount of detail perceivable with physical navigation. The largest viewing distance may be limited by walls or furniture. Another drawback of physical navigation is that it can cause more fatigue than virtual navigation [Ball et al. 2007]. The increased use of physical user movements, such as walking, for physical navigation can cause higher fatigue than the movements needed for virtual navigation, which can be performed while the user maintains a static position.

Virtual navigation can compensate for the limitations of physical navigation. If provided by the user interface, it can be used instead or additional to physical user movement to adjust the viewing distance. As the user can maintain a static position while virtually navigating, fatigue caused by walking can be reduced. Different from physical navigation, techniques for virtual navigation, such as geometric zooming, can also change the physical size of objects, which also influences the amount of detail perceivable or overview information that can be gained. Virtual navigation can be used instead of physical navigation, but it can also be used to extend the amount of detail or overview information that can be gained, beyond what is possible with physical navigation only. As limitations of physical navigation can be tackled by means of dedicated interaction techniques devised for virtual navigation, the mobile input device needed to enhance physical navigation should be able to support techniques that can be used for virtual navigation.

Traditional mice and keyboards restrict user movements, as they require a stable surface on which they can be operated. Wireless mice which can be operated in mid-air provide more mobility [König et al. 2008]. However they perform much worse compared to traditional mice [MacKenzie & Jusoh 2001]. König et al. propose laser pointer for interaction at LHRDs. Even if the performance of their laser pointer point and select interaction falls below the performance of a stationary mouse (13%), the gain in user flexibility and the intuitiveness of interaction makes this difference appear marginal [König et al. 2007b]. Therefore the solution proposed by them is a highly valuable for interaction at LHRD.

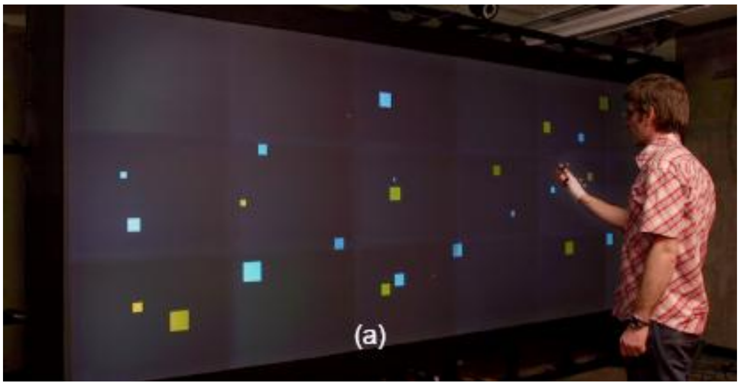


Figure 6: Distant pointing and clicking with hand gestures (taken from [Vogel & Balakrishnan 2005])

Hand gestures are another potential input modality. They do not only address the mobility requirement but can also lead to a natural, terse and powerful interaction (see chapter 3.2). User can interact directly with the application utilizing pre-acquired motor skills from real-world interaction, without the need of operating an intermediate input device, which has to be explicitly learned and can limit the input capacities of the human hand. Inspired by the work of Vogel & Balakrishnan [2005] who proposed the use of hand gestures for distant pointing and clicking on large high-resolution displays, and the potential we attribute to hand gesture interaction, we are set to investigate further into the use of hand gestures as a mobile input device for LHRD, which allows interaction from any point and distance.

3. Hand Gesture Interaction

In this chapter we will describe foundations and related work on hand gesture interaction.

3.1 Definition of Hand Gesture

> *"A gesture is a motion of the body that contains information. Waving goodbye is a gesture. Pressing a key on a keyboard is not a gesture because the motion of a finger on its way to hitting a key is neither observed nor significant. All that matters is which key was pressed"* [Kurtenbach & Hulteen 1990].

According to Kurtenbach & Hulteen, a bodily movement is considered a gesture if it contains information, it is observed and it is significant. Considering this definition, we define hand gestures in the context of human-computer interaction for the purpose of this thesis as follows: A hand gesture is a movement of the hand and fingers, performed by the user with the intention to interact with the computer. Hand and finger movements are significant and directly monitored, instead of monitoring the movement of an intermediary physical input device operated by the hand, such as a mouse or stylus. Each hand gesture conveys meaning to a computer. We thereby do not limit hand gestures to dynamic hand and finger movements, but also include shapes which can be adopted by the hand and its fingers. A shape is thereby referred to as a "hand posture".

3.2 Prospects of Hand Gesture Interaction

Hand gesture input offers distinct advantages which favour hand gesture interaction, namely: naturalness, expressiveness and mobility.

Naturalness

The hand is used every for a variety of tasks, using skills which require little thought [Sturman 1992]. For instance, humans can grab a physical object and turn it, with little or no thought on how to perform the necessary movements or coordinate the limbs involved. The focus is on the task that is performed rather than on the tool (the hand) that is used to perform it. Humans start to learn the necessary skills to manipulate their physical surroundings from the day on which they are born. Besides the use of hand movements to manipulate physical surroundings, hand movements are further used in human-human communication, either to complement speech or to substitute it in non-verbal communication. Human-computer interaction can take advantage of those pre-acquired skills and knowledge and apply hand movements in a similar fashion for interacting with digital objects. Using pre-acquired skills can make tasks easier to learn and master [Sturman 1992] because users do not have to concentrate on how to operate the input device and can instead concentrate on accomplishing the task.

Natural interaction is *"Typically used to refer to interaction techniques for the computer which are modelled on the ways people interact with physical objects in the everyday world"* [Harper et al. 2008]. Naturalness of hand gesture input can lead to a natural interaction, if interaction techniques mimic the way people interact and communicate in the everyday world. This implies that not only hand movements are used for input but also the mapping to the action they invoke resembles experience from everyday interaction and communication.

For instance, pointing with the hand, as used in human-human communication to indicate a position, has been reused by researchers in human-computer interaction to position a cursor on the display [Schapira & Sharma 2001] [Tse et al. 2007] [Vogel & Balakrishnan 2005]. In human-computer interaction, the user typically faces the display if the user aims at positioning the display cursor, hence a hand movement used in a similar manner in human-human communication can be used to accomplish a natural interaction. Pointing with the thumb is typically used in human-human communication to indicate a position located behind the speaker [Kendon 2004]. Although pointing with the thumb is a natural hand movement for indicating a location, reusing it in human-computer interaction to perform the task of positioning a display cursor would clash with user's experience and behaviour observable in human-human communication, as the display cursor is located in front and not behind the user. However, using pointing with the index finger or the palm however, could lead to a natural interaction, as those hand movements are used in human-human communication to indicate a position located in front of the speaker [Kendon 2004].

Even if hand gesture interaction is natural it may not necessarily be intuitive. Hand gestures (hand movements which convey meaning to a computer) are not self-revealing [Baudel & Beaudouin-Lafon 1993]. Unlike physical input devices whose hardware design gives hints on how to use them [Norman 2002], such as for instance buttons are for pressing and sliders are for sliding, users might not intuitively know, which hand movements are understood by the application. Performing hand movements might be easy, but users have to get hints about which of their pre-acquired skills they can use for human-computer interaction.

Expressiveness

„*The position and movements of the hand and fingers provide the potential for higher power of expression”* [Baudel & Beaudouin-Lafon 1993]. In addition to determining “hand gesture recognized yes / no”, further aspects of hand movements such as posture, movement speed or rotation of the hand can be used for input. A hand gesture can therefore not only specify a command but also its parameters [Baudel & Beaudouin-Lafon 1993]. A hand-grabbing movement combined with a rotation of the palm can be used to issue a “rotate object” command and to specify the angle of rotation.

Considering the possible expressiveness of hand gesture input, a terse and powerful interaction can be accomplished [Baudel & Beaudouin-Lafon 1993], which can empower user input.

Mobility

Unlike most other physical input devices, hands are always close to the user's position. Hands can be used as a mobile input device for human-computer interaction. Users can move freely, while having the input device nearby at any point and position.

However, much depends on the technical solution applied for tracking hand movements. While hands are always close to the user, in order to use hand movements for input they have to be tracked. Given a specific context of use, the technical solution applied for tracking hand movements should take into account the desired extend of user mobility and not impose restrictions on it.

Conclusion and Discussion

If at least one of the three characteristics (naturalness, expressiveness or mobility) is desirable for human-computer interaction in a specific context of use, hand gestures could be suitable for input. Mobility, for instance, makes hand gesture input an option for interaction with LHRD. The mobility of hand gesture input can be used to enhance physical navigation without impeding user interaction. Naturalness can make hand gesture input an option for interaction, if the input needed to accomplish a task maps well onto existing manual skills (referring to hand movements) and experience of users. The naturalness of hand gesture input can help to reduce learning time.

However, taking advantage of mobility and naturalness can lead to unwanted side effects. The use of hand gestures derived from hand movements used for human-human communication and constantly monitoring the hand can give rise to the issue of how to distinguish hand movements performed to convey meaning to the computer from hand movements performed to convey meaning to other people in human-human communication. In such cases falsely identified hand gestures might be an unwanted side effect. Charade is a system in which hand gestures are used to control a presentation. Hand gestures are only identified if the hand points near the presentation screen, hand movements performed when the hand points to other areas are ignored [Baudel & Beaudouin-Lafon 1993]. Another approach to address the issue of false positives is, to consider which hand movements are likely to be performed casually by the user in the interaction scenario. The likelihood of false positives (referring to hand gestures identified based on hand movements performed without the intention to interact with the computer) can reduced if those hand movements are not used for hand gesture input. For instance, hand movements derived from human-human interaction may not be suited to serve as hand gesture input for applications supporting collaborative group work, where people typically communicate with each other during group work.

3.3 The Human Hand

> *"With approximately 30 dedicated muscles and approximately the same number of kinematic degrees of freedom, the hand can take on all variety of shapes and functions, serving as a hammer one moment and a powerful vice or a delicate pair of tweezers the next"* [Flanagan & Johansson 2002].

The broad variety of shapes and functions the hand can take on and perform makes the hand a highly valuable tool for us to interact with our physical surroundings and to communicate with other people. We can use the hand as a powerful tool to move heavy objects or crash nutshells, to perform complex high precision tasks such as tying up shoelaces or shuffling cards, as well as soft and delicate tasks such as stroking a cat. For all those tasks hands are not only used to act but also to perceive. The highly discriminative sense of touch in the fingers [Kandel et al. 2000, p. 341-345] makes it possible, for example, to perceive information on details of the surface of objects which can then be used to adjust hand movements.

In some cars (e.g. a Peugeot 307), a remote control for the mp3 player/radio is located behind the steering wheel. Buttons placed on the remote control can be operated with the fingers, while the hand still holds on to the steering wheel. To locate the buttons, the finger tips and the sense of touch is used to identify the gaps between the buttons. Eyes are not needed for operating the remote device and interact with the mp3 player/radio, and therefore the visual channel can be used for other tasks, such as monitoring the road ahead. The hand perceives and acts for operating the remote control.

Human-Computer Interaction can use the hand for input but also for output. Movements of the hand and its finger for input from the human to the computer (=hand gesture input), where *"[…] the hand, which can technically be considered as a complex input device with more than 20 DOF, can become an easy to use high DOF control device"* [Erol et al. 2007] and the sense of touch as a feedback channel from the computer to the human.

3.3.1 Hand Movements

Limbs are in general moved with a coordinated activation of many muscles acting on skeletal joints [Kandel et al. 2000, p. 686-693]. Most of the muscles for hand movements are in the forearm [Jones & Lederman 2006, p. 16]. Power from the muscles in the forearm is transmitted into the hand by means of long tendons. Therefore most of the muscle mass used for hand and finger movements lies outside of the hand. *"This arrangement allows the hand to be light and flexible without sacrificing strength"* [Sturman 1992]. Furthermore, some muscles, known as intrinsic hand muscles, are located inside of the hand. The intrinsic hand muscles are responsible for minimal yet precise finger movements [Tortora & Derrickson 2006, p. 444].

We can distinguish hand movements in (1) palm movements which are performed mainly in moving the palm (which also moves the fingers) and (2) finger movements which can be performed by the fingers. The following movements (described in [ISO 9241-9 2000]) result in moving the palm, where the first four listed result from movements at the wrist, and the last two ones (supination, pronation) result from a movement of the forearm:

- **Flexion:** bending the hand at the wrist, toward the inside of the hand (Figure 7a).
- **Extension:** bending the hand at the wrist, away from the inside of the hand (Figure 7b).
- **Ulnar Deviation:** bending the hand at the wrist in the plane of the palm, away from the axis of the forearm, towards the direction of the little finger (Figure 7c).
- **Radial Deviation:** bending the hand at the wrist in the plane of the palm, away from the axis of the forearm, towards the direction of the thumb (Figure 7d).
- **Supination:** lateral rotation of the hand, resulting from a rotation of the forearm (Figure 7e).
- **Pronation:** medial rotation of the hand, resulting from a rotation of the forearm (Figure 7f).

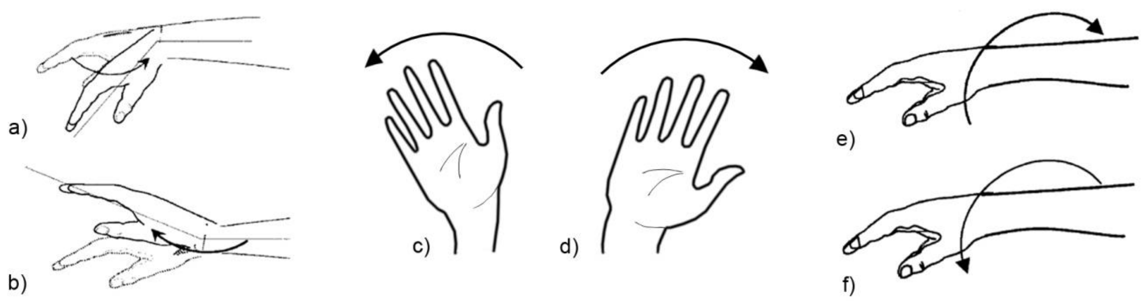


Figure 7: Palm movements, arrows indicate movement direction. (a): Flexion, (b): Extension, (c): Ulnar Deviation, (d): Radial Deviation, (e): Supination, (f): Pronation. Adapted from [ISO 9241-9 2000].

The fingers can perform the following movements (taken from [Tortora & Derrickson 2006] [Tözeren 2000]):

- **Flexion:** moving the fingertip towards the inside of the hand (Figure 8a).
- **Extension:** moving the fingertip away from the inside of the hand (Figure 8a).
- **Abduction:** moving the finger away from an imaginary line drawn through the axis of the middle finger (Figure 8b).
- **Adduction:** moving the finger towards an imaginary line drawn through the axis of the middle finger (Figure 8c).
- **Opposition:** the opposition is a unique movement of the thumb, where the thumb is moved above the inside of the palm with the possibility to touch the tips of the remaining fingers (Figure 8d).
- **Circumduction:** a circular movement of a distal limb, such as the fingers, is referred to as circumduction. However, it is not a unique movement but a sequence of flexion, abduction, adduction and extension.

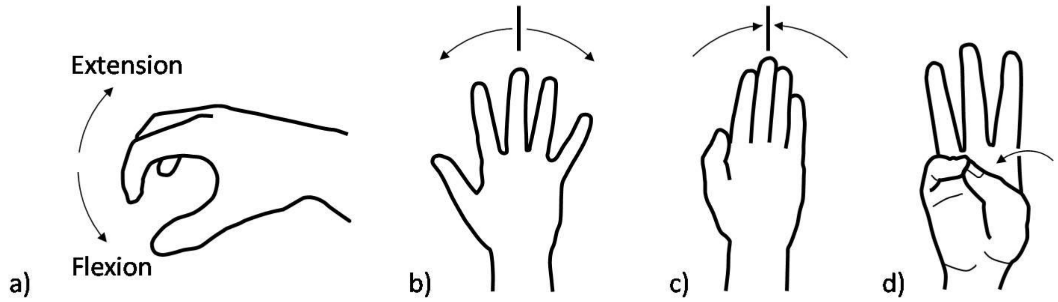


Figure 8: Finger movements, arrows indicate movement direction. a): Flexion and Extension, b): Abduction, c): Adduction, d): Opposition

Those hand movements are the biomechanical conditions given to perform hand gestures, and also describe constraints which movements are possible and which are not.

3.3.2 Sense of Touch

With the sense of touch, sensations directly applied to the skin can be perceived. Four sensations are associated with the human sense of touch: pressure, stretching, touch and vibrations. A mechanical force pressing statically against the skin creates a sensation of pressure, extending the skin is perceived as stretching, a light and short contact with the skin leads to the sensation of touch, mechanical vibrations of more than 10 Hz applied to the skin lead to a sensation of vibration [Mutschler et al. 2007, p. 698].

Fingers reveal a highly discriminative sense of touch. Although high in the fingers, the spatial resolution (called tactile acuity) is distributed in a non uniform manner in the human skin and is much lower in other parts. Tactile acuity is the ability to distinguish two simultaneously applied tactile stimuli, for instance when pressing the tips of two pencils against the skin. With the high tactile acuity at our finger tips we can identify two stimuli if the two contact locations have a distance of approximately 2 mm (or larger). At the inside of the hand we need a distance of approximately 12 mm (or larger) to identify two stimuli [Goldstein 2002, p. 540-541] [Mutschler et al. 2007, p. 698].

Figure 9: Left: A Braille display connected to a laptop. Right: Pin arrays (adapted from [BrailleNet 1998])

The high tactile acuity of our fingertips allows us to perceive detail information on the surface of objects. Braille displays (see Figure 9) are devices, which use the high spatial resolution of the fingertips, to allow visually impaired persons to interact with the computer. A Braille display can be connected to the computer, and can be used both as an output and input device. One of its main functions is the translation of written text into physical Braille characters. Each character is presented to the user with an array of pins. This pin array can resemble each Braille character in dynamically adjusting the height of each single pin. Depending on the Braille display the pin arrays are arranged in one or multiple lines. Visually impaired users can read text in moving their fingertips along the lines of pin arrays.

Besides the use of the sense of touch as a feedback channel to support human-computer interaction of visually impaired users, other areas of human-computer interaction also make use of tactile feedback. For example in providing additional feedback for text entry tasks performed on mobile devices [Brewster et al. 2007] or to signal contact of the hand with an virtual object in virtual [Scheibe et al. 2007] or augmented reality [Buchmann et al. 2004].

3.4 Tracking Hand Movements

Different approaches have been introduced for tracking hand movements to provide hand-input for human-computer interaction. Those approaches can be distinguished, according to whether objects are attached to the user’s hand or not, into (1) computer-vision-based, non-contact and (2) glove-based approaches.

For computer-vision-based, non-contact approaches, movements of the user’s bare hand are captured with one or multiple video cameras. The images are further analyzed and processed to detect the hand. No physical objects are attached to the hand to support the adjacent processing steps. With glove-based approaches, the user either wears a dedicated data glove with build-in sensors or other physical objects get attached to the user’s hand, which can be viewed as a minimized glove, to ease the detection of hand movements by tracking systems located distant from the hand.

3.4.1 Computer-Vision Based Non-Contact Tracking of Hand Movements

Computer-vision-based non-contact approaches capture hand movements with one or multiple video cameras neglecting the need to attach physical objects to the user’s hand. Therefore users can immediately start to interact. Those approaches have the potential to provide an unencumbered interaction [Erol et al. 2007] and users cannot get disturbed by potentially intrusive hardware placed at their hand.

Capturing hand motion in real time with computer-vision-based, non-contact approaches is an active area of research (see [Erol et al. 2007] for a comprehensive review on existing solutions). Applying computer-vision, non-contact approaches to capture the real 3D motion of the hand and recover the full degree of freedom (dof) hand motion from images, is a challenging problem in the context of HCI, and *"[...] several challenges including accuracy, processing speed, and generality have to be overcome for a widespread use of this technology"* [Erol et al 2007].

The design of computer-vision, non-contact solutions encounters several difficulties. The hand is a flexible object which can take on a variety of shapes. This capacity of the hand gives rise to two difficulties. The first is self-occlusion, which describes the fact that the projection of the hand onto an image plan can contain many self-occlusions, where parts of the hand overlap other parts of the hand. The second one is the difficulty that a large number of parameters have to be estimated from the derived projections, such as position and orientation of the hand itself, location of the fingertips, bending of fingers, etc. Technical difficulties are 1) uncontrolled environments, which have to be taken into account when locating the hand in the image and 2) processing speed, as a huge amount of data has to be processed [Erol et al. 2007].

To alleviate some of the difficulties, restrictions on the user or the environment can be applied by ensuring, for instance, that the palm is parallel to the image plane, which avoids overlapping of fingers, or by controlling the background in order to make the system fast and robust [Segen & Kumar 1998] [Segen & Kumar 2000] or using only a distinct aspect of hand movements for input (e.g. 2D position of the fingertip). However, those restrictions may not necessarily be of high inconvenience for the user. For instance, if the hand is typically parallel to the image plane in a given context of use, if background conditions can be easily controlled, or the derived aspects of hand movements are sufficient for the interaction they are used for. Following this, we will describe two sample solutions which do not track full 3D hand motion but still provide a highly usable basis for hand gesture interaction.

[Hardenberg & Bérard 2001] describe a system which applies computer-vision to facilitate non-contact hand gesture interaction. They detect the 2D position and 2D direction of user's fingers and associate them with the corresponding finger. They use this information along with the number of outstretched fingers as variables for defining hand gestures. Those hand gestures are then used to interact with three sample applications projected onto a wall, to control a presentation, move virtual items on the wall during a brainstorm session and virtually paint on the wall. In each sample scenario, users stand in front of the wall and interact with the projected application. Adaptive background models are used to ease the detection of fingertips in the images even with changing backgrounds originating from changes in the appearance of the graphical user interface and lightning conditions. They report a total latency of their finger finding algorithm of 26 – 34 ms (not including time needed for image acquisition).

Segen & Kumar [1998, 2000] describe a system designed to support tasks like 3D navigation, object manipulation and visualization. Their system detects the 3D position of the index finger- and thumb tip, and the azimuth and elevation angles of the finger's axis[1]. Based on this information they defined three hand gestures, a fourth gesture is included to serve as a default gesture for all other identified hand postures. The system is used in a desktop environment with two video cameras placed above a table. In order to make the system fast and robust, a high-contrast stationary background and a stable ambient illumination is required. A limitation of the

[1] *„The azimuth angle is the „left-right angle" measured in a horizontal plane and the elevation angle is the „up-down angle" measured in a vertical plane"* [Segen & Kumar 2000].

[2] See [IBM 1998] for videos of the system

system, which is mentioned by them, is the limited range of hand rotation due to the use of two cameras and their placement. However this can be compensated in adding video cameras for image capturing. Their system recognises the gestures and tracks the hand at a rate of 60 Hz (imposed by the video cameras used).

Those two sample systems provide valuable and excellent solutions for the setting they are aimed for. However, generalizing those approaches to other settings may be difficult, as a controlled background cannot always be guaranteed, holding the palm parallel to the image plane might not always be desired, or additional features of hand movements should be utilized for hand gesture interaction, e.g. using the rotation of the palm combined with a certain hand shape as an input parameter.

3.4.2 Glove-Based Tracking of Hand Movements

Besides computer-vision based, non-contact approaches, there are glove-based approaches for tracking hand movements.

Commonly used for tracking finger movements are commercially available data glove solutions, which build sensors into gloves to measure the bending of fingers capturing flexion, extension, adduction and abduction movements [5DT 2007] [Immersion 2008] (see Figure 10, left & middle). Data gloves with build-in sensors reliably deliver data on all possible finger movements and have the advantage that the quality of the data cannot be influenced by occlusion of fingers, or changing backgrounds. However, the gloves of data glove solutions typically come in only one size [5DT 2007] [Immersion 2008] [XIST 2006] and a good fit for each hand size cannot be guaranteed. A bad fit is not only able to disturb the user but can also influence the accuracy of the measured data if the build in sensors do not reflect the actual finger movements.

Data glove solutions typically provide high sampling rates, for instance 90 Hz for the CyberGlove® II or 75 Hz for the 5DT Data Glove 14 Ultra.

In order to track movement of the palm commercial data glove solutions can be combined with a tracking solution capable of detecting the orientation of an object. Therefore data glove solutions can be combined with an optical (e.g. [ART d] [Vicon 2009]) or electromagnetic (e.g. [Polhemus 2008]) tracking system. An optical tracking system uses multiple cameras to detect objects and calculates their position and orientation in reference to a predefined coordinate system. Such an object (typically consisting of a fixed arrangement of markers) has to be attached to the glove to monitor movements of the palm (e.g. at the back of the glove). Due to the use of cameras the reliability of the data on the movement of the palm is sensitive to occlusion. If the tracked object is occluded by other objects from the view of the cameras it cannot be detected. The user’s mobility range for accurate tracking depends on the amount of cameras used and their set-up. An electromagnetic tracking system detects a sensor which also has to be placed on the glove to monitor movements of the palm. The tracking system can provide information on the orientation and position of the sensor. Due to the fact that no cameras are used for tracking, occlusion of the sensor is not an issue. Electromagnetic systems can limit the range of the mobility range of the user in order to provide accurate tracking (a diameter of 2 meters in the case of the Polhemus system). The sampling rate depends on the tracking system used, for instance 50 Hz for the Polhemus system, 60 Hz for the optical tracking system developed by A.R.T. [ART b] and 120Hz for the Vicon tracking system [Vicon 2009].

Independent from the chosen combination of a data glove for tracking finger movements and an additional tracking solution for tracking palm movements, wearing a glove can encumber user interaction and give rise to hygienic problems if the same glove is worn over a longer period of time or by different users.

Figure 10: Left: the CyberGlove® II (taken from [Immersion 2008]). Middle: the 5DT Data Glove 14 Ultra (taken from [5DT 2007]). Right: The data glove of the A.R.T. finger tracking solution. Taken from [ART a]

A further glove-based solution has been developed by the company A.R.T. [ART a] [ART 2005]. It combines a minimized data glove (see Figure 10, right) with their optical tracking system. The data glove consists of a thimble set that can be attached similar to foxgloves onto the fingertips. The thimble set, available to either cover three or five fingertips, is connected to a target (an object consisting of a fixed arrangement of markers for which the tracking system can detect the position and orientation). This target is placed at the back of the hand. Markers, actively sending out infrared rays, are placed on the tip of the thimbles and onto the target. The optical cameras detect those rays and calculate the position of the thimbles (the position of the fingertips) and the position and orientation of the target (the position and orientation of the back of the hand). Therefore finger and palm movements can be tracked. From the tracked data the angles between the joints of the fingers and the orientation of the finger tips is derived. Fingers are identified via synchronized modulated flashes which synchronize the markers of the data glove with the optical tracking system.

Due to the minimized data glove which minimizes contact of the glove with the hand, hygienic problems arising for the previously described data glove solutions can be reduced. The design of the thimble sets, which are available in three different sizes, allows accustoming the data glove to a wide range of hand sizes. The markers on the fingertips are therefore always close to the fingertip whose position they measure.

However, due to the use of an optical tracking system occluded finger markers or target markers can impede tracking of hand movements which is not possible if the markers are not in the field of view of at least the number of cameras required for tracking.

The sampling rate for the information on palm movements (derived from the target) is 60 Hz. The sampling rate for the information on finger movements (derived from the thimbles) depends on the version (three vs. five thimbles) used: Sampling rate = 60 Hz / Number of thimbles. We are not aware that there currently is any other comparable system available.

3.4.3 Conclusion

Computer-vision based, non-contact solutions provide the opportunity of an unencumbered interaction. However, they are facing several difficulties when designing techniques aimed to reveal the real 3D motion of the human hand. Glove-based solutions provide high dimensional input with high sampling rates but trade it against intrusiveness.

Some researchers investigating hand gesture interaction techniques facilitate glove-based solutions, instead of purely computer-vision-based, non-contact tracking of hand movements, to explore advanced hand gesture interaction techniques before robust computer-vision-based, non-contact tracking of hand movements becomes widely available [Ni et al. 2008] [Vogel & Balakrishnan 2004] [Vogel & Balakrishnan 2005]. We adapt their idea and use glove-based solutions for tracking hand movements in this thesis, in order to ease the utilization of 3D motion of hand movements to design our hand gesture interaction techniques.

Out of the commercial data glove solution available, we consider the A.R.T. finger tracking solution to provide a good compromise between purely computer-vision-based, non-contact and data glove solutions. The least intrusive data glove reduces hygiene-related problems and those connected with bad fit present in the other data glove solutions, while providing high-dimensional output at a high-sampling rate. As a drawback, the A.R.T. finger tracking solution is sensitive to markers which are occluded from the cameras point of view, which reminds us of the self-occlusion problem purely computer-vision-based, non-contact approaches are facing. However, the A.R.T. finger tracking solution is insensitive against changing backgrounds.

3.5 Application areas for Hand Gesture Interaction

Hand gesture interaction can be found in various application areas, such as controlling home appliance, interacting in virtual and augmented reality, interacting with applications in a traditional desktop environment, interacting directly on interactive surfaces, and interacting from a distance with large displays. In this chapter we will present related work, the intension of applying hand gestures for a specific context and the impact on user interaction.

Controlling home appliance

Applying hand gestures for controlling home appliance is used to substitute for a physical remote device [Freeman & Weissman 1995], or to simplify the operation of multiple devices in avoiding the need of multiple physical devices for remote control [Lenman et al. 2002] [Tsukada & Yasumura 2002].

Tsukada & Yasumura [2002] describe how different home appliance devices can be operated from a distance with hand gestures. The movement of the finger is tracked by a mobile device they call “UbiFinger”. Users can point at the device they want to operate with the index finger. Once a device is selected it can be controlled from a distance using hand gestures that mimic hand movements used for operating the device directly (see Figure 11).

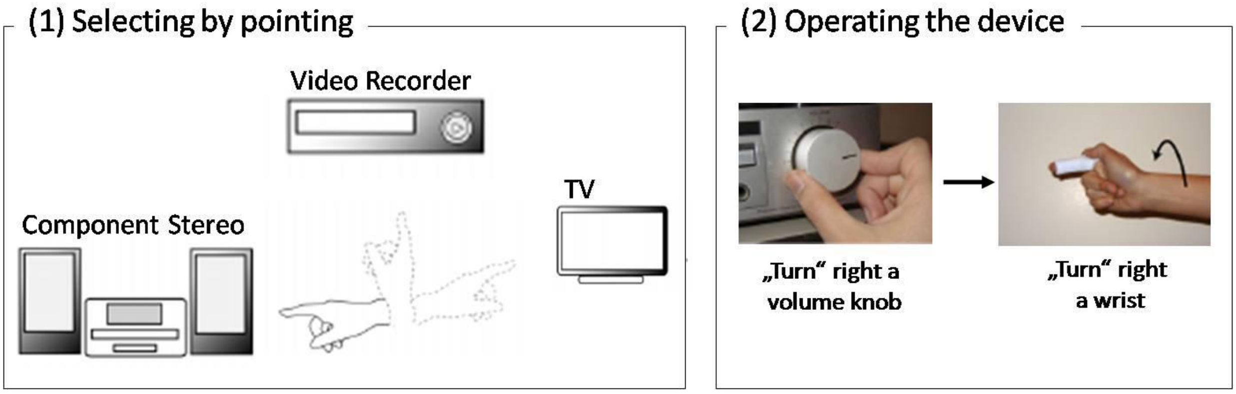


Figure 11: Interaction with the UbiFinger: Left: Selection of a device by poining. Right: Operating a selected device with hand gestures mimicking the corresponding real-world hand movements for directly operating the device (adapted from [Tsukada & Yasumura 2002])

They report that users found interaction techniques for device selection and operation easy, or even very easy to understand (except one user who found the techniques for operating the devices difficult to understand). The advantage of the "UbiFinger" interface is that users only have to learn one way how to operate a device and can use this knowledge for directly or remotely controlling it.

Interacting in Virtual and Augmented Reality

Hand gestures are attractive for the use in virtual and augmented reality as they provide the opportunity to reuse everyday hand movements [Buchmann et al. 2004] [Scheibe et al. 2007]. Therefore not only the appearance of a virtual object mimics real-world physical objects, but also techniques for object manipulation can be derived from manipulation of their physical counterparts. Therefore users can take advantage of pre-acquired skills and knowledge, which can reduce learning time and lets them concentrate on the task and not the input device.

Buchmann et al. [2004] describe an urban planning scenario, where users could interact with virtual objects in augmented reality, in using the corresponding real-world hand movements. For example users could replace buildings by grabbing them with the tips of their thumb and index fingers and move them to another location. Basing on informal user studies, the authors report that *"Many users were fascinated that they could manipulate virtual objects in the same way as real objects"* [Buchmann et al. 2004] and that most users found interaction easy and intuitive.

Interacting in desktop environments

In traditional desktop environments hand gestures have been proposed, with either the aim to perform point and click tasks and use the hand to replace the mouse as an input device [Kjeldsen & Kender 1996] [Wilson 2006] or with the aim to go beyond point and click interaction and make use of additional input dimensions that can be specified with hand gesture input for more expressive user input [Dhawale et al. 2006] [Segen & Kumar 2000] [Wilson 2006].

Wilson [2006] describes the TAFFI (Thumb and Fore-Finger Interface) prototype for hand gesture interaction in a desktop environment. TAFFI uses a camera mounted on the display to track users hand movements above the keyboard. Wilson argues that *"The space above the keyboard is a natural site for gesture input, as it enables nearly effortless switching between* *keying and gesture interaction, and allows for precise sensing of one or both hands"* [Wilson 2006]. Besides introducing hand gesture techniques for cursor control and for emulating mouse clicks, Wilson also proposes hand gesture techniques for navigation of aerial and satellite imagery. Therefore a "pinch" hand posture and hand movement is combined. For the "pinch" posture, users have to bring the tips of their thumb and index finger together. While maintaining the pinch posture users can (1) pan the view in moving the hand across the keyboard, (2) rotate the view by rotating the hand in the plane of the keyboard and (3) zoom by moving the hand up and down above the keyboard. Simultaneously panning, rotation and zooming of the view gives the interaction a fluid, direct manipulation feel, Wilson states.

While the techniques for navigation empower user input, it can be questioned if emulating the mouse in a desktop setting is particular beneficial for the user. Acquiring a mouse in a desktop setting might roughly take the same amount of time as lifting the hand to move above the keyboard. However, to operate the mouse the user can rest his forearm on the table surface, which could cause less fatigue than the corresponding hand gesture technique performed above the keyboard.

Interacting on Interactive surfaces

Hand and finger gestures are widely used for touch input on interactive surfaces. Although not enumerating all dimensions of hand movements, but only the contact of hand and fingers with a surface they provide a very appealing way of interaction.

Direct touch input with hand and fingers, for interacting with interactive surfaces is especially attractive for mobile devices, public places or collaborative settings, as it frees the user from having to carry a dedicated input device, for instance a stylus. The ability of users to easily use and combine multiple fingers for input (compared to use multiple mice or stylus simultaneously) has further been used to empower user input and create tense interaction techniques. It is worth mentioning some, such as performing a free rotation of an object by specifying the center of rotation with one finger and the rotation angle with movement of a second finger [Wu & Balakrishnan 2003], or scaling objects by specifying the scale factor with the distance between two fingers as suggested in [Wu & Balakrishnan 2003], or combining the two techniques to freely rotate and scale objects with movements of two fingers. A highly successful commercial product applying multi finger direct touch input for interaction with a mobile device is Apples' iPhone [Apple 2009]. Further example for the commercial use of single and multi finger touch interaction can be found on the Perceptive Pixel [Perceptive Pixel 2008] or Microsoft surface [Microsoft 2008] website.

Of particular relevance for our work is the use of hand gestures for distant interaction with large displays. Therefore we will describe related work from that application area in more detail in the following section.

3.6 Distant Interaction on Large Displays

The ability to use hand gestures as a mobile "input device" makes them not only attractive for direct touch input on interactive surfaces, but also for interaction from a distance with large displays. Taking advantage of this mobility, Vogel & Balakrishnan [2005] designed hand gesture interaction techniques for distant pointing and clicking on very large high-resolution displays. Baudel & Beaudouin-Lafon [1993] designed hand gesture interaction techniques for controlling a presentation projected onto a wall, where the expressiveness of hand gesture input has been used to create a terse and powerful interaction. Similar to the combined use of hand movements and spoken utterance evident in human-human communication [Kendon 2004], researchers also combined hand gestures with spoken commands for multimodal input for distant interaction with an application displayed on a large wall [Bolt 1980][Lucente et al. 1998].We will describe those systems in detail in the following sections.

Vogel & Balakrishnan's distant pointing and clicking techniques

Vogel & Balakrishnan [2005] proposed to use hand gestures for distant pointing and clicking on large high-resolution displays. They designed and evaluated three combinations of pointing and clicking hand gesture interaction techniques (see Figure 12).

To track hand movements, passive reflective markers have been attached to the index finger, ring finger, thumb and the back of the user's hand. The position of those markers has been tracked by the optical tracking system Vicon [Vicon 2009 b], which streamed the tracked data to other applications for further usage.

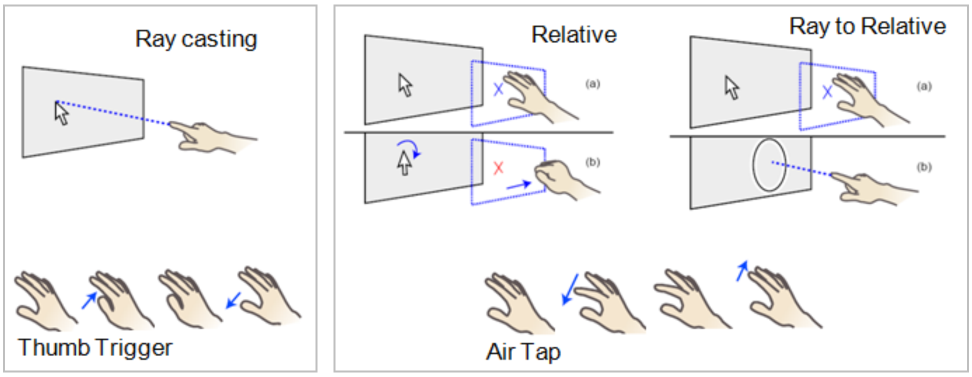


Figure 12: Three combination of point and select gestures. The top column shows the applied pointing gesture and the mapping of hand to cursor movement, the bottom column shows the gesture coupled with the above pointing technique. Adapted from [Vogel & Balakrishnan 2005].

For **pointing**, they designed three techniques, where they varied the hand postures used and the mapping from hand movement to cursor movement.

The "RayCasting" technique combines an extended index hand posture and an absolute mapping of hand movement to cursor position (see Figure 12, left). To position the cursor, an imaginary ray emerging from the tip of the index finger is intercept with the display, and the cursor is placed at the point of interception. The use of an extended index gesture for pointing is motivated by Kendon [2004], who identified pointing with the extended index finger, as being one of seven distinct pointing gestures used in human-human communication. According to Vogel & Balakrishnan [2005], using the extended index finger hand posture with an absolute mapping to cursor position based on an imaginary ray emanating from the tip of the index finger, results *"[...] in a pointing technique that it is arguably natural and conceptually simple"* [Vogel & Balakrishnan 2005].

In the "Relative" technique a "safe hand" posture is used for pointing (see Figure 12, in the top column of the Relative technique). This hand posture is derived from the thought of holding and using an invisible mouse. The "safe hand" posture is combined with a relative mapping of hand movement in a vertical plane, to cursor position. Clutching, (= disengaging the display cursor from hand movement to reposition the hand) can be performed in forming a fist with the pointing hand. Once the "safe hand" posture is adopted again, the cursor can be moved again.

The “RayToRelative” technique combines the “Relative” technique with the absolute “Ray Casting” technique. Absolute “RayCasting” can be used to do rapid coarse grain pointing. During “RayCasting” the cursor is replaced with a circle, placed at the interception of the ray emanating from the index finger with the display. The “Relative” technique is used for direct cursor control, similar to the purely “Relative” technique. The “RayCasting” technique thereby substitutes for clutching with the fist, to reposition the hand and simultaneously move the cursor near the desired target.

For **clicking**, they combined “RayCasting” with a “thumb trigger” hand gesture. For the “thumb trigger” the thumb of the pointing hand was moved towards and away from the side of the hand. The other pointing techniques (“Relative”, “RayToRelative”) are combined with an “Air tap” hand gesture where the index finger has to be moved down and up, imitating the movement conducted when clicking a left mouse button. Different to the “Air tap”, the “thumb trigger” provides implicit kinesthetic feedback as the thumb can touch the side of the hand, and an absolute down position. Both hand gestures are accompanied by acoustic and visual feedback to indicate when the hand posture is entered and exited, respectively a click is detected. Although the “thumb trigger” provides more feedback, which might support users while performing the hand gesture, early user tests revealed that the “thumb trigger” was found uncomfortable and tiring. We suspect that the reason therefore might be higher tension and unfamiliarity with the “thumb trigger” gesture. Users quite often move the index finger up and down, for example when typing on a keyboard or grasping physical objects, however moving the thumb to touch the inside of the hand is typically not performed very often. Both higher tension and unfamiliarity of the “thumb trigger” hand gesture might be the reason for the negative rating by the users when compared to the “air tap” hand gesture.

To compare the three point & clicking hand gesture interaction techniques, Vogel & Balakrishnan conducted a formal evaluation study. 12 participants have been recruited to perform simple point and selecting task on a large high-resolution display, while standing at a stationary position four meters away of the display. They applied a repeated measures within-participant factorial design, with the independent variables technique (Relative, RayToRelative, and RayCasting), distance between targets (4020mm, 2680mm, 1340mm) and target width (144mm, 48mm, 16mm). Vogel & Balakrishnan analyzed error rate, task completion time and recalibration frequency.

Results showed that “RayCasting” combined with the “thumb trigger” was faster than the other techniques when selecting large targets, or when clutching would have been required. However it also revealed high error rates, especially for small targets (width = 16 mm). The other two pointing & clicking interaction techniques showed no significant differences considering selection time or error rate. However, time needed for clutching introduced an overhead, which caused the “RayCasting” to be faster for large distances. Subjective ease of use votes of the users places “RayCasting” last with only 1 vote, compared to 5 for “RayToRelative” and 6 for the “Relative” technique.

From the results reported by Vogel & Balakrishnan, it is evident that the main issues of the absolute “RayCasting” technique are the low accuracy for small targets and the lowest ease of use score. The low ease of use score may reflect the higher tension in the hand evident for the extended index hand posture, and its inaccuracy, when compared to the other pointing techniques. The issue of high tension could be addressed in using a different hand posture, which requires less tension. Despite this drawback of the “RayCasting” technique, its absolute cursor mapping has put it prior to the other techniques, when clutching actions would have be necessary to overcome large distances. Considering large high-resolution displays, where targets can be positioned anywhere and overcoming large distances with the cursor is highly likely, a technique which does not introduce an overhead resulting from clutching actions would be desirable to decrease movement time.

Baudel & Beaudouin-Lafon's Charade System

A system aimed to explicitly support presentation scenarios with hand gesture input is "Charade" described by Baudel & Beaudouin-Lafon [1993]. A wired data glove, worn by the user is used to track hand movements.

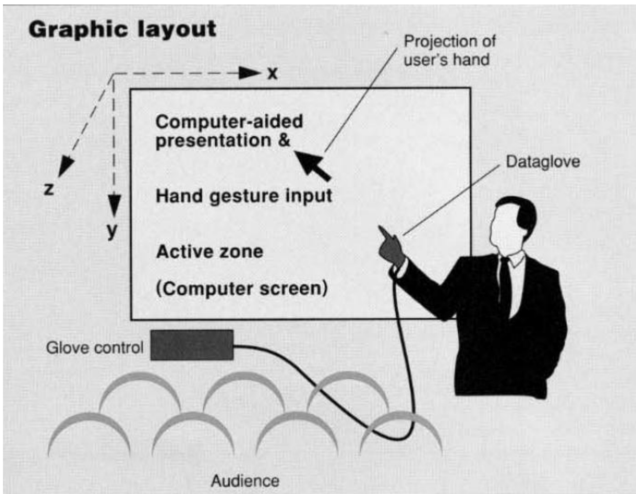


Figure 13: Schematic set-up of the "Charade" System (taken from [Baudel & Beaudouin-Lafon 1993]).

The area where the contents of the presentation are displayed is labelled as the active zone (see Figure 13). If the pointing direction of the hand intercepts the active zone a cursor is displayed following the movement of the hand. Hand gestures can be used to control the presentation, when the hand points towards the active area. Hand movements performed while the hand does not point towards the active area are ignored. This allows unconstrained gesticulation of the speaker when the hand is directed away from the presentation screen.

Baudel & Beaudouin-Lafon defined 16 different hand gestures, which the user can perform to control the presentation. Each hand gesture starts with a hand posture defined by the wrist orientation and finger bending, followed by a dynamic phase where the hand is moved, and ends when the active area is exited or a predefined hand posture, again defined by the wrist orientation and finger bending, is adopted (see Figure 14 on page 24 for the complete set of hand gestures and the action they invoke).

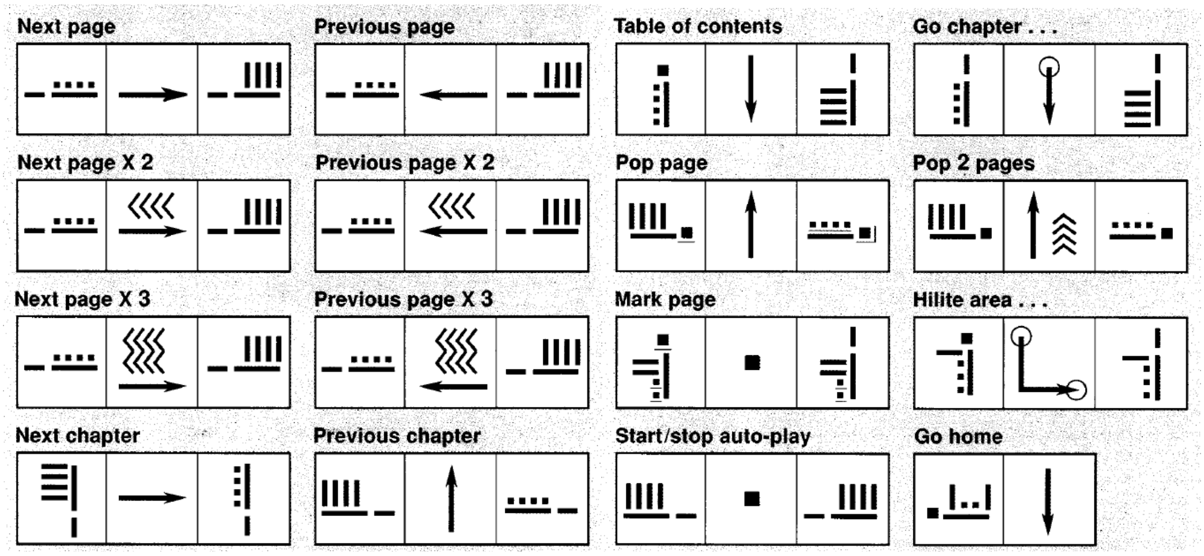


Figure 14: Set of Hand Gestures and their mapping to actions in the "Charade" system. Taken from [Baudel & Beaudouin-Lafon 1993].

The authors suggest to use an appropriate mapping of hand gestures to actions they invoke, for example moving upwards for issuing a "move up" command, or natural gestures associated with the task. For tasks without any naturally associated hand movements, for instance "changing fonts" or "save", they suggest to use speech to complement hand gesture interaction.

Compared to the previously described point & clicking hand gesture interaction techniques, the hand gesture sets found in "Charade" apply a more arbitrary mapping of hand movements onto the action they invoke. Although the hand movement used for the dynamic phase of some gestures mimic the action they invoke (for example moving up to go to the top page), the hand postures associated for the beginning and end of a gesture set are rather arbitrary, considering the mapping to the invoked action. Different to this, Vogel & Balakrishnan derived their hand gestures (except the "thumb trigger" and the "fist" for clutching) from hand movements evident in human-human communication or operating a mouse. They used the hand movements for similar tasks, hence applied a similar mapping onto the action the hand movement invokes (e.g. using pointing with the index finger as a hand gesture for a pointing task). However, Baudel & Beaudouin-Lafon do not report that users were in any way impeded by the more arbitrary mapping of hand postures to actions but quickly learned the gestures and found the interface ease to use. Baudel & Beaudouin-Lafon took advantage of the different shapes the hand can take on and movements which can be performed. They combined wrist orientation and finger bending, to define 16 gesture commands making use of the hand as a powerful input device for the Charade scenario.

Bolt's "Put-That-There" Interface

The "Put-That-There" interface, described in [Bolt 1980] combines spoken commands with hand pointing gestures for distant interaction with a large display. Users can create, move and manipulate different shaped objects, e.g. circles, squares, triangles or squares, on a large display while seated at a distant position in front of the display (see Figure 15).

Similar to human-human communication, users can talk and simultaneously point towards a location. Users can use spoken commands only or optionally combine it with hand pointing gestures. The information where users are pointing to is then combined with the spoken commands. For instance, to move a blue triangle users can simply say “Put the blue triangle to the right of the green square”. When combining speech with hand gesture pointing, users can say “Put that there” instead and point at the blue triangle while saying “that” and point at the targeted location while saying “there”. The locations which are being pointed at while saying “that” and “there” define the spatial parameters for the command.

To track the orientation of the palm or fingers Bolt used a system called “ROPAMS” (Remote Object Position Attitude Measurement System). Therefore users have been equipped with a small magnetic sensor (a cube with 0.75 inches on edge), which was connected to a transmitter cube via a small cord. This sensor could for instance be wrist-mounted or worn as a finger ring. The ROPAMS could determine the three dimensional orientation of the sensor in space and its distance to the transmitter. Information which could then be used, to determine the location on the display the user points at. A small white “x” cursor on the display provides visual feedback on the exact location which is being derived from the pointing gesture.

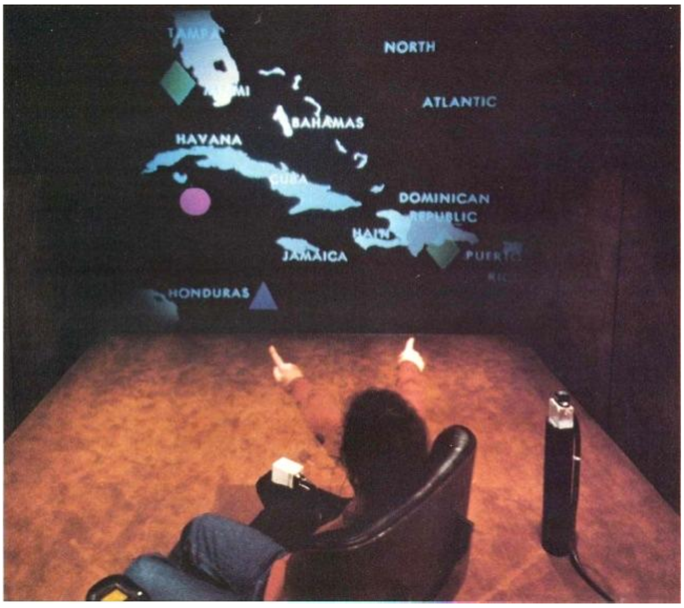


Figure 15: A user seated in front of a large display, while moving objects located on a map of the Caribbean combining hand pointing gestures and speech input. Taken from [Bolt 1980].

Users could also change the properties of existing objects, such as the colour or the size. Again, hand pointing can be used to specify the location of the target object, whereas all other information is given by means of speech. Objects can be of three different sizes – small, medium or large. When no size is determined upon creating an object the default size is medium. The size of an object could be changed with the command “make that (while simultaneously pointing towards the desired object) smaller”. Making an object smaller causes the object to become one size smaller as its current size.

Different to “Charade” and the point & clicking techniques proposed by Vogel & Balakrishnan, Bolt uses hand gestures only to specify spatial parameters. The action which should be performed (e.g. “create”) and additional parameters (e.g. “create a blue square”) are specified with speech input. Therefore the hand gesture set consist of only one pointing gesture and user input is empowered by the use of spoken commands. “Put-That-There” illustrates the opportunity to combine hand gestures with speech input to utilize a natural way of interaction.

Lucente et al.'s Visualization Space

Whereas Bolt [1980] uses hand gestures only for spatial reference and manipulates properties of objects with spoken commands, Lucente et al. [1998][2] extend the capabilities of hand gesture input in their "Visualization Space" system. Similar to "Put that there", speech is combined with hand gesture input for distant interaction with a large display. Users can specify locations with pointing gestures to complement spoken deictic words (e.g. "there" or "that") to perform object manipulation tasks.

In the "Visualization Space" system, hand gestures can not only be used to indicate a two dimensional location on the display, but also to specify further parameters, such as the size of objects or the angle of rotation. For instance, the user can select an object by pointing at it and say "Select that" and change its size in further issue the command "Make it this big" accompanied by a two handed gesture, where the hands are held apart and the distance between them specifies the desired size. Further examples are commands like "Rotate it like this" where the rotation is defined by a movement of both hands and finished with a command such as "Leave it like this".

Lucente et al. call their system *"[...] a deviceless descendant of the Put That There System"* [Lucente et al. 1998]. Unlike Put-That-There, where users have to wear a sensor in order to determine the location which is being pointed at, Lucente et al. track users' movements with video cameras and do not attach a physical device to the user.

Both systems make use of every day hand movements and their combination with spoken utterance, for instance pointing gestures to specify a location or holding two hands apart to specify the size of an object ("The fish was *this* big"). While Bolt takes only advantage of natural pointing gestures, Lucente et al. extend the use of hand gestures to specify further parameters. Considering the example of changing the size of objects: the Put-That-There interface provides three discrete values (small, medium, and large), which limits the options available to the user for interaction. Defining the size of an object with a movement of two hands, as in the "Visualization Space", makes a continuous range of values available for the user. Depending on the task either one of the two options for input might be more suitable. If the exact size of an object is not important, spoken discrete commands are fast and easy. However if the exact size should be defined, hand gesture input can complement spoken commands and fine tune, in this example, the size of an object.

"Visualization Space" and Bolts' "Put-that-there" illustrate that hand gestures can be a valuable enhancement for speech input. By taking advantage of what both input modalities are best at, a natural and intuitive way of interaction can be realized.

Conclusion

The presented related work on hand gesture interaction techniques for distant interaction at large displays, has shown that hand gestures are applied in very different ways, ranging from using them to indicate a two dimensional location, up to defining a large set of gestural commands to perform different tasks.

However, despite this different usages, related work proves that hand gesture interaction does not only meet the mobility requirement [Vogel & Balakrishnan 2005] but can also offer a very natural [Bolt 1980][Lucente et al. 1998] and expressive way of interaction [Baudel &

[2] See [IBM 1998] for videos of the system

Beaudouin-Lafon 1993]. Also differently, researchers have taken advantage of the prospects of hand gesture input, namely naturalness, expressiveness and mobility.

4. Hand Gesture Recognition

This chapter describes the recognition of static hand gestures. At the beginning a distinction between static and dynamic hand gestures will be drawn. Next we will describe our approach for static gesture recognition. To compensate for missing input data from the accompanied hand tracking approach, a "gesture memory option" is described as an additional feature of our gesture recognition approach.

4.1 Static and Dynamic Hand Gestures

Hand gestures can consist of hand postures and/or hand movements.

Hand postures are defined by static finger postures, in which only the position of the finger and the joint angles between finger bones are relevant, but not the location of the hand or movement of the palm and fingers over time. Examples for hand gestures falling into this category are forming a fist or stretch out the index finger, while the others are curled to perform a pointing gesture.

Hand movements are defined by movement sequences of fingers or hands, hence variations over time are a characteristic feature. Examples are: 1) waving one hand and 2) clapping two.

A hand gesture can be defined by a hand posture, by a dedicated hand movement or by a combination of hand postures and hand movements, for example if hand movements have to be performed, while the hand maintains a specific hand posture.

For the purpose of this thesis, hand gestures which are defined by hand postures are called **static hand gestures** and hand gestures which contain hand movements are called **dynamic hand gestures**.

4.2 Applied Approach for Static Hand Gesture Recognition

For our hand gesture interaction techniques we need to be able to recognize static hand gestures (e.g. the "pinch" and "extended index" gesture, see chapters 5.2, 7.1.2 and 7.2.2). The recognition of dynamic hand gestures is not necessary for our proposed gesture interaction techniques. Therefore we did not implement a recognition approach for dynamic hand gestures. However, if the recognition of dynamic hand gestures should be an upcoming requirement for future extensions of our set of hand gesture interaction techniques, using Hidden Markov Models might be a suitable approach, as they are widely used to model time-varying signals and recognize observed sequences [Wilson 2007] e.g. in [Nickel & Stiefelhagen 2003] [Starner et al. 1998] [Wilson & Bobid 2000] [Zobl et al. 2003].

To recognize static hand gestures we identify characteristic features for each one of our gestures independent from the format in which tracking solutions report on hand movements (e.g. contact between the tip of the index finger and thumb). We then map those features onto geometrical metrics based on the input data delivered by the applied tracking solution (e.g. angular data on bending of the fingers or 3dof data on the position of fingertips). Identifying features independently from the tracking solution delivered, allows to use the same (semantically) description of gestures for different tracking solutions and to describe those features to users without having to explain technical details. We then define thresholds for the geometrical metrics used to determine whether a gesture is performed or not. For defining those thresholds we inspect and analyse samples of the gesture under scrutiny. We then recognize each gesture in applying an algorithm, based on state dependent comparison of thresholds (described below).

We model and recognize each gesture explicitly, similar to the approach described in [Dhawale et al. 2006]. Dhawale et al. track bare hands with a single camera located above the hand and use hand gesture input for interaction with applications in a desktop setting. They recognize several gestures, such as a horizontal flip, a vertical flip or a fist (see Figure 16). The fist, for instance is distinguished from a flat hand in comparing the narrowest part at top of a hand with the widest part of the hand. A large distance between the two indicates a flat hand, as the narrowest part origins from a fingertip, whereas a small distance indicates a fist.

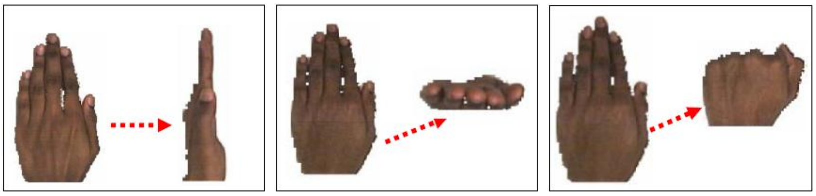


Figure 16: Sample hand gestures recognized in Dhawala et al. [2006]. Left: horizontal flip. Middle: vertical flip. Right: fist. Taken from [Dhawala et al. 2006].

A drawback of our approach of modeling and recognizing each gesture explicitly is that new gestures cannot automatically added to the set of static hand gestures which can be recognized. Using an approach which combines a clustering with a classification algorithm would provide an approach where new gestures can be added automatically. The clustering algorithm could be applied to automatically identify the characteristic features of each gesture and distinguishing the gestures from each other, whereas the classification algorithm could be used to classify hand movements based on labeled samples of the gestures (e.g. provided by the clustering algorithm).

Due to the small number (2) of static hand gestures needed to implement our hand gesture interaction techniques, we considered the drawback of not being able to automatically add new static hand gestures acceptable. However, if for future work it is expected that the number of static hand gestures significantly increases an automatically approach might be better suited.

An advantage of our algorithm for gesture recognition is that we can easily incorporate state dependent thresholds for gesture recognition, resulting in different recognition behaviour depending on the fact if users are exiting or entering a static hand gesture. This allows us to compensate for unwanted gesture recognition due to natural hand tremor or tracking inaccuracies.

The idea of the applied approach in this work for the recognition of static hand gestures can be broken down into five steps:

1. Identification of the characteristic features of a static hand gesture
2. Mapping of those features to a numerical metric and a corresponding threshold
3. Identification of the most characteristic metric
4. Determining the recognition thresholds

5. Determining a “follow-up-recognition” threshold for the most characteristic metric

The methodology and rationales applied in the gesture recognition process will be described and illustrated at the example of the “extended index” hand gesture (see Figure 18(a)).

Identification of the Characteristic Features

For the “extended index” static hand gesture, the index finger is outstretched and the middle, ring and index finger are in a relaxed bent position. Hence an outstretched index finger and a bent middle finger can be defined as characteristic features.

Mapping of Features to Numerical Metrics

The characteristic features of the static hand gestures have to be described in a way that makes it possible to derive them from the available input data for the gesture recognition.

Data glove solutions, commonly used for tracking finger movements, typically deliver angular information on the bending of fingers (e.g. CyberGlove II®) or 3 dimensional information on the position of the fingertip situated in a local “hand coordinate system” (e.g. A.R.T. finger tracking solution). Associated with those measured values is information on the sensor, that is, the finger of its origin.

For the following explanations, we proceed with the assumption that gesture recognition is combined with a data glove solution delivering 3 dimensional position data of the fingertips, which are located in a “local hand coordinate system”, as illustrated in Figure 17. We focus on those kinds of input data, to provide a clear example for the rationales applied.

Our method, however, is not restricted to those kinds of input data, but can be applied for any kind of input data reflecting movement and position of fingers with numerical measures and providing an association with the origin of the data (the finger where the data originates from).

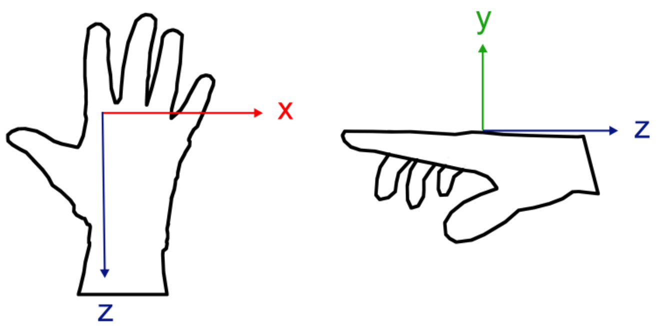


Figure 17: Coordinate system of the 3d fingertip positions, used to illustrated the gesture recognition process

Mapping of the Features to a Numerical Metric and a Corresponding Threshold

An outstretched index finger can be mapped onto the numerical metric “distance of the index finger from the x-z-plane” and a corresponding recognition threshold. A bent middle finger can be mapped onto the numerical metric “distance between the index and middle finger” and a corresponding recognition threshold (see Figure 18 (a)).

Identification of the most Characteristic Metric

To identify the most characteristic metric the changes in values for the chosen metrics, derived from a movement of the hand from a relaxed hand posture to the static hand gesture, have to be analysed. The metric whose corresponding values reveal the largest variance is then considered the most characteristic one, as its values are most affected when the static hand gesture is taken on. If the variance is similar for all metrics, the metric can be defined based on analysing semantically the static hand gesture as being the metric which most likely describes the users' intend when performing the gesture. For instance, in our example the metric onto which the outstretched index finger is mapped can be considered as the most characteristic one, as the user aims at extending his index finger and not at bending his middle finger.

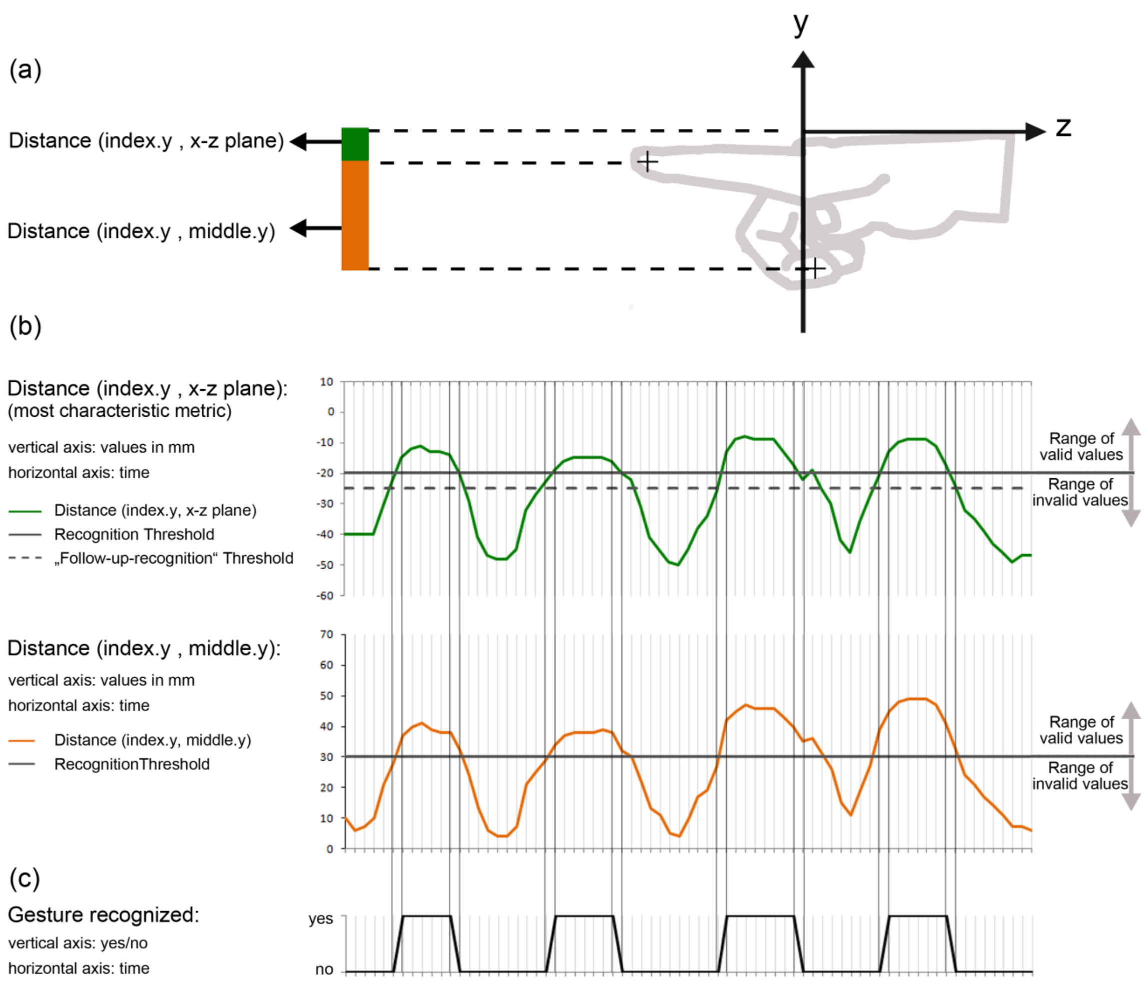


Figure 18: (a): The extended index gesture and its characteristic metrics. (b): Thresholds and schematic values of the characteristic metrics simulating a user continuously entering and exiting the gesture (c) outcome of the gesture recognition process

Determining the Recognition Thresholds

A recognition threshold is a numerical value, which divides the range of values along the corresponding metric into two distinct ranges: 1) a range of values, which can be observed when the static hand gesture is maintained, called the "range of valid values" and 2) a range of values which can be observed when the static hand gesture is not maintained, called the “range of invalid values” (see Figure 18 (b) on page 31). To recognize a static hand gesture the values of all its associated metrics have to fall within the “range of valid values”.

The recognition thresholds are defined based on sample data derived from a user that continuously changes from a relaxed hand posture to the inspected static hand gesture, e.g. the “extended index” gesture in our example. The sample data are thereby generated by the tracking solution which should be combined with the gesture recognition. In Figure 18 (b) on page 31 the values of the metrics of the “extended index” gesture are shown, observed while a user performed four “extended index” gestures. High values for both metrics indicate that during those frames the “extended index” gesture was maintained.

The recognition thresholds should be chosen so that within this training data each “extended index” gestures, but none of the other hand postures, are classified. In order to achieve this, the weakest gesture has to be identified as the gesture with the smallest maximum values for the characteristic metrics (in our example: “extended index” gesture no. two in Figure 18 (b) on page 31).

The recognition threshold for the most characteristic metric should be defined in a way that the weakest gesture just gets classified, with an added tolerance range to compensate for hand tremor and tracking inaccuracy. Therefore the recognition threshold for the most characteristic metric is the highest value for the “weakest” gesture plus a tolerance range. The recognition thresholds for the other metrics should be positioned farer away from the highest value of the weakest gesture (also including the same tolerance range). As a result, the recognition threshold for the most characteristic metric is the most restrictive one and should be the last one to be crossed when the static hand gesture is entered, respectively the first one to be crossed, when the static hand gesture is exited again (the chosen recognition thresholds for the “extended index” gesture and the outcome of the gesture recognition process are illustrated in Figure 18 (b) + (c) on page 31).

That the recognition threshold for the most characteristic metric is the first threshold that is being crossed when exiting the gesture, is a necessity for the mechanism applied to avoid unwanted recognition of the gesture while the static hand gesture is exited. The mechanism applied to avoid such false positives is based on a “follow-up-recognition” threshold and described in the following section.

Determining a “Follow-up-Recognition” Threshold for the most Characteristic Metric

If a static hand gesture is exited, slight fluctuations of the values for the characteristic metrics around the recognition threshold, due to hand tremor or tracking inaccuracies, can cause unwanted gesture recognition (as it is the first threshold crossed and the values of the other metrics are still within the “range of valid values”). As the user intends to exit the static hand gesture, such a false positive should be avoided because it does not match the users’ intention. Note that we define the threshold for the most characteristic metric to be the first on to be crossed on purpose. In doing so, we achieve a deterministic behaviour which we can use to avoid false positives due to slight fluctuations in the tracked values.

To avoid such false positives, additional to the recognition threshold a “follow-up-recognition” threshold has to be defined for the most characteristic metric. This “follow-up-recognition” threshold is determined in adding a tolerance range to the value of the recognition threshold of the most characteristic metric. The same tolerance range as the one used when defining the recognition threshold can be used, as the same rationale applies (= to compensate hand tremor and tracking inaccuracies).

This “follow-up-recognition” threshold has to be crossed when a static hand gesture is exited before the same gesture can get recognized again. Note that the “follow-up-recognition” threshold does not enlarge the “range of valid values” defined by the classification threshold.

The result of these two thresholds for the most characteristic metric is that a static hand gesture has to be left explicitly and hand tremor or tracking inaccuracy do not trigger unwanted gesture recognition while a gesture is exited. As a necessity, it has to be ensured that from the number of recognition thresholds for a static hand gesture, the recognition threshold of the most characteristic metric is crossed first when the gesture is exited. Otherwise slight fluctuations around other recognition thresholds could trigger false positives, as those fluctuations are not compensated by a corresponding “follow-up-recognition” threshold.

The chosen “follow-up-recognition” threshold for the “extended index” gesture is illustrated in Figure 18 (b) on page 31.

The described approach enables a fast recognition of static hand gestures, as only a fixed number of values for predefined metrics (two in the case of the “extended index” gesture) have to be calculated and compared to predefined recognition thresholds and a “follow-up-recognition” threshold.

4.3 Compensating Missing Values in Input Data

This approach to recognize static hand gestures can be used with input data originating from different solutions used for tracking hand movements. Solutions which rely on optical sensing techniques are sensitive against occlusion of markers or fingers which can result in missing values. Those missing values can impede the recognition of gestures, which might then impede user interaction if no compensation is provided by the system.

For the purpose of this thesis, we used input data delivered by the A.R.T. finger tracking solution and Whitey, a novel data glove solution which we will describe in chapter 6.

Both glove-based hand movement tracking solutions use optical cameras to detect the position of the fingertips by identifying markers attached to a glove. If those markers are not visible for the cameras, no information on their position can be delivered. To compensate missing values for finger positions in the input data, we introduced a gesture memory option which can be activated for gesture recognition.

Gesture Memory Option

If data of fingers needed for the recognition of a static hand gesture are missing, the memory option checks if this specific gesture has been recognized in the last frame where data for the finger was present. If so, it is assumed that although the finger is currently not traceable, the gesture is still maintained and hence the gesture gets recognized again.

This is similar to an idea proposed by Letessier & Bérard [2004]. They describe a system where a purely computer-vision-based, glove-free approach is applied to track hands in front of a large surface. If a finger is no longer detectable, they consider a certain time window before the finger gets reported as having disappeared. If the finger becomes visible again during the time window the tracked position is further used and no disappear event is generated.

Different to Letessier & Bérard [2004], which use a fixed sized window, for the gesture memory option the user can specify how long a gesture can be reused while missing values for the finger are present in the input data. If the gesture has been reused for the specified maximum duration, the gesture will no longer be recognized until the fingers are visible again. To minimize the disturbance of the user in such cases, the “follow-up-recognition” threshold (see chapter 4.2) can be deactivated. This way the user does not have to explicitly exiting the gesture to cross the “follow-up-recognition” before the gesture can be recognized again, but can instead maintain the gesture while optimizing the visibility of the markers until the gesture gets recognized again.

The gesture memory option is of particular importance for interaction techniques where maintaining a static hand gesture is mapped to a continuous action. For instance when mapping the “pinch gesture” to a dragging action, as described in chapter 7.1.2. If the gesture recognition fails while the user is still maintaining the static hand gesture, the dragging task would be interrupted by the system and the users’ attention would be drawn away from the task.

5. Hand Gestures as a Pointing Device

Physical navigation at LHRDs substantiates the need for an input device which allows interaction from any point and distance to enhance physical navigation and not impede a fluid human-computer interaction. Hand gestures as an input device can fulfill this mobility requirement (see chapter 2.4).

With hand gestures as an input device, aimed to support human-computer interaction on LHRDs in general and not targeted at a specific application, the question arises, which interaction tasks should be supported. Casiez et al. [2008] state that *"Pointing at a target is a fundamental and frequent task in graphical user interfaces (GUIs) […]"*. Accot & Zhai consider that pointing, which they describe as *"[…] moving a cursor into a graphical object with an input device and clicking a button" "[…] undoubtedly remains the most universal interaction paradigm across diverse application domains and contexts"* Accot & Zhai [2002]. Those two statements indicate that pointing and selecting (if we consider clicking a button as a selection) are common basic tasks in many of today's graphical user interfaces. Therefore we designed interaction techniques for those two basic and commonly found interaction tasks.

In the following chapter we will describe our techniques for pointing and selecting. Furthermore, we will discuss aspects which could influence the performance of hand gesture interaction, namely the influence of limb segments and additional tactile feedback. A formal evaluation study, aimed at accessing the usability of our techniques and the influence of tactile feedback and movement direction (resulting in differences in movement of limb segments) will be described. The outcome of this study will be presented, discussed and recommendations for future work will be proposed.

5.1 Pointing "Palm Pointing"

Parts of this chapter have been published in [Foehrenbach et al. 2008], but have been further enhanced for this thesis.

Kendon [2004] describes a variety of every day gestures which are used in combination with speech. These kinds of gestures are interesting for human-computer interaction, as they are already known by the potential users and could therefore lead to a decreased learning effort and a better recall when used for interaction. In the context of human-computer interaction pointing is used to specify a location on a display with the user facing it; hence gestures which are used in this manner should be used.

Kendon [2004] identifies seven pointing gestures (also referred to as deictic gestures), which are used in combination with speech in human-human communication. Those gestures can be classified into three categories: pointing with the extended index finger, pointing with the open hand and pointing with the thumb (see Figure 19, on page 36). All categories share the same semantic meaning, where

> *"Pointing gestures are regarded as indicating an object, a location, or a direction, which is discovered by projecting a straight line from the furthest*

point of the body part that has been extended outward, into the space that extends beyond the speaker" [Kendon 2004].

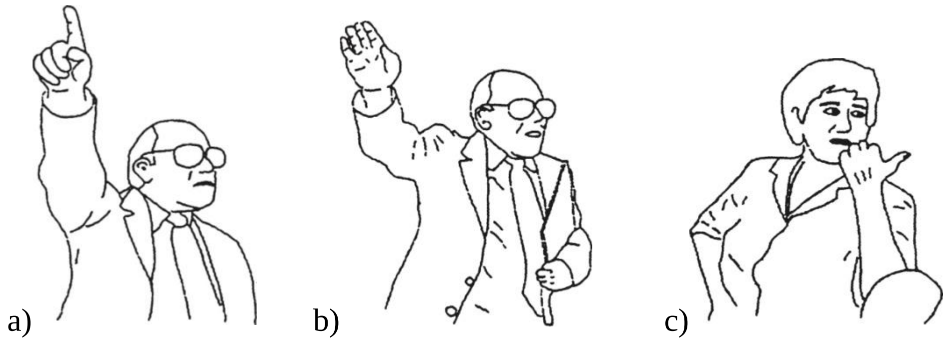


Figure 19: Pointing a) with the extended index finger, b) with the open hand and c) the thumb. Taken from [Kendon 2004]

Whereas pointing with the thumb is not used for objects, locations or directions in front of the speaker, the remaining two categories are used in this manner. However, even if both gesture categories share the same semantic theme the usage is slightly different. An extended index finger is used when one specific object or location is referred to, whereas pointing with the open hand indicates that the object is related to the topic but is not explicitly mentioned.

The exact location of a specific object is what users aim for, when positioning the cursor over a target, which describes the usage of the extended index in pointing. Even if the semantic meaning would be identical, the extended index gesture bears some drawbacks when used for human-computer interaction. Vogel & Balakrishnan [2005] evaluated three combinations of point and click hand gestures and found that pointing with the extended index finger showed the highest error rate and the lowest ease of use score (1 out of 12). Another drawback is that pointing with the extended index finger requires higher tension than pointing with the open hand. These drawbacks discourage the usage of the extended index gesture for pointing. Pointing with the open hand requires less tension, which makes the open hand gesture a better candidate considering biomechanical load in this comparison. The usage would also resemble every day gesticulation, not as much as the extended index, but considering the discussed issues of both gestures, the open hand seems to be the best choice for being used as a pointing gesture.

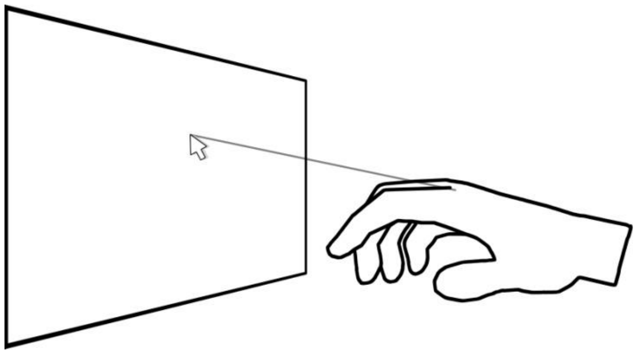


Figure 20: The "palm pointing" gesture used for pointing

We therefore use the open hand pointing gesture with an absolute mapping for cursor positioning. A straight line, defined by the orientation of the palm is projected and intercepted with the display. The display cursor is placed at the point of interception (see Figure 20). This is in line with Kendons' regard on pointing gestures described above. Practical experience with the introduced hand gesture pointing technique has revealed that using the position above the joint connecting the index finger with the palm as a starting point for the projected straight line leads to a cursor position which is intuitively expected by the user.

The display cursor is always in line with the pointing hand, and loosing track of the cursor, as reported of being a usability problem with a mouse (using a relative mapping for cursor positioning) at large displays by Robertson et al. [2005], can be prevented.

We call the combination of pointing with an open hand and an absolute mapping of the orientation and position of the hand, as described above, "palm pointing".

5.2 Selecting "Pinch to Select"

Parts of this chapter have been published in [Foehrenbach et al. 2008], but have been further enhanced for the purpose of this thesis.

Additional to pointing, we also want to support selecting with hand gestures. Ideally, such a selection gesture should fit well when used in combination with the pointing gesture and should furthermore be already well known from every day gesticulation or similar every day "selection" actions.

Besides pointing gestures, Kendon [2004] also describes a so called "R-Family" of "precision grip" gestures that are used when the speaker wants to be very exact and precise about something and therefore special attention is needed. Selecting in the context of human-computer interaction shares this meaning, as the user wants to exactly select one specific object of the application.

Figure 21: Precision grip gesture. Taken from [Kendon 2004]

When performing a gesture of the "R-Family" the tips of the index finger and the thumb are brought together to form a shape that resembles a circle or a ring (see Figure 21), a finger movement that can be used in combination with the selected pointing gesture quite well. Yet another advantage is that the movement of the gesture mimics the action of doing a left mouse click with every computer user being familiar with. It also provides implicit touch feedback due to the fingertip contact signalling that the gesture has been performed, which leads to little ambiguity from the users point of view [Wilson 2006] of whether the gesture is performed or not. Because of the similar meaning, the additional relation to simple mouse click actions and the implicit feedback, we picked this gesture to be used for performing a selection (Figure 22 on page 38). If used to invoke a selection action with hand gesture input, we refer to this hand gesture as "pinch gesture".

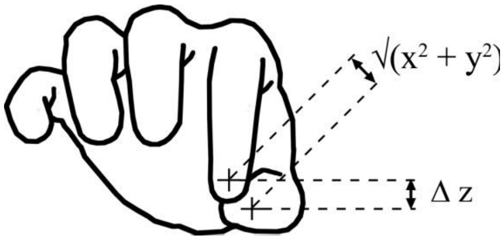


Figure 22: The "pinch gesture" used for selection

A selection, which is mapped onto a single left mouse click, is triggered when the pinch gesture is recognized. A single "click" sound is used to give acoustical feedback to the user.

For gesture recognition the approach described in chapter 4 is applied. As characteristic feature of the pinch gesture we identified the contact between the tip of the thumb and index finger. Referring to the coordinate system illustrated in Figure 17 on page 30, we mapped this feature onto two metrics: (1) the distance along the z-axis and (2) the Euclidian distance of the x and y values between the tip of the thumb and index finger. While the first metric captures the movement of the most active index finger, the second metrics is used to ensure that the fingers are actually brought together to form a ring. For the two metrics we further defined recognition thresholds. The "follow-up-recognition" threshold, which should prevent unwanted gesture recognition while the gesture is exited, is thereby defined in respect to the first metric, as those values are stronger affected by the movement performed when entering respectively exiting the pinch gesture.

5.3 Influence of Limb Movements on User Performance

Empirical evidence suggests that user performance varies, depending on the limbs segments incorporated in the user movements to operate an input device [Balakrishnan & MacKenzie 1997]. Performance further varies with differences in movements of the same limb segments resulting from different directions the input device is moved [Balakrishnan & MacKenzie 1997] [Dennerlein et al. 2000]. While some physical input devices can compensate for those differences (e.g. with force feedback to guide user movements) [Dennerlein et al. 2000], such a physical compensation by means of the input device is not possible when pointing is performed with the hand freely moving in mid-air. Therefore, different movement patterns of the arm and hand, differing in the limb segments and their coordination, may influence the pointing performance of users with our proposed hand gesture interaction techniques.

In the following section evaluations on the effect of limb segments on user performance will be presented and discussed.

5.3.1 Empirical Evidence

With the aim to reveal the effective Index of Performance for the finger, wrist and forearm, Balakrishnan & MacKenzie [1997] conducted a formal evaluation study featuring a serial one-directional tapping task. They evaluated three "limb" and two "stylus" conditions. For each one of the three limb conditions a dedicated custom build input device has been used, which allowed movement of the inspected limb but immobilized the other limbs. Targets for the limb conditions have been aligned horizontally. For the stylus conditions participants used a stylus held between the thumb and index finger to perform the task. In the "left/right" stylus condition, the targets were aligned horizontally, whereas in the "forward/backward" stylus condition the targets were aligned vertically. [3] 10 participants took part in the evaluation and in total 45,000 trials have been conducted. Results revealed that participants performed worse in the finger condition (3.15 bits/s), followed by the wrist (4.08 bits/s), forearm (4.14 bits/s), stylus “left/right” (4.20 bits/s) and the stylus “forward/backward” scoring best with 4.47 bits/s.

The outcome of this study confirms that user pointing performance varies, depending on the limb segments operating the pointing device. Furthermore, the results of the two stylus conditions suggest that users pointing performance also depends on the type of movement which should be performed with the pointing device (in this case horizontal vs. vertical device movement), and the differences in physical limb movements required to control the different device movements.

Also not primary investigating on the issue of the influence of different limb movements on user performance, Dennerlein et al. [2000] reported on an experiment where they ascribed observed differences in user performance to differences in joint kinematics, in particular the multi-joint coordination. For a tunnel steering task, conducted via a stationary mouse and a regular display, they observed that the 10 participants were able to guide the mouse cursor more quickly through horizontal tunnels compared to vertical tunnels. They argue that horizontal mouse movement can be achieved primary in moving the wrist, whereas for vertical movement the arm has to be moved away from the body, which furthermore includes the elbow and shoulder joint, requiring a *“[...] movement of greater inertia and multi-joint coordination – a higher level of difficulty”*[Dennerlein et al. 2000]. Applying force feedback to the mouse, which repels both cursor and participant’s hand from the tunnel boundaries towards the centre of the tunnel, could reduce the observed difference in performance. This additional guidance supports joint coordination in compensating irregularities in the participants’ movements and supports the conclusion that multi joint coordination can be the source of the differences in movement time.

Similar to the two “stylus” conditions in [Balakrishnan & MacKenzie 1997] the two “tunnel” conditions in [Dennerlein et al. 2000] lead to two different movement directions of the cursor respectively the input device. Both studies used the same input device for the two movement directions and results of both studies showed that different user performance could be observed for the two input device movement directions. Different directions of input device movement require different physical user movements, which suggests that differences in limb movement, either in the limbs performing the movement or the differences in joint coordination highly influences user performance.

5.3.2 Conclusion

It is evident from related work that the incorporation of different limbs for operating an input device influences the user performance achieved with the device. Balakrishnan & MacKenzie [1997] showed that isolated limb movement of different limbs results in different user performance in a one-directional serial tapping task.

However, operating input devices mostly involves not only movement of an isolated limb but a coordinated movement of different limbs. For a stylus and a stationary mouse, which are operated with such coordinated limb movements, it has been shown that different directions of input device movement lead to different user performance [Balakrishnan & MacKenzie 1997] [Dennerlein et al. 2000]. This empirical evidence suggests that different limb movements,

---

[3] Balakrishnan et al. call vertically aligned targets “horizontal targets” and horizontally aligned targets “vertical targets”

differing in the limbs involved and the coordination of joints, are a highly influencing factor for user performance.

While Dennerlein et al. [2000] could compensate differences in limb movements with force feedback, which guided user movements, for the proposed hand gesture point and select interaction technique performed in mid-air, no external physical support can be provided to compensate for irregularities in users' movements.

Considering the proven influence of differences in limb movement on user performance, in particular for different input device movement directions, a similar influence can be expected for the proposed hand gesture interaction techniques. We will evaluate the influence of differences in limb movements on user pointing performance with the proposed hand gesture pointing technique in the formal evaluation study described in chapter 5.5.

5.4 Tactile Feedback

Parts of this chapter have been published in [Foehrenbach et al. 2008], but have been further enhanced for this thesis.

When interacting with real-world physical surroundings humans rely on many senses, with sight, hearing and touch complementing, substituting or confirming each other. The absence of visual or tactile perception makes the manipulation of physical objects much more difficult. When grasping objects for example, we use visual and tactile feedback to judge whether we can now lift and hold the desired object. The extent on which humans rely on those senses can be experience with trying to hold up a glass of water blindly or with a numb hand.

With grasping as a metaphor for selecting digital objects in human-computer interaction, the question rises, whether tactile feedback can also improve selection tasks or if visual feedback alone is sufficient. There are many applications and studies stating that tactile feedback indeed can improve the performance of users in human computer interaction.

Before reviewing and discussing previous work in chapter 5.4.2, the terminology will be clarified and defined.

5.4.1 Terminology

When reviewing related work on tactile and forced feedback enhanced interfaces, several terms are used interchangeably [Forlines & Balakrishnan 2008]. This diversity in terminology raises the need for a clear definition of the two feedback modalities and distinction between them. For the purpose of this thesis, tactile feedback is defined and distinguished from force feedback based on the ISO 9241-9, the human somatosensoric system and following the distinction found in [Forlines & Balakrishnan 2008].

In [ISO 9241-9 2000] tactile feedback is described as the *"indication of the results of a user action transmitted through the sense of touch"*. We refine this definition to also incorporate the qualities of the sense of touch, according to the foundations in the human somatosensoric system, and extended the definition to also take into account the impact on user movement. Thereby, we classify tactile feedback and distinguish it from force feedback as following:

- **Tactile feedback** indicates results of a user action. Tactile feedback is transmitted by the human sense of touch which can perceive vibration, pressure, stretching and touch

[Mutschler et al. 2007, p. 698]. Tactile feedback, unlike force feedback cannot restrict user movements.

- **Force feedback** indicates results of a user action in applying force with various strength to the user. It can actively restrict users in their movements if the applied force is large enough. The feedback is sensed through the sense of touch and proprioception.

Using these definitions, we can identify different terms used to describe tactile and forced feedback in related work, which highlights the necessity to pay attention to the way terms are used and what qualities of the feedback are referred to in literature. For example, in [Leung et al. 2007] and [Buchmann et al. 2004] the term haptic feedback is used to describe tactile feedback, whereas in [Akamatsu & MacKenzie 1996] [Hoggan et al. 2008] [Poupyrev et al. 2002] and [Poupyrev & Maruyama 2003] the term tactile feedback is used. Besides the usage of haptic feedback to describe tactile feedback, it can also be found to include tactile and force feedback e.g. in [Burdea 2000] [Hinckley 2003] [Scheibe et al. 2007].

Applying the above introduced definition of tactile feedback and its distinction against force feedback, related work on the use of tactile feedback for pointing devices and the observations on the impact of it will be described in the following chapter.

5.4.2 Tactile Feedback for Pointing Devices

Scheibe et al. [2007] observed that enhancing hand gesture interaction with tactile feedback seems to increase the reliability of interaction tasks. In a pilot study eight participants were asked to perform common interactions in a virtual car cockpit using the corresponding real-world hand movements while wearing a tactile data glove solution. Tactile feedback, sensed as an ongoing vibration on the fingertips, was given when contact of a virtual object with a finger occurred. Tasks were performed with and without the additional feedback. Results showed that participants clearly preferred the tactile system and it was observed that particular small, almost by the real hand occluded objects were operated with greater reliability when tactile feedback was given. Hence tactile feedback seems to improve hand gesture interaction, however the outcome of the study neither gives evidence of the detailed impact on performance nor on error rate nor on movement time.

Other areas in the field of Human-Computer Interaction already make use of tactile feedback. Braille displays allow visually impaired users to explore the internet, mobile phones vibrate when a text message is received, and input devices give tactile clues like the discrimination between keys on keyboards.

By comparing the results of a typing task performed by typists and casual users using a conventional and a piezoelectric keyboard Barrett & Krueger [1994] found out that the performance of both user groups was significantly higher with the conventional keyboard. Here, lack of the familiar haptic feedback (kinesthetic feedback through key travel and tactile through key discrimination) directly decreases the performance.

Effects of enhancing keyboard interaction with a stylus on a PDA with tactile feedback were evaluated by Brewster et al. [2007]. Participants performed a text entry task once in a laboratory and once inside an underground train. A vibrotactile actuator at the back of the device was used to generate two different stimuli which were used to either indicate a successful button press or signal an error. Results showed that tactile feedback improved the number of corrected errors significantly in both settings, reduced the error rate in the laboratory setting and lead to a lower overall workload of the participants who preferred the tactile system over the non-tactile system.

Another evaluation considering pen-based input was conducted by Forlines & Balakrishnan [2008]. Here, tactile feedback was added directly to the stylus. In a selection task they did not only study the effect of different feedback conditions (tactile plus visual vs. visual only) but also direct vs. indirect input and selecting using pointing vs. crossing. Tactile feedback was given to confirm a successful selection. The authors discovered that although tactile feedback didn't show significant beneficial effects for all conditions, it improved the selection time for indirect pointing and direct crossing selection tasks. This outcome suggests that tactile feedback, while having the potential, doesn't per se guarantee for an improved interaction but that the accompanied interaction technique also influences the benefits of tactile feedback.

Akamatsu & MacKenzie [1996] found that the performance of a modified mouse could be improved through additional tactile feedback. In the tactile feedback condition a solenoid driven pin stimulated the tip of the index finger once the cursor overlapped the target area. The feedback was turned off when the target was selected, or the cursor was moved outside of the target area. Note that this differs from the feedback in [Forlines & Balakrishnan 2008], as it is given before the user performs a selection task. Compared to the other feedback conditions, results showed that tactile feedback lead to the highest index of performance with 6.4 bits/s.

Based on these mixed results it seems to be critical to distinguish between different forms of tactile feedback when discussing its usefulness. We identified two different approaches on how to provide tactile feedback.

1. **Proactive feedback:** The feedback is given prior to a certain interaction and indicates a call for action by the user. This means that as soon as a tactile feedback is sensed, the user has to perform a (predefined) action (e.g. click on an object). It might be that the tactile feedback is given until the action is performed. The cited studies by Akamatus & MacKenzie [1996], Scheibe et al. [2007], and Barret & Krueger [1994] can be classified in this category.
2. **Retroactive feedback:** In this case, tactile feedback is given after an interaction has been performed by the user. Here we have to distinguish between two different kinds of feedback. Positive feedback means that tactile feedback is given to indicate that an interaction or task was performed correctly. Negative feedback means that tactile feedback is given to indicate an error or mistake, requiring the user to repeat or correct the action. The cited study by Forlines & Balakrishnan [2008] belongs to the positive retroactive feedback category while the study of Brewster et al. [2007] provides both, positive and negative feedback.

Summarizing the results of the different studies, the proactive feedback seems to increase performance or at least user satisfaction while the results for the retroactive feedback are more mixed. The study by Brewster et al. [2007] might suggest that negative retroactive feedback has a higher influence on user performance. However more research in this area is needed to clarify this issue.

However, in case of combining tactile feedback with hand gesture interaction for WIMP or similar interfaces, the proactive feedback approach seems to be more promising. In such a case, the user needs help in pointing to and selection of an object. Since the computer does not know which object the user is interested in, giving retroactive feedback is not possible. Providing proactive feedback is further in line with the analogy to real-world interaction, where tactile feedback is given to indicate that an object can be lifted and tactile feedback is present as long as the object is moved. Hence, the hand gesture interaction, described in the chapters 5.1 and 5.2, is combined with proactive tactile feedback in a similar way as Akamatus & MacKenzie [1996].

5.5 Evaluation of Hand Gesture Performance

Large parts of this chapter have been published in [Foehrenbach et al. 2008], but have been further enhanced in the previous sections.

We conducted a controlled experiment to assess and compare the usability of the presented pointing and selecting hand gestures (see chapter 5.1 and chapter 5.2) with and without tactile feedback as an input device for large high-resolution displays. Therefore the experiment took place in front of the Powerwall of the University of Konstanz, a large high-resolution display. In the following chapters the experimental settings and the hypothesis will be described. We will report the results in chapter 5.5.7, followed by a discussion and our conclusions in chapter 5.5.8. Eventually, we will propose possible implications for interaction design in chapter 5.6.

5.5.1 Materials

The Powerwall of the University of Konstanz is a wall-sized display with a resolution of 4640x1920 pixels and a physical dimension of 5.20x2.15 meters. It uses a multi projector system with soft-edge blending and is equipped with an optical tracking system developed by A.R.T. This tracking system uses six infrared cameras to cover the area in front of the display. The cameras are able to identify the position and movement of markers that can be placed on persons, e.g. to assess their current location and use this as an input variable. In combination with the A.R.T. finger tracking solution, this system was used for finger tracking. The data glove associated with the finger tracking solution consists of several markers on the back of the hand as well as on three fingers – the latter were attached similar to foxgloves (see Figure 23 on page 44). This construction enables the tracking of the exact position of one's hand as well as single fingers. If every marker is visible for the cameras, this system reaches an accuracy of <1mm. We used this commercial data glove, as it can be accustomed to most of the hand- and finger sizes and should therefore be adjustable to fit most participants. Furthermore hygienic issues, arising from the use by many different users, are minimized, as only a small area of the hand and finger is in contact with the data glove solution (for details on the used data glove solution and comparison to other solutions see chapter 3.4.2 and 6.5). We modified the attachment of the marker on the back of the hand to improve the visibility of the marker for the cameras and therefore increase the tracking quality. In order to provide tactile feedback, we used an extension of this system described in [Scheibe et al. 2007]. Around the inside of the three fingertips covered by the markers, so-called shape memory alloy wires are attached. A wireless connection provides the possibility to attach a low voltage which is perceived by the user as a continuous vibration.

The tasks (see chapter 5.5.2) were presented and interaction was recorded via IEval, a software tool that can be used for pointing device experiments [König et al. 2007b].

To accommodate for the natural hand tremor we integrated a band-pass filter that provides dynamic smoothing of the interaction without restricting fast movements.

We designed a short pre-test questionnaire to assess the participants' prior experience as well as some demographic data (see Appendix A). For subjective assessment of the different experimental conditions we used the questionnaire provided by the ISO 9241-9, which asks participants to rate one device and then rate the second device in comparison to the first device. Users rated the non-tactile as the first device (absolute measurement) and the tactile feedback as the second device (relative to the non-tactile variant). The questionnaire consists of items like overall satisfaction as well as accuracy and fatigue of fingers/wrist/arm, etc. In total it comprises 12 items that have to be rated on a 7-point-scale (see Appendix A).

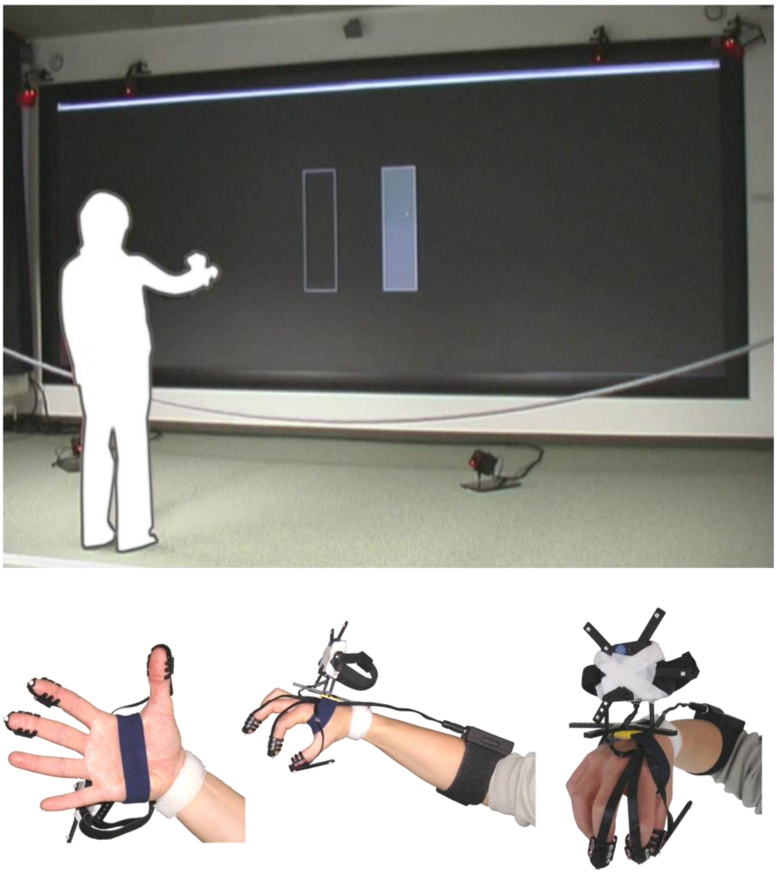


Figure 23: Experimental set-up (top), modified data glove (bottom)

5.5.2 Tasks

We based our experiment on Fitts' Tapping Task as described and suggested by ISO 9241-9 to assess the performance of pointing devices. These tests are widely used and accepted (see [Soukoreff & MacKenzie 2004] for a review). We used the one-directional tapping task which consists of two rectangular targets that are furthermore varied in terms of their width (W) and the amplitude (A) between them. Participants were asked to click on each of these targets in an alternating manner as fast and precise as possible. This "clicking" was done by using the selection gesture illustrated in chapter 5.2.

In the tactile condition, tactile feedback was provided while the cursor overlapped the target area. The tactile feedback to the user's tips of the active fingers (index and thumb) was turned off only after selecting the target or after the cursor was moved outside of the target area. This integration of the tactile feedback is based on the work by Akamatsu & MacKenzie [1996] who provided tactile feedback in a similar way while testing an enhanced mouse.

We furthermore varied the target alignment, using horizontal as well as vertical aligned targets (see Figure 24).

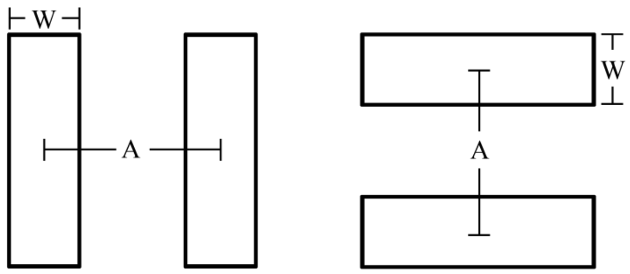


Figure 24: Horizontal (left) and vertical (right) alignment of tapping targets

To cover a wide set of difficulties that can be encountered when interacting in front of the Powerwall, we initially used 3 (W) x 3 (A) combinations for horizontal tasks and 2 (W) x 2 (A) combinations for vertical tasks. The latter was due to the limited vertical size of the Powerwall (2.15m compared to the 5.20m in horizontal) and the necessity that participants may also "overshoot" a target. Larger amplitudes or target widths for vertical tasks may have otherwise resulted in participants performing a selection gesture outside of the display. The exact pixel-values can be seen in Figure 25 as well as the resulting indexes of difficulty. However during the experiment we observed that participants moved themselves to a larger extent in front of the display than expected, triggering the tracking cameras ineffective for the outer parts of the display. Therefore we had to exclude this amplitude for further analysis, resulting in a 3 (W) x 2 (A) combination for horizontal tasks and the corresponding reduction in terms of the index of difficulty from 5.6 bits maximum to 4.6 bits maximum (see Figure 25 (b)).

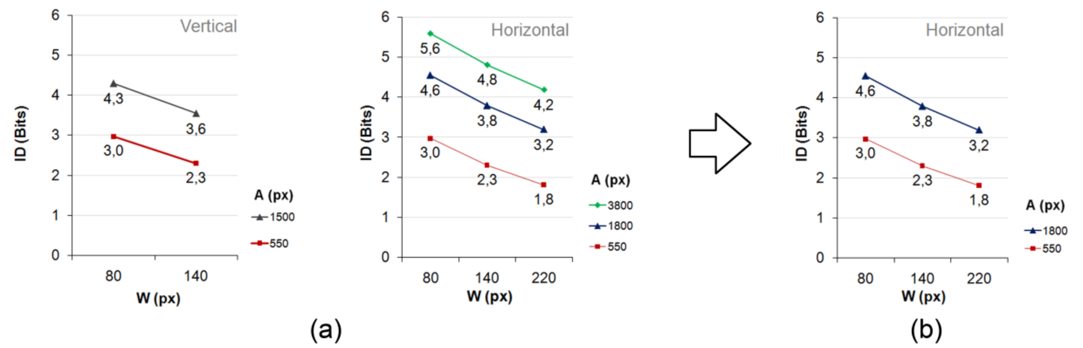


Figure 25: (a) Initial design of W x A combinations and resulting index of difficulties (different colors: amplitudes, x-axis: target sizes, left: horizontal, right: vertical) (b): Resulting W x A combinations for horizontal alignment after exclusion of one amplitude condition (3800px)

5.5.3 Hypothesis

This section describes our experimental hypothesis as well as their foundation in the current literature.

H1: Tactile vs. Non-Tactile

We assumed that tactile feedback would result in a significant performance improvement, expressed by the effective index of performance (IPe) measurement. This hypothesis is in line with the literature review presented in chapter 5.4.2 that strongly suggests that tactile feedback is able to improve user performance in many ways, ranging from lower error rates to lower movement times. The effective index of performance includes both, movement time and error rates (see [Soukoreff & MacKenzie 2004] for details) and therefore provides an appropriate measurement for this hypothesis.

H2: Horizontal vs. Vertical Target Alignment

We assumed that the index of performance for horizontal targets (see Figure 24, left on page 45) would be significantly higher compared to the vertical target alignment (see Figure 24, right on page 45). This hypothesis is in line with findings by Dennerlein et al. [2000]. In an experiment featuring a tunnel steering task, conducted via a stationary mouse and a regular display, they observed that users were able to guide the mouse cursor more quickly through horizontal areas of the task compared to the vertical areas. They ascribed this effect to differences in the joint kinematics, in particular to the multi-joint coordination (see chapter 5.3.1). In a similar way, horizontal and vertical hand movement also relies on different muscles and joints, therefore similar results were expected.

5.5.4 Experimental Design

We used a 2x2 within subjects design with feedback (tactile, non-tactile) and target alignment (horizontal, vertical) being the independent variables. A latin square design was used for counter-balancing in order to address possible effects of sequence, learning or fatigue. Our participants were randomly assigned to one of the resulting four experimental groups. As dependent variable we used the measurements provided by ISO 9241-9, namely movement time (MT, in milliseconds), error rate (ERR in %), and the effective index of performance (IPe in bits/s). The latter combines the movement time and error rate in one single measurement. Since participants were asked to perform a task as fast and precise as possible it should be considered as the most important measurement. The measures were calculated using the following formulas:

Index of difficulty : $ID = \log_2 (A / W + 1)$

Effective width : $W_e = SD(\Delta o) \times 4{,}133$

Effective index of difficulty: $ID_e = \log_2 (A / W_e + 1)$

Effective index of performance: $IP_e = ID_e / MT$

5.5.5 Participants

We selected 20 participants to take part in our experiment. Of those, 15 were male and five female. The average age was 30.8 years with a standard deviation of 9.9 years. All of them were regular computer users, while 13 already had some experience with large displays (standard projector or Powerwall). None of the participants had prior experience with a data glove or something similar.

5.5.6 Procedure

Each session started with the pre-test questionnaire. Users were then equipped with the data glove followed by a short functionality test of the tactile feedback. In the next step, participants were asked to step in the centre in front of the Powerwall, three meters away from the display. They were instructed about the interaction, the gestures they should use to interact, and to be as fast and precise as possible.

A training session was started then, consisting of a full block of vertical and horizontal tasks as well as non-tactile and tactile feedback, whereas the sequence was based on the participants assigned test condition. During training we used 2 (W) x 2 (A) combinations and ten trials for each combination, resulting in 160 trials. The selection of the reduced WxA combinations was done based on the goal to keep the training rather short and at the same time to reach similar difficulty levels as in the following real tasks. During these each participant completed two blocks of the assigned condition, and now 16 trials for each WxA combination, resulting in 832 trials. All participants together completed 16,640 trials of which 12,800 trials were used for analysis, due to the tracking problem mentioned in chapter 5.5.2.

			Trials	
			Per Participant	Total
Training		2 (W) x 2 (A) x 10 Trials x 2 feedback type x 2 target alignment	160	3200
Test	Horizontal:	3(W) x 3(A) x 16 Trials x 2 feedback type x 2 Blocks	576	11,520
	Vertical:	2 (W) x 2(A) x 16 Trials x 2 feedback type x 2 Blocks	256	5,120
			832	**16,640**
Analysis	Excluded:	3(W) x 1(A) x 16 Trials x 2 feedback type x 2 Blocks	192	3,840
				12,800

Table 1: Trials Counts

After completion of the tapping test, participants were asked to fill in the ISO 9241-9 questionnaire. The experiment lasted in total about one hour per session and participants were given 5 EUR as compensation.

5.5.7 Results

This section describes the analysis and results of our experiment. First the identification and treatment of outliers will be described, followed by the model fit and the results.

Treatment of Outlier

Before calculating the dependent variables, outlier resulting from accidental double clicks and other anomalies have been identified and removed from the trials. At first, wrong side outlier [MacKenzie & Oniszczak 1998], which result from accidental clicks, immediately following a successful selection, have been removed. Those trials have been identified by comparing the position of the trial with the centre of the two targets. Trials being closer to the previous target than to the target which had to be selected, in respect to the axis of approach, are considered wrong side outlier and have been removed from the trials (67 trials, 0.5 %). Furthermore, following a recommendation in [Soukoreff & MacKenzie 2004] statistical measures were used to identify outliers. Therefore, for each feedback type, target alignment and amplitude combination the mean movement time, the mean distance from the target centre along the axis of approach and the corresponding standard deviations have been calculated. Trials were removed if their movement time wasn't within the range of ± 3 standard deviations around the movement time mean (188 trials, 1.5 %), or if their distance from the centre of the target wasn't within the range of ± 3 standard deviations around the mean center distance (136 trials, 1.1%).

Model Fit

At the beginning of the analysis we calculated the model fit, averaged across all participants, for Fitts' Law. The results showed a very high model fit for each of the factor combinations, with r² constantly above .99. Therefore we can assume that the Fitts' Law model fits quite well for our experiment.

	Regression Function (MT in sec)	r^2	ID Range
Horizontal, tactile feedback	MT = 0.330 + 0.212 ID	0.991	1.8 – 4.6
Horizontal, no additional tactile feedback	MT = 0.269 + 0.242 ID	0.991	1.8 – 4.6
Vertical, tactile feedback	MT = 0.303 + 0.264 ID	0.994	2.3 – 4.3
Vertical, no additional tactile feedback	MT = 0.305 + 0.265 ID	0.998	2.3 – 4.3

Table 2: Model fit for each factor combination

H1: Tactile vs. Non-Tactile

Our first hypothesis stated a significant difference in favour of the tactile feedback in terms of the effective index of performance (IPe). Results of our RM-ANOVA however show that this is not the case. For both horizontal and vertical target alignment the non-tactile feedback performed better, however the differences are very small and not significant (horizontal means: non-tactile 3bits/s, SD: 0.29 bits/s vs. tactile 2.99 bits/s, SD: 0.31 bits/s, $F_{1,19}$ = .053 p = .820; vertical means: non-tactile 2.53 bits/s SD: 0.23 bits/s vs. tactile 2.46 bits/s, SD: 0.28 bits/s, $F_{1,19}$ = 3.637 p = .072, see Figure 26). Therefore we have to reject our hypothesis in favour of the null-hypothesis, stating there is no significant difference.

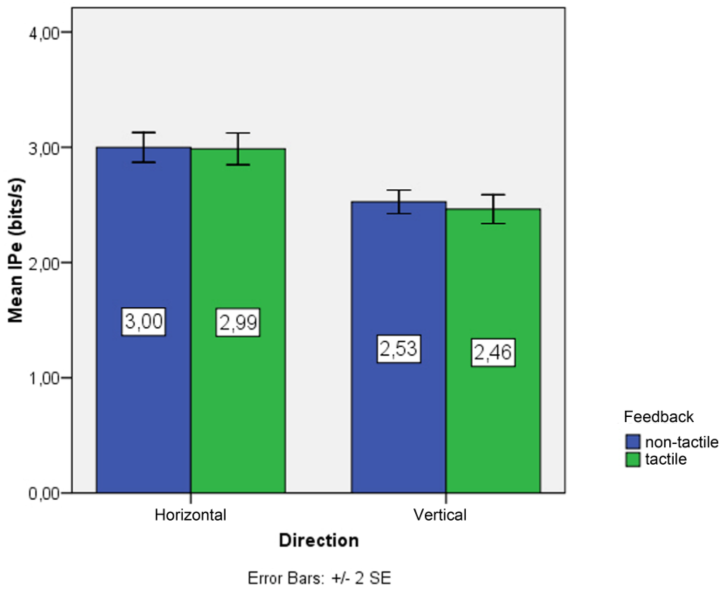


Figure 26: Effective index of performance for horizontal and vertical target alignment

H2: Horizontal vs. Vertical Target Alignment

Our second hypothesis stated a significant difference in favour of the horizontal target alignment compared to the vertical one in terms of the effective index of performance. As it turns out, this is indeed the case ($F_{1,19}$ = 124.857 p < .001, horizontal mean: 2.99 bits/s, SD: 0.29 bits/s vs. vertical mean: 2.49 bits/s, SD: 0.25 bits/s) Therefore, we can accept our second hypothesis.

Effect of Tactile Feedback on Error Rate and Movement Time

We further analyzed the effect of the tactile feedback in terms of error rate and movement time. Results show that the movement time is slightly lower for both vertical and horizontal target alignment when providing the user with tactile feedback. However these differences are not significant (see Figure 27, on page 50). (Horizontal: $F_{1,19}$ = 3.84 p = .065, 1,021.77 ms, SD: 124.94 ms vs. 988.64 ms, SD: 119.15 ms, Vertical: $F_{1,19}$ = .049 p = .827, 1,174.36 ms, SD: 131.23 ms vs. 1,171.28 ms, SD: 145.37 ms)

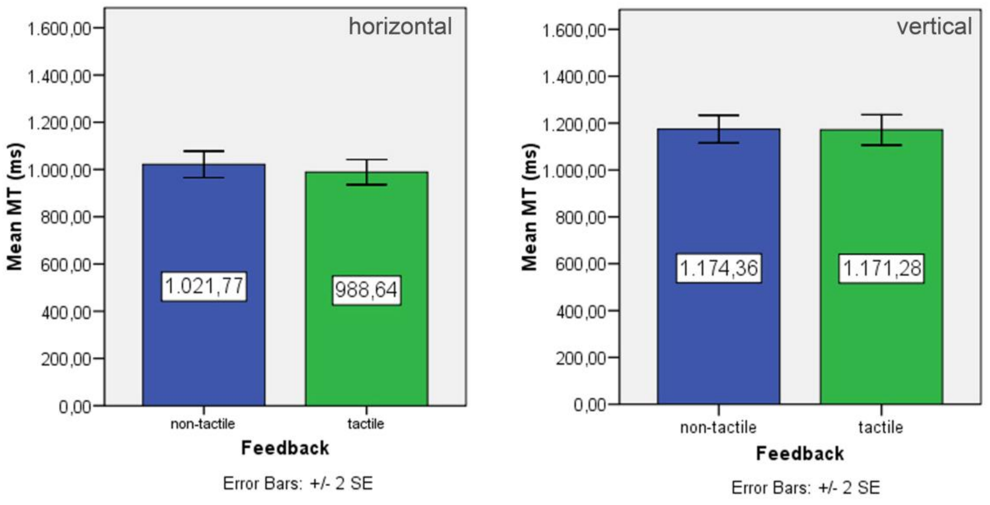


Figure 27: Influence of tactile feedback on movement time

Regarding the error rate results look different (see Figure 28). For the horizontal target alignment we discovered a significant higher error rate when using tactile feedback ($F_{1,19} = 9.17$ $p = .007$, 10%, SD: 4.8% vs. 12%, SD 6.2%) – for vertical alignment the difference was not significant ($F_{1,19} = 2.61$ $p = .112$, 14%, SD: 5.6% vs. 15%, SD: 5.9%).

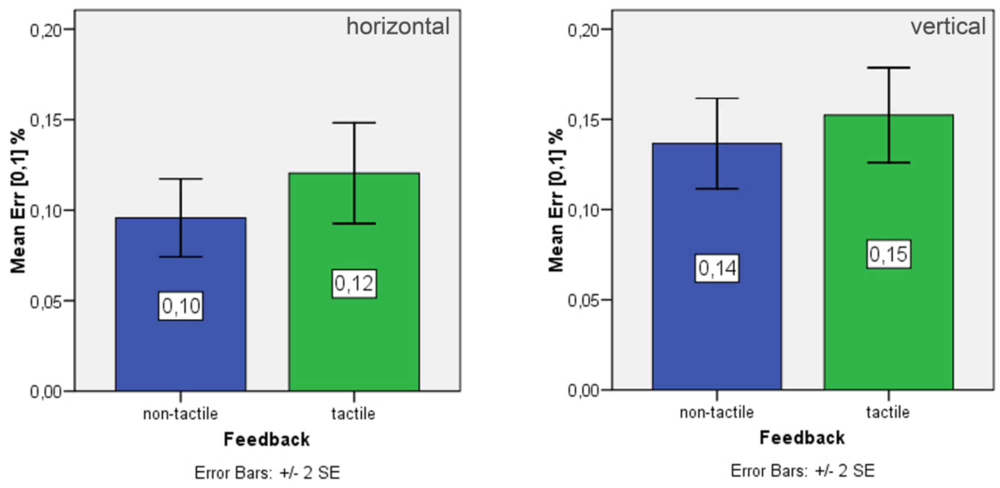


Figure 28: Influence of tactile feedback on error rate

Questionnaire

Regarding the subjective feedback derived from the questionnaire our participants rated nearly the entire items positive for the non-tactile feedback (with the exception of arm fatigue, see Figure 29). The second part of the questionnaire asks to rate the tactile-feedback relative to the non-tactile. Results show that our participants either liked or disliked the tactile-feedback, resulting in three nearly discrete groups (7 dislikes, 7 likes, 6 undecided, see Figure 30). We looked for correlations between task performance and whether a participant was in the "I like tactile" or "I dislike tactile" group. However there was no significant effect.

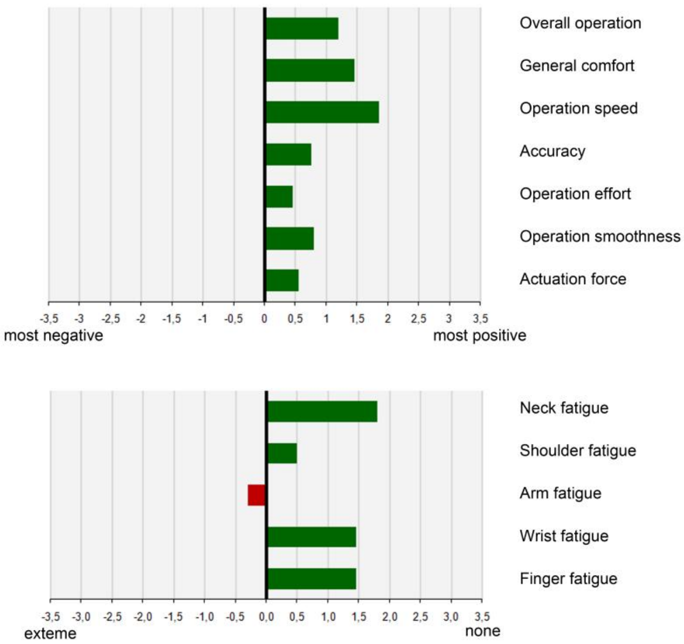


Figure 29: Subjective user rating for non-tactile feedback

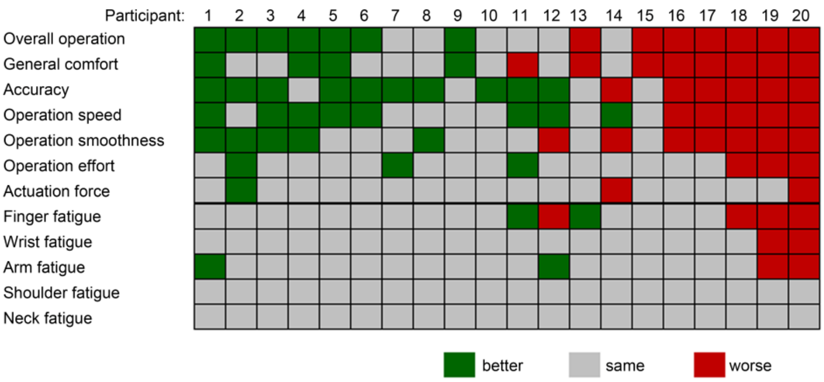


Figure 30: Relative user rating for tactile feedback (Participants ordered according to their rating)

5.5.8 Discussion and Conclusion

Based on previous findings of Kendon [2004] and the experimental results of Vogel & Balakrishnan [2005] we identified suitable gestures for pointing and selection tasks and realized gesture recognition in combination with a commercial finger tracking device. The non-tactile version of our hand gesture interaction was very well received by the participants, with 11 positive and only one negative rated item on the ISO satisfaction questionnaire. Also the effective index of performance with a mean of 2.53 bits/s for vertical and 3 bits/s for horizontal target alignment is promising and suggests that hand gesture interaction provides an adequate and valuable interaction technique for large, high-resolution displays.

Tactile Feedback

Besides investigating the general usability of hand gesture interaction for large, high-resolution displays another contribution of this study is the evaluation of the effect of tactile feedback on it. The findings from [Scheibe et al. 2007] and [Akamatsu & MacKenzie 1996] discussed in chapter 5.4.2 suggest that proactive tactile feedback may improve user performance since the additional information channel can complement or substitute visual information. However results of this evaluation study show no significant effect in terms of effective index of performance and even a small but significant higher error rate for horizontal target alignment when using tactile feedback.

One explanation might be that participants did not take advantage of the additional feedback since they relied more on their visual observations when initiating a selection, as this is more common and known. So, the performance did not show a difference as tactile feedback might have simply be tolerated but not used by the participants.

However in the negative case, the additional tactile feedback could even interfere with the visual information. It is known from cognitive science that tactile and visual stimulations are not processed with the same lag and velocity and measured reaction time differ [Serge 1997]. Users may react irritated if the same information (target reached) gets delivered from different channels at different times. Moreover some participants mentioned that they felt to be set under pressure by the additional feedback, what could also be a reason for the slight drawback considering the error rate.

Basically, the findings of previous research on tactile feedback could not be directly transferred to our proposed hand gesture interaction. The empirical results showed no benefit of tactile feedback at least in our test setting, in which visual and tactile information were provided to code the same event redundantly. Regarding future research we think that a more systematical understanding and analysis of tactile feedback is needed. While the classification in proactive and retroactive feedback based on the current literature is a first start, our results suggest that there are clearly additional factors that influence the utility. It might even be that the technical implementation of the tactile feedback plays an important role – while it is quite common for mobile phones to be equipped with some kind of vibration technique, it might be at first rather inconvenient to feel a vibration directly at the fingertips without physically touching an object. We suggest the intensified use of longitudinal designs for future studies, which can help to further clarify the influence of such factors.

Movement Direction

Furthermore our study confirmed the findings of Dennerlein et al. [2000] concerning the effect of movement direction on user performance. The results showed with 2.99 bits/s horizontal versus 2.49 bits/s vertical a significant effect in terms of the effective index of performance. Similar to them this effect could also be due to differences in joint kinematics and the different muscle groups incorporated in the two movements. Physical movement for vertically arranged targets relies on more movement in the upper arm compared to horizontal movements in both evaluation studies. That different limb movements can reveal different user performance is a known issue in human-computer interaction [Balakrishnan & MacKenzie 1997] [Dennerlein et al. 2000]. Therefore the differences in limb movements might be the reason for our observed the differences in performance.

However in contrast to the results of Dennerlein et al. [2000], the tactile-feedback did not compensate the differences between horizontal and vertical target alignment. This could be due to the fact that in Dennerlein's study force feedback restricted mouse movement whereas tactile feedback in our evaluation only served as additional information but did not physically hinder users in their movement and therefore provided no additional guidance to improve the lower physical performance during vertical movements.

Another reason for the differences in performance might be the combination of the gestures used. Performing the selection gesture could lead to a slight repositioning of the cursor, due to correlated muscle movements at the back of the hand which are captured by the hand target, used for gathering the orientation and position of the hand. When holding the inside of the hand facing the floor while interacting, this could affect the performance measure of trials for vertically arranged targets, but not for the horizontally arranged targets. However, a lower performance for vertical movement directions could also be observed for participants holding the inside of the hand facing the left wall (see participants 5 and 17 in Figure 31), which makes differences in limb movements more likely for being the main reason of the observed differences in user performance.

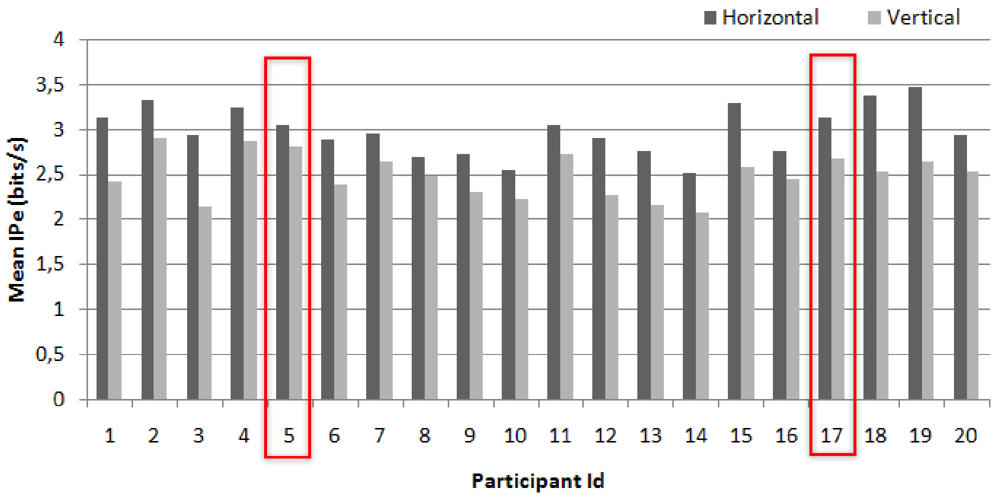


Figure 31: Performance of horizontal movements over vertical movements in terms of the effective Index of Performance, averaged for each participant across tactile and non-tactile feedback conditions.

The influence of different movement directions should be considered when designing interaction techniques and user interfaces for hand gesture interaction at large high-resolution displays. In Chapter 5.6 we will describe possible implications.

We also suggest research activities to further invest on the influence of movement direction on user performance at large high-resolution displays. If differences in limb movement are the reason for the different performance, the same effect can be expected for interaction with other input devices held in mid-air, applying an absolute mapping from input device movement to cursor movement. Influence of movement direction on user performance could therefore be a general issue for distant interaction in this context.

Tracking hand movements

During the evaluation study we observed that our participants moved their hand, equipped with the data glove, to a greater extent than expected. Wide hand movements triggered the tracking cameras ineffective for the outer parts of the display. User movements were therefore limited to a small area above the indicated stationary point. Such a restriction of user movements is critical in the context of interaction with LHRD, where supporting the mobility of the user is a key concern when designing interaction techniques.

To increase users mobility, the number of cameras used for tracking could be increased, but it is troublesome that even with six cameras user movements are limited to such a small area. Because of the observed limitations of the A.R.T. finger tracking solution and the hygienic and fit issues evident for alternative data glove solutions, we decided to design a novel data glove solution, which still addresses the issues of hygiene better than alternative commercially available data gloves.

The fact that arm fatigue was rated negative by our participants further urges the need for a light-weighted data glove, in order to avoid increasing the physical effort needed to perform the hand gesture interaction techniques. The designed novel data glove solution, which we named Whitey and its components are described in chapter 6.

5.6 Implications for Interaction Design

The evaluation described in the previous chapter 5.5 shows that performance for rapid target acquisition tasks, conducted with hand gesture interaction techniques on a large high-resolution display, significantly depends on the direction of target approach movements. Horizontal target alignments which lead to a horizontal approach direction hereby outperform vertically aligned targets. The influence of movement direction during approaching a target has been ascribed to two reasons. First, the combination of gestures used, second, differences in physical limb movements. For both of the ascribed reasons, implications on designing interaction and user interfaces can be derived with the aim to improve user performance.

Compensating Unwanted Cursor Repositioning due to the Combination of Gestures Used

The combination of gestures used for point and select could lead to a slight repositioning of the cursor while performing the select gesture. Although less likely for being the main contributor to the observed performance difference (see chapter 5.5.8), this effect should be minimized.

To meet the users intend the unintentional repositioning of the cursor during a selection should be reversed in a way that the point of selection matches the intended point. Whenever a selection is triggered, the actual cursor position should be set to the intended cursor position. A dedicated filter applied to the cursor position could be used to provide such a system behaviour.

The intended cursor position could for instance be derived by going backwards a fixed, yet adjustable, amount of time in a history of cursor positions. Keeping a history and not just reposition the cursor along a predefined vector covers all possible hand postures. Therefore each potential influence on the cursor position is covered and not just the repositioning along the vertical axis when holding the inside of the hand facing the floor. Going back in the history a fixed, yet adjustable, amount of time and not define gesture dependent thresholds to determine the intended cursor position in the history of cursor positions makes the mechanism independent from the used gestures. Therefore this mechanism can be used to also cover unwanted cursor repositioning with other input devices, for example when pressing a button on a mouse held in mid air.

Compensating Influences on User Performance of Differences in Physical Limb Movements

The main reason for the discrepancy in performance has been ascribed to the differences in the user movements for horizontal vs. vertically movement directions. A derived guideline for designing user interfaces for LHRDs operated with hand gesture interaction techniques can therefore be “avoid vertical and force horizontal approach directions in (rapid) target acquisition tasks”. Consequences of this are, but aren’t limited to, the following recommendations:

1. If it is likely that a common task requires users to alternate between elements, arrange those elements horizontally.
2. Put elements (e.g. used to switch views, for activation or deactivation of options, or invoking functions) in a position where horizontal approach directions are more likely.
3. When using context sensitive menus put most often used options in the axis of horizontal movement directions.

As an example, imagine a user zooming stepwise into a visualization (see Figure 32) using an interface element similar to the one in Figure 32. Whenever the user accidentally enlarged the visualization too much, the user immediately zooms back out to a previous state, using the button at the bottom of the interface element. The two elements for “zooming in” and “zooming out” are aligned vertically. Repositioning the element in a way that this alignment would be horizontal (see Figure 33 on page 56) would increase the performance of the user for this specific task.

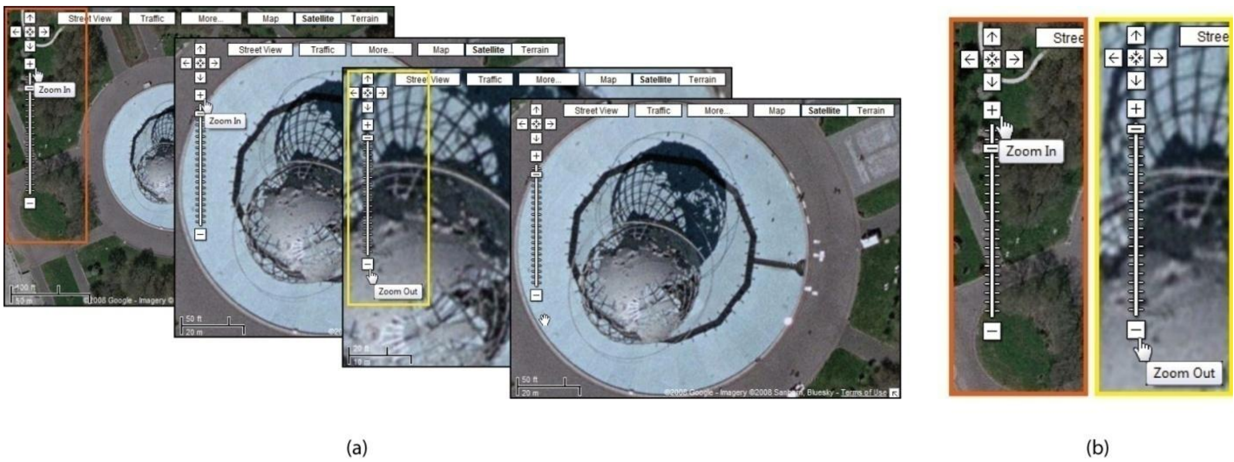


Figure 32: Stepwise zooming into a geographical map followed by a reverse zoom out step (a). The buttons for the “zoom out” and “zoom in” function are arranged vertically (b).[4]

[4] Screenshots taken from http://maps.google.co.uk/maps, (last accessed on Jan. 15, 2009)

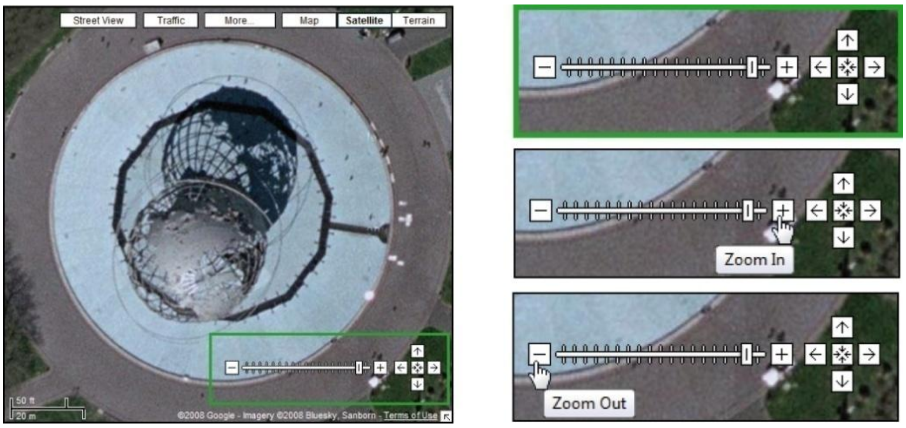


Figure 33: Altered UI Element, where the icons for the zoom out and zoom in function are arranged horizontally (modified Screenshot).[5]

However, it has to be carefully considered, which task is more common: the alternation between the elements, or a single activation of one of the elements. If a single activation is performed more often the positioning at the bottom of the screen would interfere with the second recommendation. According to the second recommendation, the original position would be better for a single activation task, as the element is approached through a horizontal movement.

Similar considerations can be made for drop down menus commonly found in WIMP applications. The items within those menus are arranged vertically, as for example in the Microsoft Office Word 2002 screenshot shown in the top row of Figure 34. Arranging them horizontally, as found in Microsoft Office Word 2007 (see Figure 34) would be in favour for the use with hand gesture input, following the second recommendation.

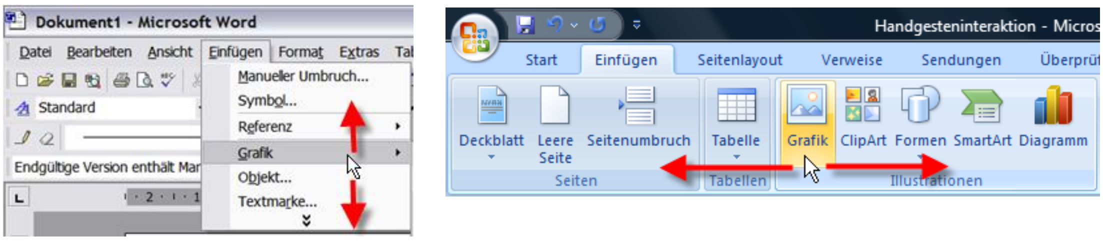


Figure 34: Vertically arranged items in the drop down menu found in Microsoft Office Word 2002 (left) compared to horizontally arranged items found in Microsoft Office Word 2007 (right).

---

[5] Screenshot taken from http://maps.google.co.uk/maps (last accessed on Jan. 15, 2009) and modified

6. Whitey: a Novel Data Glove Solution

Using glove-based solutions for tracking hand movements is used by several researchers to explore advanced hand gesture interaction techniques before robust computer-vision-based, non-contact tracking of hand movements becomes widely available [Ni et al. 2008] [Vogel & Balakrishnan 2004] [Vogel & Balakrishnan 2005]. Glove-based solutions can provide high-dimensional output at high sampling rates, which can then be used to model and realize hand gesture interaction techniques.

We identified the commercial A.R.T. finger tracking solution as one which reduces hygiene-related problems and those connected with bad fit present in other data glove solutions (see chapter 3.4). However, in a formal evaluation study we observed that this solution significantly limited user mobility (see chapter 5.5). Supporting user mobility is a key concern when designing interaction techniques for LHRDs, which makes the A.R.T. finger tracking solution not a practical solution, at least for our setting. Those observed restriction on user movements, and the issues of hygienic and fit evident for alternative data glove solutions motivated the design of a novel data glove solution.

Besides increasing user mobility, we also wanted to consider further characteristics when designing a novel data glove solution:

1. The novel data glove solution should address the hygienic problems better than other commercially available data gloves.
2. It should be adaptable to many hand sizes to provide a good fit.
3. Our formal evaluation study has revealed that arm fatigue is an issue with hand gesture interaction at LHRDs. Heavy data gloves can cause faster fatigue than light weighted data gloves. Therefore we wanted our data glove to be light weighted.
4. The time needed for calibration needed for each user should be small, as we observed in our evaluation study that this can be a time consuming task.
5. The introduced average delay of tracking and delivering the tracked data should be small, to leave room for gesture recognition and graphical output, without going beyond 20 ms, a lag (the delay between input action and output response) that is not noticed by users and does not degrade user performance [ISO 9241-410 2006].

Whitey, the novel data glove solution we designed, and which will be described in the following sections, combines a fixed arrangement of markers attached to a textile glove, an optical tracking system and a finger classification algorithm. The optical tracking system tracks the 3d position of the markers and the 3d position and orientation of the target, which are attached to the glove and reflect movements of fingers and the palm. We use a textile glove to reduce the time needed for calibration. However using a glove is not a necessity, as we will describe in the following chapters.

To ease subsequent modelling, we apply a novel finger classification to associate the tracked 3d positions to the finger they originate from. To achieve this, our finger classification algorithm combines knowledge on the physical arrangements of the markers and the target in relation to the hand, biomechanical constraints of the human hand, and heuristics of user movements.

During the process of associating the data delivered by the optical tracking system to fingers, we transform the data in a way that effects of physical limb movements, other than resulting from the fingers are eliminated. The resulting smaller feature space (which is rotation and position invariant and located in a local hand coordinate system) reduces the complexity of the input data. Transferring raw input data into a local coordinate system is accounted for to eliminate *"[…] a large source of the irrelevant variation present in the raw signal, thus easing subsequent modeling, and can be superior to using only derivative information"* [Wilson 2007].

Concurrently, we also keep the raw input data and associate it with the hand and its fingers, thereby giving subsequent steps the opportunity to choose which variations are considered irrelevant and which are not. Therefore subsequent modelling can make use of both: the 3d position of fingers and the 3d position of the hand and its orientation as they are directly measured, and 3d position of fingers located in the smaller feature space of the local hand coordinate system. For instance, a subsequent gesture recognition process can consider only the position of fingers located in the local hand coordinate system to recognize static hand gestures independently from the position and rotation of the hand. Information on the rotation and position of the hand, which might be irrelevant for recognizing static hand gestures, could however be relevant when designing hand gesture interaction techniques as they can serve as further input dimensions.

In the following sections, we describe Whitey and its components, based on the initial implementation we realized. Therefore we combined a textile glove with the optical tracking system developed by the company A.R.T.. Such an optical tracking system is installed to cover the area in front of the Powerwall of the University of Konstanz a wall-sized high resolution display (see chapter 5.5.1) where we explore our hand gesture interaction techniques for LHRDs.

However, besides this particular implementation, the ideas and concepts behind Whitey, allows it to be transfer to other settings as well. What parts are essential and which ones can be realized otherwise will be described explicitly in chapter 6.6.

6.1 Tracking Hand Movements

For our implementation we used the commercial infrared optical tracking system developed by the company A.R.T. [ART d], however other tracking solutions providing a similar functionality can be used as well (see chapter 6.6).

This optical tracking system can be used to track objects in a three dimensional room. It is capable of tracking the position of single "markers" or the position and orientation of "targets", which consist of a fixed arrangement of at least four markers (see Figure 36 on page 60). For markers three degrees of freedom (3dof) can be tracked, which describe their position in a three dimensional coordinate system. The fixed arrangement of markers that form a target allows to track six degrees of freedom (6dof) for the target, its 3d position and its rotation along the axis of a three dimensional coordinate system. All three rotations describe the orientation of the target within a three dimensional coordinate system. This coordinate system is defined by conducting a calibration process for the tracking system and is called the "room coordinate system" (described below and illustrated in Figure 35).

Markers and targets are tracked with a frequency up to 60 Hz if they are in the field of view of at least two cameras [ART 2005]. In Whitey, those markers and targets are attached to a textile glove, which will be described in the next section. In our setting the optical tracking system is installed at the Powerwall of the University of Konstanz (see chapter 5.5.1). Four optical cameras are positioned on top of the display with two located at each side, to cover the area in front of the display.

The system requires a calibration of the room, which is covered by the installed cameras and hence in which tracking of markers and targets is possible. The calibration defines a three-dimensional Cartesian coordinate system and its origin. The 3dof and 6dof data refers to this coordinate system, which is called the “room coordinate system”. The room coordinate system of the infrared optical tracking system installed at the Powerwall of the University of Konstanz has its origin at the bottom left corner of the display of the Powerwall, the x-y plane parallel to the display surface with the x axis parallel to the bottom with positive values right of the origin and the y-axis parallel to the walls with positive values above the origin. The z-axis forms a right-handed coordinate system with positive values in front of the Powerwall (see Figure 35).

Figure 35: The “room coordinate system” of the optical tracking system installed at the Powerwall

Single markers can be tracked without further requirements, whereas for targets an additional calibration process is needed before they can be tracked by the tracking system. The target calibration process is required to: 1) assign a unique target id, 2) determine what the position data of the target should refer to, either to a dedicated marker of the target or the centre of all markers (this position is also referred to as the origin of a target, see Figure 36 on page 60), and 3) configure how the orientation of the target should be calculated. With the “due to room” option, the orientation of the target during the beginning of the calibration process is defined as being perfectly aligned with the axis of the room coordinate system, therefore the axis of the “body coordinate system” are parallel to the axis of the room coordinate system and no rotation is evident.

Single markers differ from targets in the degrees of freedom of the available information (position: 3dof vs. position and orientation: 6dof) and in the assignment of the id. Whereas targets are assigned a unique id which can be used for identification, markers are assigned the next highest available and not already used id. A marker is assigned an id at the first time it is tracked by the cameras, this id is bound to the marker as long as the cameras keep tracking it, and losing track of the marker causes the id to get discarded.

6.2 Arrangement of Markers and a Target

We define a fixed arrangement of markers and a target in reference to the hand and finger. Those markers and the target define the locations which are being tracked. To keep the markers and the target in a stable position during hand movements, we attach them to a conventional textile glove, intended for cosmetic and medical use. The textile glove is fast to put on and can furthermore avoid contact of the skin with the marker. With the kind of marker we used (passive spherical markers, which we will describe below) contact with the skin can reduce their capacity of reflecting infrared light.

Using a glove to attach the markers is not mandatory. We used a textile glove due to the reasons above and the fact that attaching the markers and the target to a glove can reduce the amount of time needed for calibration of the system if different users want to use it. Although the markers and targets can be attached to the glove in a way that removing them is easily done (e.g. in using counter sunk bolts), which can be utilized to wash the glove in order to reduce hygienic-related problems, not using a glove can further reduce hygienic-related problems. Hence there is a tradeoff between minimizing the calibration effort and reducing hygienic-related problems.

Any type of marker, which can be tracked by the used tracking system, can be used for the data glove. We decided to use passive spherical markers, which are small light-weighted ball shaped objects covered with retro-reflective material reflecting infrared rays. They have the advantage of being visible for the cameras from all sides, a property shared only by the so called “big active markers” which are too big and heavy to be used for a data glove (for information on the marker types available and a comparison of them see [ART c]).

The glove of the A.R.T. finger tracking solution is equipped with small active markers, required for finger identification via synchronized modulated flashes. Those small active markers are not visible from all sides but have to be held with the marker facing the cameras. This restriction may be the reason for the small area in which the users hand could be tracked correctly during the evaluation study described in chapter 5.5. Using spherical markers therefore increases the area in which the markers can be tracked and hence better support the mobility of the user.

Using passive markers, which only reflect but do not actively send out infrared rays, avoids the need for providing power supply. As a result we maintained the weight of the data glove low, and the data glove is robust against accidental drops.

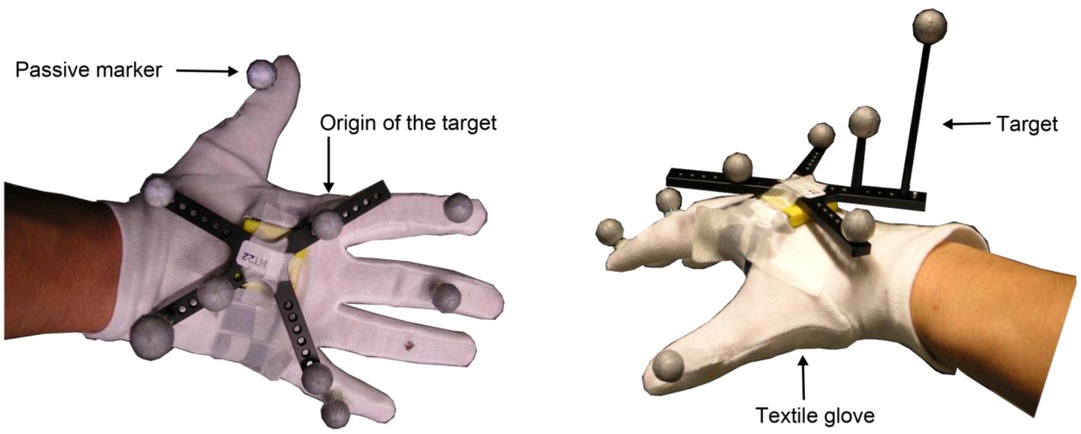


Figure 36: Custom build data glove of Whitey, with passive markers and a target attached to a textile glove

The markers are placed on top of the outer phalanx[6] of the thumb and index finger and the middle phalanx of the middle and little finger (see Figure 36). The ring finger is not equipped with a marker as the ring finger is one of the least independent fingers when it comes to isolated finger movements [Häger-Ross & Schieber 2000].[7] Positioning the marker on top of the fingers has the advantage that movements of the fingers towards the inside of the hand aren't hindered and using other input devices such as a keyboard is possible without the necessity to put of the data glove. Attaching the marker right in front of the fingertip or underneath the finger would impede finger movements much more than the chosen position. However, forming a fist is a posture that is restricted by the placement of the index finger marker on the outer phalanx. Placing the index finger marker at the middle phalanx would enable this posture, but with the aim to receive more precise data on the tip of the index finger the outer phalanx was chosen for the placement of the index finger marker. Data on the tip of the index finger and thumb is important in the context of this thesis, as those two fingers are the most relevant for the classification of the used gestures (see chapter 5.1, 5.2, 7.1.2 and 7.2.2 for a description of the used gestures). Although we placed the marker for the middle and ring finger onto the middle phalanx to allow users to close their hand, it is also possible to place them onto the outer phalanx if more precise information of those fingers is required. However, placing the marker for the middle finger onto the outer phalanx may cause the second heuristic applied in our finger classification algorithm to fail easier, as the outer phalanx can be moved to a greater extend with an abduction movement (the heuristic will be described in chapter 6.3.1, section "Step 3: Classification of the Fingers into Index, Middle, Little and Thumb").

For information on the orientation of the hand, a target is mounted on the back of the palm, with one marker located above the joint connecting the index finger with the palm. We define this marker to be the origin of the target and the marker whose position will be reported as the position of the target. We have chosen this location, to use the position of the palm as the starting point of the imaginary ray used for positioning the display cursor with our "palm pointing" hand gesture interaction technique (described in chapter 5.1). Through incremental tests we found that this location leads to an intuitive way of positioning the display cursor. We placed a marker at this position and assigned it to be the origin of the target, to get the starting point for our imaginary ray directly delivered by the tracking system. Other locations for the origin of the target are also possible, but would then require calculating the starting point for our imaginary ray. The design of the target is due to the location of the optical cameras and our aim to provide a good visibility of the target for the cameras when the user is moving. Therefore we use a larger amount of marker than necessary for defining a target and arrange them in a way that, for most of the hand movements we expect our users to perform, at least two cameras have a sufficient view onto the target in order to be able to identify it.

In order to define the default orientation of the target, referring to an orientation where the target is considered to be perfectly aligned with the room coordinate system and no rotation is evident, the target has to be calibrated. Concurrently with defining the default orientation during the calibration a unique id is assigned to the target, which will be used by the subsequent finger classification algorithm to assign the 6dof data to the corresponding hand. In our setting this default orientation is described by a posture when a user, equipped with the glove, points

---

[6] The bones of the fingers are called phalanx. The phalanx located in the finger tips is called outer phalanx. For the index, middle, ring and little finger the neighboring bone of the outer phalanx located closer to the palm of the hand is called the middle phalanx.

[7] As a curiosity, in the 19th century the range of independent movement of the ring finger was increased for pianist by surgically dividing the accessory tendons binding the ring finger to neighboring fingers [Parrot & Harrison 1980].

directly towards the display of the Powerwall. An imaginary ray, emerging from the joint connecting the index finger with the palm and following the orientation of the palm, should thereby describe a straight line which intersects the display surface at the exact same point on the x-y plane as where the ray is emerging from the hand of the user. The tracked orientation of the target describes the deviation from this default orientation.

We use the orientation of the target in our finger classification algorithm and the position of the target to translate and rotate the position of the finger marker to the position they would inhabit in a “local coordinate system”, illustrated in Figure 40 on page 66. In order to apply biomechanical constraints and heuristic knowledge on hand movements, for assigning tracked positions to the origination finger it is essential that the axis of the local coordinate system are parallel to those illustrated in Figure 40 on page 66, the origin however can be different. In our setting we achieve this in calibrating the target with a posture described above (with the “due to room” option in the A.R.T. tracking system). If other systems for tracking the position and orientation of markers and targets are used other mechanism may apply.

6.3 Hand- and Finger Classification

In this chapter, the algorithm for hand and finger classification and the rationales behind it will be described. As described at the beginning of this chapter, we apply a finger classification algorithm to ease subsequent modelling, in assigning the tracked 3dof data to the finger they originate from and the 6dof data to the hand they originate from. Our algorithm transforms the assigned finger position into a smaller feature space, describing a local coordinate system, which is hand rotation and hand position invariant, while concurrently keeping the raw data delivered by the tracking system and also assigns it to its originating finger respectively hand. Therefore subsequent steps can make use of what is more suitable for them.

6.3.1 Algorithm

The arrangement of the markers and the target, described in the previous chapter, in combination with the optical tracking system is used to deliver the input data for the hand and finger classification process. Although used in this setting, the algorithm can also be used with other tracking solutions. Essential for the algorithm is that 3dof data reflecting the position of fingers (referring to the location where the markers are positioned to track the finger movement) and 6dof data reflecting the movement of the palm, located in the same coordinate system, tracked simultaneously and delivered as one set of data, is provided for input (see chapter 6.6). To illustrate our algorithm and its steps, we pursue with our concrete realization combining the optical tracking system of the company A.R.T. with our data glove.

The algorithm receives input from the tracking system consisting of a set of 3dof and 6dof data, which is streamed via UDP. The 3dof data describes a 3d position located in the “room coordinate system” of the tracking system, the 6dof data describes a 3d position and the orientation of a target, describing the deviation from the target orientation during the beginning of the target calibration process. The orientation is described with the rotation around each one of the three axes of the room coordinate system which is reported by the tracking system with a rotation matrix.

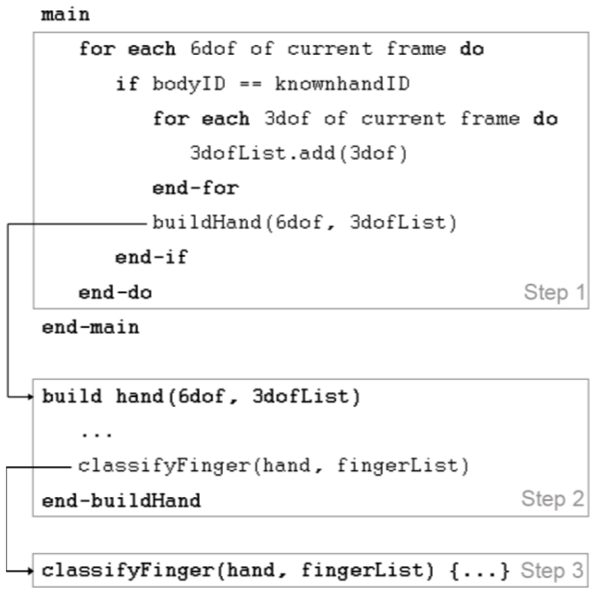


Figure 37: Structure of the algorithm to classify the hand and the corresponding fingers.

Each UDP packet delivers a frame containing data of all tracked objects. Our finger classification algorithm extracts the 3dof and 6dof data from the received frames and processes it to detect the hand, the fingers and determines what 3dof data describes which finger. This process of identifying and classifying the hand and the fingers from the input data can be divided into three consecutive steps (see Figure 37):

1. Identification of the palm target, potential finger marker and extraction of the corresponding 6dof and 3dof data
2. Noise removal and assignment of finger data to the corresponding palm
3. Classification of the finger data into index, middle, little finger and thumb

This general approach of the hand- and finger classification is similar to the one described in [Hardenberg & Bérard 2001], where video images containing bare hand movements are processed to: 1) find finger objects, 2) analyze which objects belong to the same hand and to filter finger-like objects and 3) sort finger objects according to their relative geometrical position and classify them according to their position relative to the palm and relative to each other.

The two approaches differ in the hand shapes that can be recognized by subsequent steps. Whereas with the approach described in [Hardenberg & Bérard 2001] two-dimensional projections of the hand parallel to the image plane can be identified, our approach provides output which can be used to identify three-dimensional hand shapes and the orientation of the hand (for a description of the provided output data see chapter 6.3.2). Therefore the range of hand gestures, which can be used for gestural interaction techniques is much wider, and gives interaction designers a greater flexibility when designing hand gesture interaction techniques.

Step 1: Identification of the Palm Target, Potential Finger Marker and Extraction of the Corresponding 6dof and 3dof Data

If the frame with the input data includes 6dof data, the associated id is inspected whether it is the id of a known palm target. In that case the 6dof data and the whole set of 3dof data within this frame are extracted. Each single submitted 3dof data set is considered to potentially originate from a finger marker.

Step 2: Noise Removal and Assignment of Finger Data to the Corresponding Palm

Figure 38 shows the pseudo code for the algorithm performing step 2, the noise removal and assignment of the finger data to the corresponding palm respectively hand. References in the following description refer to this figure.

```
buildHand(6dof, 3dofList)
  hand = 6dof
  hand.setAllFingers('not classified')          a
  for each 3dof in 3dofList do                  b
    if 3dof is to remote
       continue with next 3dof                  c
    finger = 3dof                               d
    translate and rotate finger                 e
    if finger is targetMarker
       continue with next 3dof                  f
    fingerList.add(finger)                      g
  end-for
  hand = classifyFinger(hand, fingerList)       h
end-buildHand
```

Figure 38: Noise removal and assignment of fingers to the corresponding palm

(a): at the beginning a new hand object is created based on the 6dof data of the palm target. The rotation matrix thereby describes the orientation of the palm of the hand and the 3D position of the target describes the location of the palm of the hand within the room coordinate system (the location of the palm is defined as the position of the marker located on top of the joint connecting the index finger with the palm of the hand). The fingers objects of the newly created hand object are marked as not classified.

(b): Each 3dof data set is processed in the same way to eliminate noise and identify data originating from markers placed on the fingers of the data glove:

First the Euclidian distance between the 3dof data and the palm is calculated and verified if the distance lies within a predefined threshold. This threshold depends on the size of the data glove and should be chosen as the maximum distance of a finger marker from the origin of the palm target enlarged by a certain amount to compensate for tracking inaccuracy. (c): if the distance is larger than the threshold the 3dof data originates from other non-glove related markers located in the tracking area and is discarded, (d): otherwise the 3dof data is used to create a new finger object. The 3dof data is used to define the 3d position of the finger object. Note that the threshold used for this comparison also defines how close two data gloves can be located beside each other before finger marker of the gloves can no longer get correctly assigned and the classification of fingers fails.

(e): the 3d position of the newly created finger object is duplicated. While one 3d position stays unchanged the other duplicated 3d position gets translated and rotated to describe the position the finger would inhabit if the palm of the corresponding hand would be placed at the origin of the “room coordinate system” and the orientation of the palm would match the orientation during the start of the data glove calibration process (see Figure 39). The 3d position of the palm of the hand is used for translation while the orientation of the palm of the hand described with the rotation matrix is used to rotate the 3d position of the finger object. Each finger object therefore holds information on its position in the “room coordinate system” and its position in a local hand coordinate system which is hand rotation and hand position invariant. Those transformed finger positions can ease subsequent modelling and are also essential for classification of the fingers and noise identification.

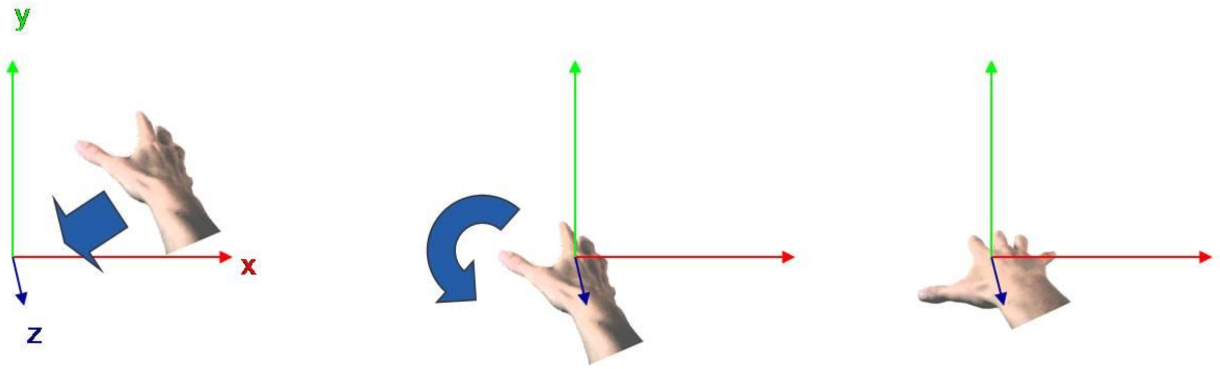


Figure 39: Translation (left) and rotation (middle) of 3D finger positions and the resulting posture and position of the hand object (right).

Even after 3dof data located too far away from the hand has been discarded, the remaining finger objects may still result from noise in the 3dof data set. Delivered 3dof data may not only result from single markers but also originate from markers belonging to the target. Normally, the A.R.T. tracking software identifies marker belonging to a target and excludes their 3dof data from the output data. However, in some cases the marker belonging to a target are not correctly identified and they appear as 3dof data in the output data, additional to the 6dof target data [ART 2006].

In order to be able to identify 3dof data resulting from markers belonging to a target, information on the position of the target markers is needed. During each target calibration process the A.R.T. tracking software calculates and saves the position of the markers in relation to the target origin and the target orientation during the beginning of the calibration process. We use this information, provided by the A.R.T. tracking software, on the position of target markers to identify data originating from target markers.

(f): each translated and rotated finger position is compared with known target marker positions. A certain tolerance range is considered in the comparison, as an exact comparison would probably not reveal a match (the fact that the tracking system was not able to identify the 3dof data to originate from a target marker implies that the corresponding position may not be located exactly at the known target marker positions). If a finger position is identified as the position of a target marker the finger object gets discarded, (g): otherwise the finger object is placed in a list of finger objects. (h): the list of finger objects and the hand objects serves as input data for Step 3 of our finger classification algorithm.

Step 3: Classification of the Fingers into Index, Middle, Little and Thumb

In the previous two steps noise in the input data has been identified and eliminated. The position and orientation of the tracked hand has been identified and the position of fingers belonging to the hand.

What is left is the association of the finger positions to the finger from which they originate. At this state we only know “finger positions”, but not which position belongs to the thumb, the index, middle and the little finger. We are able to say “one of the fingers is located at position (x_1,y_1,z_1)” but not “the finger located at position (x_1,y_1,z_1) is the thumb”.

Unlike gloves with build-in sensors, which not only measure the bending of the finger but also indicate which finger is measured, the markers used for optical tracking of the finger positions give no indication which finger they measure. Hence we only have finger positions and have to consider the range of motion for each single finger to determine the finger associated with each position.

In step 2 the positions of the finger objects have been translated and rotated. In applying this transformation the resulting positions are located in a “local hand coordinate system”, and are hand orientation- and hand position-invariant. The position of the palm target is placed at the origin of the local hand coordinate system and the fingers are arranged in a way as if the hand would be held perfect alignment to the axis of the coordinate system (illustrated in Figure 40). Whereas it is mandatory that the axes of the “local coordinate system” are parallel to the ones illustrated in Figure 40, the origin does not necessarily have to be at the illustrated location.

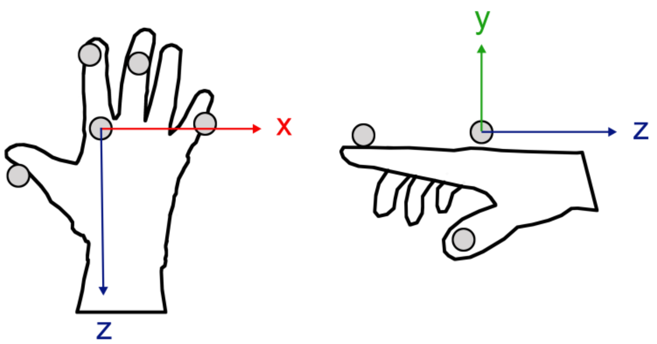


Figure 40: Position and orientation of the palm for classification of the corresponding fingers

With this transformation of the observed actual finger positions, the range of motion for each finger position depends only on the angular movements of the fingers. Influences from other movements, for example in the joints of the wrist, elbow, shoulder etc. have been eliminated. The relevant angular finger movements are: flexion, extension, abduction, adduction and opposition (see chapter 3.3 for a description of these movements and Figure 8 on page 13 for an illustration). Knowledge on the anatomical angular movements of the fingers which serves as biomechanical constraints, knowledge on the position of the markers on the textile glove, and heuristics on finger movements are used to associate a finger type to each of the finger positions, hence classify the fingers. The pseudo-code for the finger classification algorithm is shown in Figure 41 and described in the following section. References in the following description refer to this figure.

```
    classifyFinger(hand, fingerList)
a     if fingerList.size != 4 return hand
      hand.little = finger with MAX(x)
b     remove little from fingerList
      numFingersInThumbReach = 0
      for each finger in fingerList do
        if (finger.z > minZValueThumb) numFingersInThumbReach++
c     end-for
      if (numFingersInThumbReach == 1)
d       hand.thumb = finger with MAX(z)
      else
e       hand.thumb = finger with MIN(x)
      end-if
      remove thumb from fingerList
f     if x_dist(MIN(finger.x), MAX(finger.x)) > minDistNeighbouringFingers
        hand.index = finger with MIN(x)
g       hand.middle = finger with MAX(x)
      else
h       if y_dist(MIN(finger.y) - MAX(finger.y))*(-1) < minDistFingerUnderneath
          hand.index = finger with MIN(x)
i         hand.middle = finger with MAX(x)
        else
          hand.index = finger with MAX(y)
j         hand.middle = finger with MIN(y)
        end-if
      end-if
      return hand
    end-classifyFinger
```

Figure 41: Finger classification algorithm

(a): First it is assured that the list of finger objects holds the exact number of finger objects. If this is not the case no finger classification is performed. The algorithm classifies fingers in relation to the position of the other finger objects and correct finger classification is only possible if all finger markers have been tracked. The likelihood that no classification can be performed, due to the fact that too little or too many finger objects have been identified, depends on the setting of the optical cameras and the hand movements performed by the user. Once a marker is occluded and cannot be tracked by the cameras the classification cannot be performed, similar if more marker are tracked, for instance if other markers are located very near to the hand.

This drawback of our algorithm (the dependency on an exact amount of tracked fingers) can be minimised in activating a “finger memory option” which keeps state of previously classified hands and its fingers and includes this knowledge in the classification (the so called “finger memory option” which will be described in chapter 6.3.3).

(b): The little finger is the first one that can be classified, as the finger object with the largest x-value. The possible angular finger movements restrict other fingers to take on larger x-values in their transformed positions in the local hand coordinate system (see Figure 40 for the referred local hand coordinate system).

(c): Amongst the remaining three fingers, the thumb is classified next. Therefore the z-values of the three finger objects are inspected and verified if they are large enough to fall within the reach of the thumb, hence are larger than the smallest z-value the thumb can take on. This threshold has to be measured beforehand for each glove (like the other thresholds used for comparison and described in the following sections). (d): if the z-value of only one finger object falls within this range it is classified as the thumb, (e): if the z-value of more than one finger object falls within this range, the finger object with the smallest x-value (the most left finger) is classified as the thumb. Classifying the most left finger object in this constellation as the thumb is based on a heuristic that considers which finger postures are more likely to be adopted by the user. Larger z-value of the index and middle finger are easier and more likely to be achieved with the thumb positioned left to the index finger than with the thumb positioned between the index and middle finger. On the left in Figure 42, the finger posture is shown where this heuristic fails and data originating from the index finger marker would wrongly be classified as data originating from the thumb. As can be seen, this is not a commonly used posture.

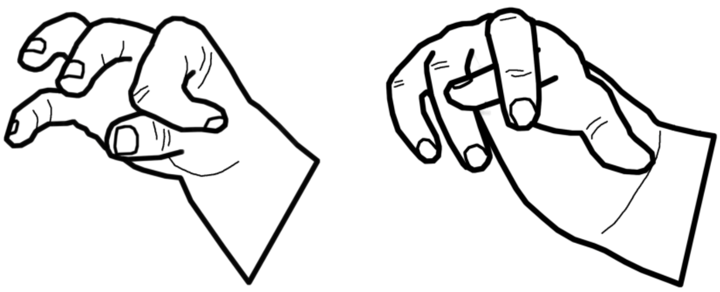


Figure 42: Finger postures where the classification fails

(f): To classify the index and middle finger in the remaining two finger objects the distance of their x-values are compared against the threshold "minDistNeighbouringFingers". The value for the "minDistNeighbouringFingers" threshold is the largest of the values for the distance(x-value of the index finger, x-value of the middle finger) measured for the three finger postures:

1. The index finger is positioned beside the middle finger, with the two finger sides aligned and touching
2. The index finger is moved underneath the middle finger and positioned as far right as possible with the middle finger moved as far as possible to the left
3. The index finger is moved above the middle finger and positioned as far right as possible with the middle finger moved as far as possible to the left

If the distance falls below this threshold it is possible that the two fingers have been moved underneath each other, otherwise the index finger is positioned left to the middle finger and (g): the finger object with the smaller x-value is classified as the index finger and the finger object with the larger x-value is classified as the middle finger.

(h): if the x-distance of the two finger objects falls below this threshold, the y-distance of the two objects is compared against the "minDistFingerUnderneath" threshold. The value for the "minDistFingerUnderneath" threshold is the largest of the values for the distance (y-value of the index finger, y-value of the middle finger) measured for the two finger postures:

1. The index finger is moved underneath the middle finger and positioned as far right as possible with the middle finger as far left as possible. Furthermore, the height distance between the two fingers should be minimised.
2. The same posture as the previously described one, just with the index finger moved above the middle finger.

If the y-distance of the index and middle finger fall below this threshold, it would not have been possible that the two fingers have been moved underneath of each other, (i): hence the index finger still holds the smaller x-value, and the index and middle finger are classified accordingly.(j): If the y-distance of the two finger objects is larger than the threshold, the finger object with the higher y-value is classified as the index finger and the finger object with the smaller value as middle finger. This classification is based on a heuristic that considers which postures are more likely to be adopted by the user. Therefore an index finger positioned above the middle finger is considered to be more likely than the index finger being positioned underneath the middle finger. The index finger is for example positioned above the middle finger, when the index finger is used to point out things positioned to the right of the pointing person. The posture where this heuristics fails and data originating from the middle finger would wrongly be classified as originate from the index finger is shown right in Figure 42. As can be seen, it is a rather unlike posture. Placing the marker or the middle finger onto the outer phalanx instead onto the middle phalanx may cause this heuristic to fail easier, as the outer phalanx of the middle finger can be moved easier above the outer phalanx of the index finger compared to the middle phalanx of the middle finger.

6.3.2 Summary and Discussion

The hand and finger classification algorithm described above uses information on the orientation of the palm to reduce the complexity and diversity of user movements that can influence the position of single fingers in the tracked three dimensional room. Due to this noise can be eliminated and the position of finger markers can be associated to the corresponding fingers. After the hand and finger classification has been processed, the following information is available for further use:

- 3d position of the palm of the hand in the "room coordinate system". The position of the palm of the hand thereby refers to the position of the marker placed on top of the joint connecting the index finger with the palm of the hand.
- Orientation of the palm of the hand, hence orientation of the hand, describing the rotation along the three axes of the "room coordinate system".
- 3D position of the thumb, index finger, middle finger and little finger in the "room coordinate system" and in the "local hand coordinate system". The position of the thumb and index finger refer to the marker placed on top of the outer phalanx, the position of the middle and little finger refer to the marker placed on top of the middle phalanx.
- Information if a finger could not be classified

The algorithm allows a fast classification, where data of each finger and the hand gets tracked and processed with a frequency of 60 Hz (imposed by the tracking frequency of the used optical tracking system). Once the needed thresholds are measured, the algorithm can be applied without any additional calibration process for several users whom the textile glove fits, which reduces the device acquisition time to the time needed to put on the glove.

A critical factor influencing the quality of the classification is the setting of the cameras and the visibility of markers for the cameras while the user is interacting in front of the display. In the above-described approach it is necessary that exactly four fingers are tracked, otherwise the finger classification will not be processed. While this increases the reliability of the classification, it also makes the algorithm prone to occlusion of markers.

The hand and finger classification with its association of finger to the delivered finger positions can provide input data for a subsequent gesture recognition process. If the recognition of some gestures only requires data on the position of a few but not all fingers, it would be desirable to perform finger classification even if not all finger markers are visible. Excluding unneeded finger markers, could also reduce the sensitivity against non-visible markers as non-relevant finger marker (for recognizing a gesture) can thereby not impede finger classification.

Two alternations, which can address the sensitivity of the algorithm against markers that are not visible for the tracking cameras have been implemented and will be described in the following chapter. Those two alternations aim at increasing the quality and reliability of the finger classification.

6.3.3 Increasing the Quality and Reliability of the Finger Classification

In the previous chapter it has been pointed out that the introduced finger classification process is prone to missing values for finger positions. When used with an optical tracking system such missing values can be present when markers are not visible for the tracking cameras. If only one marker is not tracked, no classification is processed. As a consequence only data on the hand but no position data of its finger is available for further processing (assuming that the palm is tracked).

If for the following gesture recognition only a few but not all finger positions are relevant, and some finger might even not be needed for any of the used gestures it would be desirable if the influence of unneeded finger markers could be reduced. Excluding finger markers or assign finger to positions even if not all finger markers are tracked could increase the quality and reliability of the finger classification and the following gesture recognition.

This has motivated the design and implementation of two alternations. The "two finger glove" only uses and classifies markers for fingers needed to recognize the used gestures and gets rid of the other markers (at the glove and in the classification process). The "finger memory option" allows the assignment of tracked 3d marker position to the corresponding finger, even if not all finger markers have been tracked. This "finger memory option" can be used in combination with the four- and the two-finger glove versions.

Two Finger Glove

For the two-finger glove the markers on top of the little and middle finger are removed from the textile glove, for cases where subsequent gesture recognition only requires information on the two remaining fingers. Therefore influences from markers not relevant are eliminated and a potential occlusion of irrelevant markers cannot impede finger classification. Due to the use of counter sunk bolts the irrelevant markers can be easily removed from our glove, without the need of having to build a new glove. To support a two finger glove step 3 of the algorithm, illustrated in Figure 41 on page 67 and described in chapter 6.3.1, section "Step 3: Classification of the Fingers into Index, Middle, Little and Thumb", is modified. The initial requirement with the number of finger objects in the list (a) is altered so that, depending on the glove version (2 vs. 4 fingers), the corresponding number of finger objects has to be tracked. The assignment of the little finger is skipped (b), and after classifying the thumb the remaining finger gets identified as index finger, hence the identification of the index- and middle finger is skipped as well (f - j).

Finger Memory Option

The finger memory option uses of the characteristic of the tracking system that, even so single markers do not get assigned a unique id, markers maintain the same id for the period of time in which they are constantly within sight of the tracking system cameras. Once 3dof data is assigned to a finger object and the type of finger is classified, the id of the marker from which the 3dof data originates is kept in the memory. During the next classification process the id of the newly created finger objects are compared to the ids of the previously classified finger objects. If a match is found, the new finger object is immediately identified as the finger type in which the marker was classified in the previous classification process, and the finger object is not added to the list of fingers. Therefore Step 2 has been modified, where the finger objects are created and added to the list of finger objects (see Figure 38, on page 64). Furthermore Step 3 in the classification process (see Figure 41, on page 67) has been modified to take into account already identified fingers. The amount of identified fingers is added to the number of objects in the list of finger objects. The resulting number of tracked finger objects is then compared with the required number of tracked finger objects (a). Furthermore, before the section for classification of each finger type is carried out, it is checked whether the finger type is already identified and, if so, the corresponding section is skipped.

Tradeoffs

Both alternations can reduce the sensitivity of the algorithm against occlusion of markers.

The two-finger glove explicitly excludes markers not needed for the following gesture recognition. Which minimizes the amount of markers that could be occluded and reduce the reliability of the finger classification, but also reduces the amount of information available for gesture recognition.

The finger memory option allows assigning of data to fingers, even if some finger markers could not be tracked and data on their position is missing. Furthermore, the memory option can help to avoid misclassification of fingers for postures where the two applied heuristics fail (for a description of the applied heuristics see chapter 6.3.1, section “Step 3: Classification of the Fingers into Index, Middle, Little and Thumb”). The two affected finger postures are illustrated in Figure 42 on page 68. If the user performs one of those postures, and the finger memory option is activated, the id of the thumb, index and/or middle finger could be known from previously unambiguous postures and the heuristics do not have to be applied for finger classification. However, the finger memory has the drawback that, once a wrong classification of a finger occurs, it is propagated through the following frames and is resolved only when new ids are assigned to the wrongly classified finger markers. This propagation of classification errors could lead to obscure and incomprehensible system behaviour.

Reducing the Likelihood of Misclassifications with the Finger Memory Option

[Hardenberg & Bérard 2001] and [Letessier & Bérard 2004] describe systems where fingers located in front of a wall are tracked using only video cameras and no gloves for input. To eliminate misclassifications of fingers in [Hardenberg & Bérard 2001] it is suggested to choose finger-positions closest to the last known position of a finger. Letessier & Bérard [2004] apply a similar approach to classify fingers, where each tracked fingertip in frame t gets assigned the id of the closest memorized finger at frame t-1.

Inspired by the proposed approaches, we could extend our finger classification algorithm and introduce a certain range which can additionally be considered when assigning data from marker with previously unclassified ids to fingers. The position which should be assigned to a finger could be verified if it falls within a certain range of the last known position of the finger. If not, the data is not assigned to the finger, which keeps his “unclassified” state. Although not all possible misclassifications could be avoided with such an additional constrain, some of them, for example assigning data originating from the little finger to the thumb could be prevented.

Reducing the Propagation of Classification Errors with the Finger Memory Option

The duration of the propagation of classification errors could be minimized, if we utilize our algorithm used for finger classification without the finger memory, to verify the 3d positions assigned to fingers with the finger memory option. Thereby misclassifications could be detected and resolved. Such a “reassurance” mechanism could be processed if four fingers are classified, but not for a smaller number of fingers. Nevertheless, the propagation of classification errors could be automatically resolved by our finger classification algorithm, once four finger objects are visible, without the need that each marker gets assigned a new id.

What to use When

Much on the decision which option – finger memory vs. no finger memory – and which glove version – two vs. four-finger glove – should be used, depends on the quality of the tracking system and the gestures used.

As a recommendation it should be said that, if the two-finger version is sufficient for gesture recognition, than the four fingers glove version should not be used. Furthermore, if the tracking quality is good, for example if a large number of cameras are installed, and it is unlikely that markers cannot be tracked, the finger memory should be activated to reduce ambiguous postures during the classification process.

6.4 Alternative Data Glove Solutions

Besides Whitey, described in the previous chapters, there are commercial data glove solutions available, for tracking user hand and finger movement, while given users the ability to move. In the following section we will describe alternative data glove solutions, before comparing Whitey and them in the following chapter.

The A.R.T. finger tracking solution is similar to the design in a sense that it also combines an optical tracking system with marker placed on fingers and a target to reflect movements of the palm. The CyberGlove® II and the 5DT Data Glove 14 Ultra are two commonly used build-in flexion sensor based data glove solutions. A detailed description of the A.R.T. finger tracking solution can be found in chapter 3.4.2. The last two data glove solutions will be described in the following sections.

CyberGlove® II

The CyberGlove® II [Immersion 2008] is a textile glove with build-in flexion sensors to measure angular information on the fingers, wrist and palm (see Figure 43, left). The glove can be purchased with 18 or 22 build-in sensors. The design of the glove with the open fingertips allows the use of other input devices, e.g. the keyboard, while wearing the glove. Due to the use of build-in sensors occlusion of fingers is not an issue which can influence the quality of the tracked data. However, the fact that the textile glove covers almost the whole hand can raise hygienic issues. The glove is available in one size only, which might cause a bad fit for users with differing hand and finger sizes. A bad fit is not only able to disturb the user but can also influence the accuracy of the measured data if the build in sensors do not reflect the actual finger movements.

Figure 43: Left: the CyberGlove® II (taken from [Immersion 2008]). Right: the 5DT Data Glove 14 Ultra (taken from [5DT 2007]) without the needed target.

In order to track data on the position and orientation of the palm of the hand, the CyberGlove® II can be combined with a tracking solution capable of detecting the orientation of an object. Therefore data glove solutions can be combined with an optical (e.g. [ART d] [Vicon 2009]) or electromagnetic (e.g. [Polhemus 2008]) tracking system. When combining it with an optical tracking system a target consisting of a fixed arrangement of markers is placed on the back of the hand. This target then gets tracked by the cameras of the optical tracking system which delivers data on its position and orientation.

Angular information on the bending of fingers (flexion and abduction), wrist (flexion and abduction) and palm (palm-arch) is delivered by the data glove with a frequency of 90 Hz (average delay: 5 ms (1000ms/90)/2). The frequency of the deliverance of information on the position and orientation of the palm of the hand depends on the tracking system used, for example 60 Hz (average delay: 8 ms) when the CyberGlove® II is combined with the optical A.R.T. tracking system.

The system requires a calibration session for each new user.

Information on the weight of the glove is not available but from the design and the components used it can be estimated to be above 60 g.

5DT Data Glove 14 Ultra

A data glove, similar to the CyberGlove® II is the 5DT Data Glove 14 Ultra [5DT 2007] (see Figure 43, middle). It differs in the number of sensors with 14 vs. 18 or more for the Cyber Glove® II. The reduced number of build-in sensors is reflected in the lower number of information measured.

Angular information on the bending of fingers (flexion and abduction) is delivered by the data glove with a frequency of 75 Hz (average delay: 6ms). Information on the position and orientation of the palm of the hand can be gathered in combining the data glove with an additional tracking solution similar than described for the CyberGlove® II.

The glove weights approximately 300 g.

For the glove the same benefits and drawbacks as for the CyberGlove® II are evident. As there are: use of other input devices while the glove is worn due to open finger tips, hygienic issues caused by the fact that the textile of the glove covers most of the hand, the possibility of a bad fit due to a one sized glove.

6.5 Comparison

We introduced Whitey as a data glove solution for tracking users hand movements. Whitey combines a fixed arrangement of markers and targets attached to a textile glove, with an optical tracking system. A finger classification algorithm delivers information on the 3d position of the finger tips located in a “local hand coordinate system”, 3d position of the finger tips and the palm located in a “room coordinate system” which describes the room in which the user interacts, and rotation information of the palm.

With its current implementation Whitey introduces a similar short delay than the other data glove solutions (Whitey: 9 ms resulting from 8 ms for tracking the target and markers + estimated 1 ms for the finger classification algorithm, A.R.T. finger tracking solution: 25 ms, CyberGlove® II: 8 ms, 5DT Data Glove 14 Ultra: 8 ms). With the exception of the A.R.T. finger tracking solution, those delays can be further reduced in using an optical tracking system which can track objects with a frequency of more than 60 Hz. With 9 ms Whitey falls below 20 ms, a delay between the user input action and the resulting feedback which is not noticed by the user [ISO 9241-410 2006] and leaves room for gesture recognition and other subsequent steps before feedback on the user input is given and perceived by the user.

The available data on the position and orientation of the palm of the hand is identical for all considered data glove solutions. However, considering the information available for the fingers, things look different. Whitey delivers information on the position of the fingertip in relation to a local hand coordinate system and in relation to the room in which the user operates. The A.R.T. finger tracking solution, not only delivers information on the position of the finger tips, but also the orientation of the finger tips and the angles on the bending of the fingers. The two gloves with build-in flexion sensors (CyberGlove® II, 5DT Data Glove 14 Ultra) do not deliver information on the position of the fingertips, but angular information on the bending of fingers. The CyberGlove® II furthermore provides information on the abduction of fingers. The CyberGlove® II and the 5DT Data Glove 14 Ultra track all five fingers, the A.R.T. finger tracking solution can track three or five fingers. With Whitey two or four fingers can be tracked. While Whitey delivers sufficient information for the context and the gestures it is used for in this thesis, the other data glove solutions can deliver more or other kinds of information (five fingers, bending of fingers) which could be important to consider in other settings and usages. Also the main disadvantage of Whitey compared to the data gloves with build-in sensors (CyberGlove® II, 5DT Data Glove 14 Ultra) is its sensitivity against occlusion of finger markers and resulting missing values in the derived information on position of the fingertips.

Hygienic issues are best addressed with the A.R.T. finger tracking solution. Whitey comes second; as the textile glove can be easily swapped, washed and new gloves build fast. Accuracy and reliability issues, arising from a bad fit of the data glove are best addressed by the A.R.T. finger tracking solution. With the size independent arrangement of markers and a target and the finger classification algorithm which can be adapted to different sized gloves in measuring the thresholds needed for finger classification for each different size, Whitey comes second. Therefore the A.R.T. finger tracking solution and Whitey are superior to the other systems considering hygienic and the ability to customize it to different hand sizes.

Whitey requires one initial set-up (which we could also call calibration) per glove to measure the thresholds needed for the finger classification algorithm. Therefore the device acquisition time can be reduced and different users can almost instantly use Whitey. Once a glove which has been calibrated fits a user, the corresponding thresholds can be applied for classifying the fingers of this user.

The glove with the attached marker in its described design weights 51 g, and is therefore the lightest glove when compared to the other data gloves. As the additional weight of a data glove influences fatigue when interacting with hand gestures in mid-air, providing a light weighted glove can help to reduce fatigue as compared to heavy-weighted data gloves which can cause fatigue more quickly.

Due to the placement of the markers on top of the fingers tips, and not above or in front of them, other input devices, such as a keyboard, can be used by the user while wearing the glove.

The main motivation for the design of a novel data glove solution was the restriction of user movements we observed with the A.R.T. finger tracking solution (see chapter 5.5.2). When used in front of the Powerwall of the University of Konstanz, where six optical cameras have been installed, users were restricted to stand at a static predefined position in order to track the hand movements. Although already positioned at one place, moving hands to a great extend towards the right-hand or left-hand side of the user could trigger some of the optical cameras ineffective and lead to insufficient tracking quality.

In an informal user study, we observed that Whitey increased user mobility, and users were not bound to one static position for interaction. Even so a smaller number of cameras (4 instead of 6) and the same optical tracking system have been used as for the formal evaluation study, where we observed the limitation of user movements (we will provide a detailed description of the conducted informal user study in chapter 8).

6.6 Summary of the Scope of Generality

We implemented Whitey to facilitate the optical tracking system developed by the company A.R.T. to track the 3D position of markers placed on fingers and the 3D position and orientation of a target placed on the back of the palm. However, the finger classification algorithm can also be combined with other tracking technologies able to provide the required input data: 3dof position of the fingers (referring to the position we placed the markers onto the finger) and 6dof data of the palm, all located in and referring to the same coordinate system a unique identifier for the data on the palm, and the 3dof and 6dof tracked simultaneously and combined to one set of data. Using a non optical tracking solution could thereby reduce the sensitivity against occluded markers.

To apply the finger classification algorithm, the arrangement of markers or sensors depending on the tracking solution, capturing finger movements has to follow our proposed design. For the target, or sensor, it is essential that it captures the movement of the palm, in order to be able to only consider influences from movements around the joints of the fingers on the position of the fingers and no other movements such as for instance around the wrist or elbow when assigning tracked positions to the corresponding finger objects. The tracked position and rotation data should allow a transformation of the input data into a three-dimensional Cartesian “local hand coordinate system” where one axis is orthogonal to an outstretched index finger (the “x-axis” in our example implementation), one axis is parallel to the direction of an outstretched index finger (the “z-axis” in our example implementation) and the third axis forms the coordinate system (the “y-axis” in our example implementation). If the described finger classification algorithm with its thresholds should be used without any alternations, the “local hand coordinate system” should be defined as illustrated in Figure 40 on page 66. It is not mandatory to use a right-handed coordinate system or to locate the origin at the illustrated position.

However, using a left-handed coordinate system requires alternations of the finger classification algorithm.

The use of a textile glove to attach the marker to is not a necessity. However, one advantage of using a textile glove is that the thresholds used by the finger classification algorithm only need to be measured once per glove. Each user whom the glove fits, can therefore instantly use Whitey without the need of an individual training session. The users' fingers can be classified based on the thresholds defined for the textile glove. Another advantage is that the glove is faster to put on than to attach each single marker to the corresponding finger. Nevertheless, not using a textile glove can further help to reduce hygiene-related problems.

Although described for a right hand, the finger classification algorithm can also handle left hands. Therefore the arrangement of markers and the target has to be applied to a left hand. After transforming the tracked data into the "local hand coordinate system", for each set of position data the value of the axis orthogonal to the outstretched index finger has to be multiplied by -1.

6.7 Conclusion and Outlook

Whitey is an approach which can be used as an inexpensive and flexible extension for optical tracking systems to facilitate tracking of hand movements in order to explore hand gesture interaction techniques. However, if occlusion of markers hinders the classification of fingers and our proposed compensations cannot improve the classification of fingers to the desired extend, combining an optical tracking system with data-glove solutions with build-in sensors may be a more suitable solution.

We have not yet explored the ability to utilize Whitey with other tracking solutions. However, this might be an interesting aspect for future work especially when considering that non-optical tracking solutions might reduce the sensitivity against occluded markers.

7. Hand Gestures for Virtual Navigation

In chapter 5, we introduced hand gestures as a pointing device. With the proposed interaction techniques hand gestures can be used for pointing and selecting from any point and distance from the display, to enhance physical navigation for interaction on large-high resolution displays. Physical navigation describes the use of physical user movements (e.g. walking or turning) to change the viewing distance between the user and objects on the display in order to navigate within a presented information space. From a distant position users can gain an overview of the displayed information space, while moving closer to the display reveals more details and users can gain in-depth knowledge.

While physical navigation has the advantages that it can lead to a higher spatial understanding and is more efficient than virtual navigation in the context of spatial visualization on LHRDs, it also causes higher fatigue than virtual navigation and may not always be sufficient [Ball et al. 2007]. Physical navigation is limited by the user's ability to change the distance to parts of the display. This ability may be restricted by external constrains, such as furniture or the fact that some parts of the display and its containing objects always stay distant to the user, for examples the upper area of the display (see chapter 2.4).

Therefore, further interaction techniques are needed to complement or substitute physical navigation. Users have to be able to use virtual navigation additional to physical navigation to:

1. Move distant content (either parts of the displayed information or single objects) closer to their position, in order to perceive detail information
2. Increase the size of small objects in order to make their details easier to perceive
3. Decrease the size of displayed information to gain overview information

In the following chapters two hand gesture interaction techniques addressing those three requirements will be described. "Grab to drag", can be used to virtually move objects or the whole displayed information space, addressing the first requirement. "Position to zoom", is a technique that can be used to increase and decrease the visual size of displayed information, addressing the second and third requirement.

7.1 Moving Distant Content Closer to the User

In the previous section it was argued that interaction techniques are needed which can be used to move distant parts of the displayed information and distant objects closer to the user's position. Although not specifically targeted to be used with hand gesture input, several solutions for those two questions have been proposed by researchers and can also be found in existing applications. In the following section we review related work and will describe which of those proposed virtual navigation techniques we aim to support with hand gesture input and the reasons therefore.

7.1.1 Related Work

Frisbee and Vacuum use dedicated widgets to bring distant objects closer to the user position. “Frisbee” described by Khan et al. [2004] provides a portal to other parts of a large display. It consists of two elements, a circular telescope and a circular target widget, where the telescope provides viewing and manipulation of the space covered by the target widget. Whereas the telescope is located closely to the position of the user, the target can be positioned at distant locations. Upon facilitating interaction with distant objects, the telescope also incorporates telescope and target controls.

The “Vacuum”, described by Bezerianos & Balakrishnan [2005] is a technique whereby a circular widget with an arc of influence can be invoked by the user. Proxies of objects within this adjustable arc are then pulled towards the widget invocation point. Instead of the distant objects the user may easier reach and manipulate their close by proxies.

Drag-and-pop and drag-and-pick, described by Baudisch et al. [2003] are two techniques intended to quickly drag icons to distant icons of target objects (drag-and-pop), or access distant icons (drag-and-pick) at a large display with pen or touch input. After dragging is detected, potential target icons are identified within a certain arc around the position of contact along the movement direction. The two techniques differ in the identification of the target icons. Whereas drag-and-pop is initiated when an icon is dragged and only proxies of icons where dropping would be possible are brought close, drag-and-pick is initiated when dragging is performed on empty screen space and proxies of all icons are brought close to the dragging location. Different to the Vacuum, drag-and-pop and drag-and-pick do not rely on a dedicated widget but are in an “always on” mode.

Frisbee, Vacuum, drag-and-pop and drag-and-pick are strongly connected to the underlying application, due to the need of semantical knowledge for identifying potential targets. While those approaches are targeted at direct touch or pen input, where accessing objects is limited by the physical reach of the user, the possibility of selecting objects from a distance with the introduced hand gesture point and selection techniques (see chapter 5.1 and 5.2), allows accessing distant objects or parts of the display.

Due to this possibility to assess distant parts of the object, dragging and panning can be used for bringing objects or parts of the display closer to the user position. Dragging, widely supported by different applications, is used to move objects. Panning resembles dragging, but moves not only single objects but the whole display space, for example panning of landscapes in map applications (e.g. in [Ball et al. 2007] or Google Earth [Google 2008]). Supporting dragging and panning tasks with hand gesture input could further maintain the directness of our absolute pointing technique, where the display cursor is always located in line with the direction of the palm of the hand.

The support of dragging and panning in existing applications, the fact that they are already known by many users and also their compatibility with our absolute pointing technique, motivated us to design hand gesture interaction techniques to support dragging and panning tasks. In supporting those two tasks, the stated requirement above to move distant content closer to the position of the user is addressed. In the following chapter 7.1.2 we will describe the hand gesture interaction technique we designed.

7.1.2 Dragging and Panning “Grab to Drag”

A suitable metaphor for dragging is to grab an object move and release it. Panning can be seen as being the same task only applied to a different object. To identify a suitable hand gesture we inspected natural hand grabbing movements with the aim to find a natural way of performing a dragging and panning task.

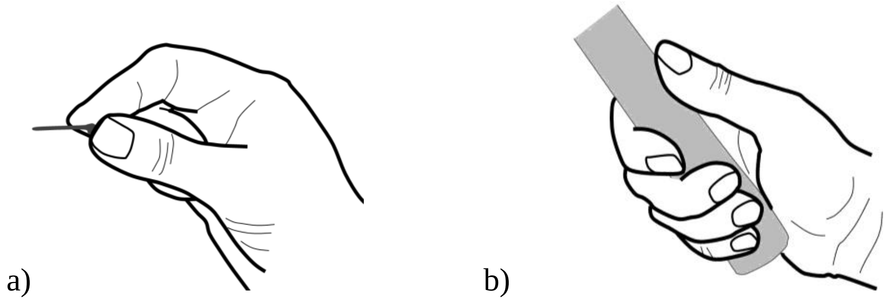


Figure 44: a) The tip pinch, a precision grip, b) a power grip. Adapted from [Jones & Lederman 2006, p. 139]

Grabbing movements performed with the hand aiming to grab a physical object can be distinguished according to the two dominant finger postures, into “precision grips” and “power grips” (see Figure 44). If a “power grip” is used, the grasped object is in contact with large areas of the inside palm and the inside of the fingers and movement is achieved by moving the whole hand and arm. “Power grips” are used when the primary objective is force, for example when using a hammer. Opposite to “power grips” are “precision grips”, where grasped objects are held between the tip of the thumb and index finger (sometimes also the middle finger). “Precision grips” are used if precise control of the grasped object and the grasping forces is aimed for. [Jones & Lederman 2006, p. 138]. Precise control of the object is what a user wants when dragging a virtual object. The similar semantic meaning and usage with physical objects in real-world interaction makes the “precision grip” hand postures suitable for being used for dragging and panning tasks.

A number of hand postures evident for “precision grips” are described by [Jones & Lederman, 2006, p. 138]. With the “tip pinch” (see Figure 44 a), one of the “precision grip” hand postures, the object is grasped between the tip of the thumb and index finger. Kendon [2004] identified a similar hand posture in the context of human-human communication. The so called “R-Family” of “precision grip gestures” is realized by moving the tip of the index finger and thumb together to form a ring. The “R-Family” gestures are used when the speaker wants to be very exact and precise about something and therefore special attention is needed. We chose this hand posture to be used as a hand gesture to invoke a selection action (see chapter 5.2). Similar to the difference between selection and dragging tasks, the two gestures – the “tip pinch” for grasping objects and the “R-Family” for being very precise about something in human-human communication – differ in the duration for which the hand posture (contact between the tip of the index finger and thumb) is maintained. When used during human-human communication, the hand posture is only maintained for a short amount of time, whereas for grabbing tasks the hand posture is maintained as long as the object is held. This difference perfectly fits in with the distinction of selection and dragging tasks, where selection is achieved within a short period of time, while dragging is carried out for a longer period.

As an analogy to manipulation of physical objects and the good fit with the proposed hand gesture selection technique, the “tip pinch” grip is used to invoke dragging and panning actions with hand gesture input. We named this hand posture “pinch gesture” when used for issuing a selection action with hand gesture input, similar to this we also refer to this hand posture as “pinch gesture” when used for dragging actions.

The two techniques (selecting and dragging, respectively panning) are combined as follows: when the pinch gesture is detected, the object at the cursor position is selected. If the pinch gesture is maintained, moving the hand can be used to move the selected object (=dragging). If no specific object has been selected the whole information space can be moved (=panning).

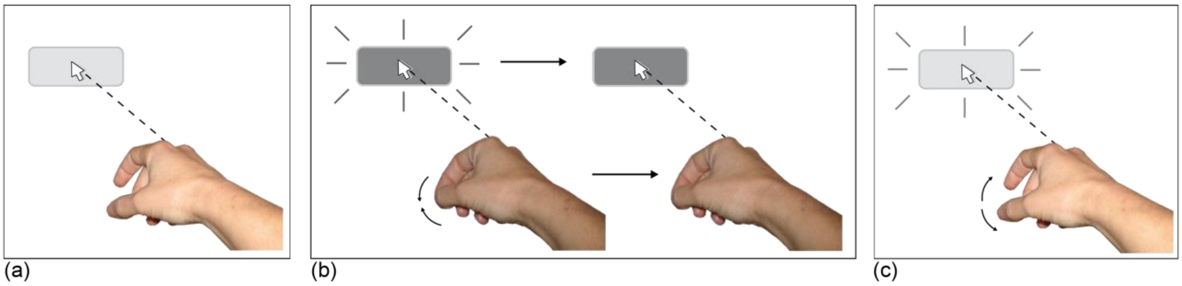


Figure 45: Dragging of an object using hand gesture input, illustrating the fluent interaction of (a) pointing, (b) selecting, dragging and (c) releasing.

Using this gesture interaction technique not only mimics similar interaction with physical objects it can also be easily combined with the already introduced techniques for pointing and selection and thereby supports a fluid interaction. The used “pinch grip” gesture further has the advantage of implicit feedback through the finger tip – thumb contact, which makes the gesture unambiguous for users and stabilizes the hand posture during movements.

The “pinch gesture” is recognized in exactly the same way as done for the selection technique (see chapter 5.2), with the distinction that it is mapped onto a left-button-down-mouse event which is released only when the pinch gesture is released. Similar to the selection technique acoustic feedback is given when the gesture is first detected.

7.2 Changing the Scale of Displayed Information

With physical navigation only, the amount of detail and overview which can be gained by users can be limited. To compensate for those limitations and allow users to not physically move for navigation we argued that techniques for virtual navigation are needed (see chapter 2.4). Therefore we identified three requirements. One of them is to provide the opportunity for users to move distant content closer to the users’ position, which we addressed with our “Grab to drag” technique introduced above. Furthermore to provide the opportunity for users to increase the size of small objects in order to make their details easier to perceive, and decrease the size of displayed information to gain overview information (see chapter 7). Those two requirements can be addressed with techniques applying a geometric zoom.

7.2.1 Geometric Zoom

Interaction techniques changing the scale of visual representations share the common idea of applying a zoom-based approach to magnify or de-magnify the size of displayed information. Zooming that changes only the scale of the visual representation is called “geometric zooming”, while zooming that changes the appearance, hence the visual representation of objects based on the amount of space available is called “semantic zooming” [Furnas & Bederson 1995]. For the purpose of changing the scale of information, in order to easier perceive details, respectively gain an overview, we will focus on a geometrical zoom which changes only the scale of the visual representation. Techniques facilitating a geometrical zoom can differ in several characteristics, for instance the number of zoom factors, the display area affected and the flexibility given to the user for specifying a zooming reference point (for an overview and a detailed description of zooming techniques see [König 2006]).

Number of Zoom Factors

One characteristic in which the geometric zooming techniques differ is the number of zoom factors provided. If only a few discrete zoom factors are provided (as for example by TapTap [Roudaut et al. 2008], a technique aimed to ease selection of small targets with finger input at small mobile devices (see Figure 46 on page 82) the user can choose one of the available zoom factors and the corresponding magnification respectively demagnification. If the number of zoom factors is higher and the distance between the factors is small the scale changes more smoothly. If accompanied by an input mechanism which allows users to continuously and fluidly change to the next zoom factor (e.g. in using the mouse wheel for input as done in Google Earth [Google 2008] or implicitly switch to the following zoom factor as long as users are pressing a predefined key as done in ZoomWorld [RaskinCenter]) users are under the impression of a smooth continuous zoom.

Using only a few predefined discrete zoom factors can be faster than using a smooth continuous zoom, if the aimed magnification or demagnification is large compared to the current scale (due to the increased distance between the zoom factors). However, the corresponding change in scale may not always match users' intend. The magnification (or demagnification, in case users want to zoom out) depends on the information the user wants to perceive and is influenced by the size of the area the user is interested in (small objects may need a larger magnification than medium-sized objects) and the distance between the user and the area of interest. Smooth zooming provides more zoom factors and thereby more flexibility to adjust the scale to match user's need.

For changing the scale of the displayed information at a LHRD, with the aim to compensate for the limits of physical navigation or substitute physical with virtual navigation, applying a smooth continuous zoom seems to be better suited. Additional to the advantage that a smooth zooming can be used to fine-tune the scaling factor to meet the users need of magnification respectively demagnification, a smooth zooming also mimics physical navigation, where moving towards or away from objects also smoothly scales the perceived amount of details.

Display Area Affected

Another characteristic in which geometrical zooming techniques differ is the amount of display area affected by the zoom. Zooming can either affect only parts of the displayed information, for example a distinct object or area, or the whole displayed content. Expanding targets[8] are objects which smoothly increase their size when the cursor is moved towards them. The closer the cursor, the larger the object gets [McGuffin & Balakrishnan 2002]. This is one example of a zooming technique enlarging only parts of the displayed content yet another example is TapTap [Roudaut et al. 2008] a technique which enlarges the area underneath the fingertip if an unsuccessful selection is detected on a touch sensitive display (see Figure 46 on page 82). Examples for zooming techniques where users can change the scale of the whole displayed contents are commonly found in geographical information systems, for example in Google Earth [Google 2008], Nasa World Wind [Worldwind 2006], or [Ball et al., 2007].

---

[8] Different interactive implementations of expanding targets can be found at http://www.dgp.toronto.edu/~mjmcguff/research/expandingTargets/ (last accessed on Jan. 15, 2009)

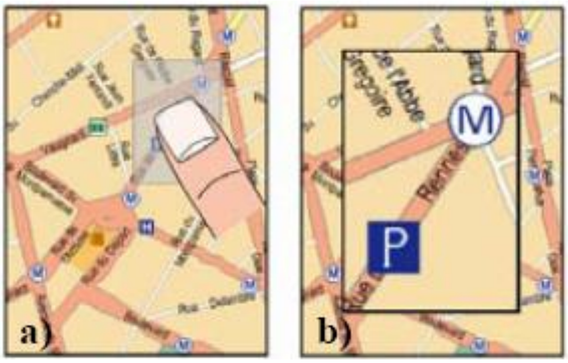


Figure 46: TapTap, where a fixed zooming factor is used to magnify a distinct area of the screen (taken from [Roudaut et al. 2008])

Changing only parts of the displayed content is sufficient if users are interested in small areas. It fails when users are interested in larger areas, including not only a dedicated object but also its context, or if users want to decrease the whole displayed content to gain a global overview. Both tasks can be supported by applying a change in scale to the whole content being displayed.

Zooming Reference Point

During zoom, the display content is scaled in reference to a zooming reference point. If this zooming reference point is located in the centre of the display (e.g. Google Earth [Google 2008]), objects located at the outer part of the display move closer to the display boarder during zooming in until they are eventually moved outside of the display due to the increased size of objects located closer to the zooming reference point. When zooming out, the size of objects located around the zooming reference point decreases, which brings objects which have been placed outside of the display back into the area that is being displayed. If the zooming reference point is not located at the centre of the display, but placed at another location, the contents of this location are brought into focus, respectively moved towards the centre of the display during zoom [König 2006] (see Figure 47).

With a fixed reference point during zooming, it may occur that objects or parts of the display the user is interested in are moved beyond the display boarder before their aimed enlargement is achieved. In such cases, the user has to interrupt the zooming task to move parts of the displayed information closer to the centre of the display (for example in panning the whole displayed information space) before the user can pursue with the zooming task. Using an adjustable zooming reference point instead of a fixed one can address this issue.

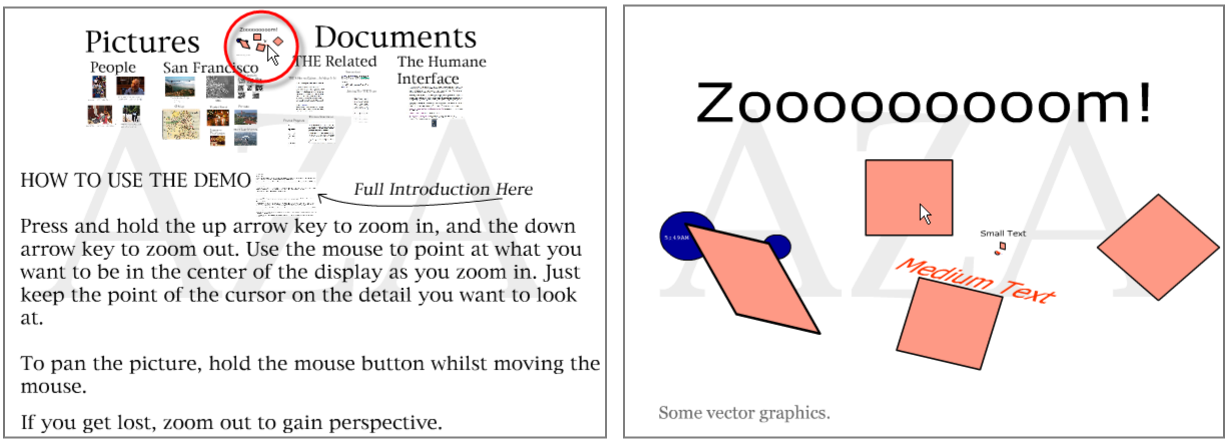


Figure 47: ZoomWorld, a zoomable user interface. The display cursor position specifies the zooming reference point and thereby determines what area is moved towards the center and brought into focus during zooming in. Left: Overview of the information objects located at the user interface. Right: Details of an area after zooming in.

ZoomWorld [RaskinCenter] is a flash prototype of a zoomable user interface, envisioned by Jef Raskin (for detailed information on Raskins' vision of a zoomable user interface see [Raskin 2000]). In ZoomWorlds' user interface, different information objects are located within a plane that infinitely expands in both dimensions (x and y). For navigation within this plane users can apply panning (for moving the plane) and zooming (for scaling the plane). Similar to the zoom technique in Google Earth, zooming in the ZoomWorld can be applied to scale the whole displayed content smoothly. However different from Google Earth, in ZoomWorlds' zooming technique the user can additionally specify the zooming reference point and further adjust it during zooming. While zooming in, the contents located close to the reference point are moved towards the centre of the display, therefore users can not only change the scale of the information but although fine-tune the area of interest without interrupting the zooming task.

7.2.2 Zooming "Position to zoom"

As a technique for virtually changing the scale of the displayed information we decided to design hand gesture input to support a zooming technique for smooth continuous zooming of the whole displayed content with an adjustable zooming reference point.

We decided to support such a kind of geometrical zooming technique due to the following reasons.

The ability to specify the zooming reference point and further adjust it during zoom, combined with a scaling of the whole displayed information does not require that users exactly specify the object which should get enlarged. Instead, the user can "roughly" specify the area of interest at the beginning of the zoom and fine-tune the area and the objects which get enlarged and brought into focus during zooming in adjusting the zooming reference point. This is especially helpful when very small, hardly perceivable and specifiable objects must be enlarged. To do so with a sufficient degree of precision can be a difficult task, owing to small size and to natural hand tremor if absolute pointing is used to indicate the location of the objects. Both difficulties can be alleviated by "roughly" specifying the area of interest and the opportunity to fine-tune the area of interest during zooming.

Given the variety of possible distances between the user's position and the area or object of interest on a LHRD, the flexibility to adjust the scaling with a large number of zoom factors with small distance can be utilized to adjust the magnification respectively demagnification to match user's intention. Furthermore, providing the zoom factors in the described way and combining it with the opportunity of continuously zooming in respectively out, closely mimics physical navigation, where user movements towards or away from the display lead to a smooth and continuous change in granularity of the perceivable information.

To support a zooming technique that smoothly changes the scale of the whole displayed content with the flexibility to specify and adjust the zooming reference point the user has to be able to specify (1) a zooming reference point and (2) the direction of zoom - zooming in vs. zooming out.

Specifying the Zooming Reference Point

For specifying the zooming reference point we use the extended index pointing gesture is used. This hand posture is commonly used in human-human communication to point out an object or location of interest [Kendon, 2004], which is semantically similar to the user's intention when pointing out the area of interest. The extended index gesture has been ruled out by us for the pointing task in favour of the palm pointing gesture, due to the higher tension and the low pointing accuracy (see chapter 5.1). However, those drawbacks do not hinder the use in this case. Using a gesture with lower tension is important for tasks that are conducted frequently, whereas a slightly higher tension is acceptable for less frequently used gestures. Furthermore, pointing accuracy is not as relevant when an area of interest has to be specified, which can be fine-tuned during the interaction, as compared to the accuracy required for pointing actions which may be followed by a selection of the target that is being pointed at.

When the user adopts the extended index gesture, the current display cursor position is considered the zooming reference point. During zooming the user can dynamically change the zooming reference point in moving his hand. The display cursor position is derived from the users hand position and orientation in the same way, as it is done for pointing: an imaginary straight line, defined by the orientation of the palm is projected and intercepted with the display. The display cursor is placed at the point of interception (see chapter 5.1 for a description and Figure 20 on page 36 for an illustration).

The extended index gesture is recognized with the approach described in chapter 4. Acoustic feedback is given to the user if the gesture is first detected.

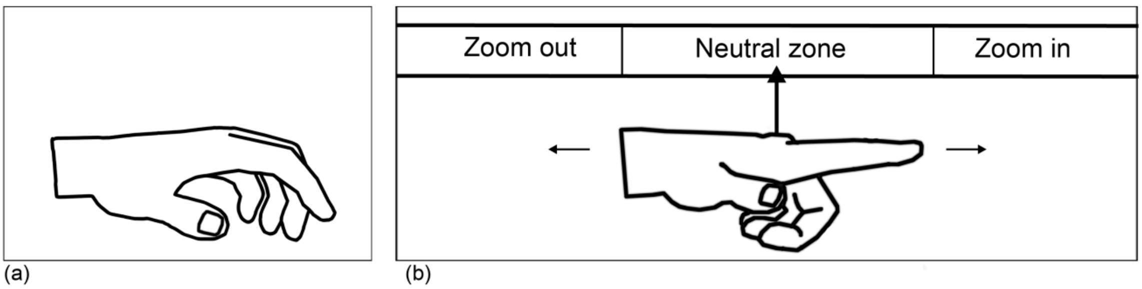


Figure 48: Hand gesture zoom technique, with movement controlled zoom mode. (a) No zooming will be issued if the zooming gesture is not recognized (b) if the user adopts the zooming gesture three distinct zooming areas are created and movements of the hand along the areas result in the corresponding action.

Specifying the Zoom Direction

Once the extended index gesture is detected, a zooming mode is activated and maintained as long as the user maintains the extended index gesture. While the zooming mode is active, the user can specify the direction of zoom with a movement of his hand towards or away from the display surface. Moving towards the surface, triggers zooming in while moving away from the surface triggers zooming out.

Using movement direction to specify the zooming direction is motivated by the observed natural behavior of people who move closer to an object when they want to see more details and step back if they want to gain an overview with less details. Thereby user behavior evident for physical navigation is reused for virtual navigation, which should ease the learning and recall effort for the virtual navigation technique.

If the extended index gesture gets detected and the zooming mode gets activated, the physical 3d position of the hand is used to divide the room in front of the display in three distinct fixed zooming areas, whose boundary planes are parallel to the display surface: 1) a “neutral area” where no zooming action is issued is created around and containing the position of the hand when the extended index gesture was detected and the zooming mode activated, 2) a “zoom in” area located ahead of the “neutral area” and 3) a “zoom out” area located behind the “neutral area”. Moving the hand into one of the areas triggers the respective action (see Figure 48). The extent of the “neutral area” can be configured according to the preferences of the user. Values ranging from 2 cm to 4 cm have been found to be pleasant during informal testing with multiple users. For stopping the zooming, users can simply release the extended index gesture or move their hand back into the range of the “neutral area”.

It is worth noting that users can, but are not required to physically move towards or away from the display surface and that small hand movements are also sufficient if the user does not want to physically move from his position.

A zoom technique applying a similar mechanism in order to specify the direction of zoom is described by Adams et al. [2008]. In their head-to-zoom technique users can trigger zoom actions in moving their head towards or away from the display, while seated in front of a regular display. The “zoom in” and “zoom out” areas (located closer respectively farer away from the display surface) are divided by a narrow neutral zone. In contrast to position-to-zoom the fixed and “always on” zoom areas can impede users in their natural movement and force them to hold their head still within a small area, if they do not want to trigger any zoom actions. This is different with position-to-zoom, where areas are created only on request and don’t interfere with other unintentional user movements.

To gain first experience and initial user feedback on our hand gesture interaction techniques for zooming, dragging and panning we conducted an informal user study. This study and our findings will be described in the following chapter.

8. Informal User Study

With the aim to gain initial user feedback and first experience on the usability of our extended set of hand gesture interaction techniques and Whitey, we conducted an informal user study.

While users interacted with different applications they were being observed by the author of this thesis. Those observations, feedback from users and own experience of the author are the basis for the findings reported in chapter 8.3.

In the next section we will describe the apparatus used. Following this we will give information on the participants and introduce the applications they interacted with and the tasks they performed. Finally, we will report our findings.

8.1 Apparatus

Our users interacted with applications running on the Powerwall of the University of Konstanz. Different to the apparatus used for our controlled experiment (see chapter 5.5.1), the optical tracking system of the company A.R.T. used four, instead of six, cameras, which were placed on top of the display on the right and left hand side to cover the area in front of the display (see Figure 49, left). This system in combination with the novel data glove solution Whitey (described in chapter 6) was used to track the movement and position of users hand and fingers.

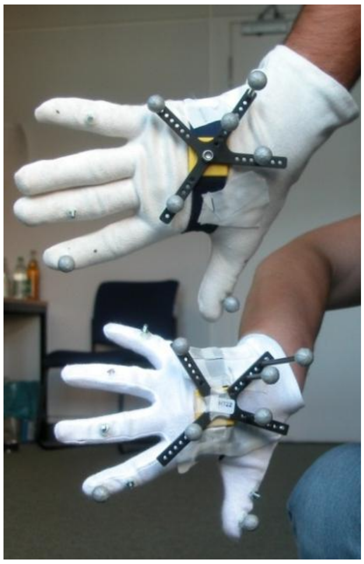


Figure 49: Left: The Powerwall of the University of Konstanz and the four cameras belonging to the optical tracking system of the company A.R.T. Right: The two data gloves used for interaction. On top the large sized and underneath of it the small sized data glove.

To achieve a good fit of the data glove, a small or a large sized novel data glove has been used for interaction, depending on the physical size of the users' hand. Those two data gloves differ in the size of the glove and the arrangement of the marker of the hand target, but are otherwise identical in their design and function (the different arrangement of the hand target marker is used to uniquely identify each data glove). See Figure 49, right for the two data gloves.

8.2 Participants and Tasks

The hand gesture interaction techniques have been used to interact with different applications performing different tasks. In this section we will describe the applications participants interacted with, the tasks they performed, provide information on the participants and on the applied procedure.

8.2.1 Pointing, Selecting and Dragging

The hand gesture techniques for pointing, selecting and dragging have been used to interact (1) with applications with traditional Windows, Icons, Menus, and Pointing (WIMP) interfaces and (2) “NipMap”, a multimodal drawing application.

Interaction with Common Applications with WIMP Interfaces

Six participants, recruited from university’s staff and students and from outside the university, were asked to perform select and dragging tasks in common applications with the hand gesture interaction technique.

Users therefore were equipped with the data glove, followed by a short explanation of the gestures and the interaction techniques. After a short training which allowed the users to become accustomed to the gestures and interaction techniques, they could move freely and interact with no predefined tasks.

As a starting point a typical Windows XP Server desktop was displayed on the display of the Powerwall. Along the desktop icons a number of MS windows explorer applications were opened, showing the contents of different folders. For some of the participants we also run Winamp [Nullsoft], an application which can be used to play multimedia files, such as music or video files (see Figure 50, left). We also let users interact with MS Paint, a drawing application coming together with Windows operation systems (see Figure 50, right).

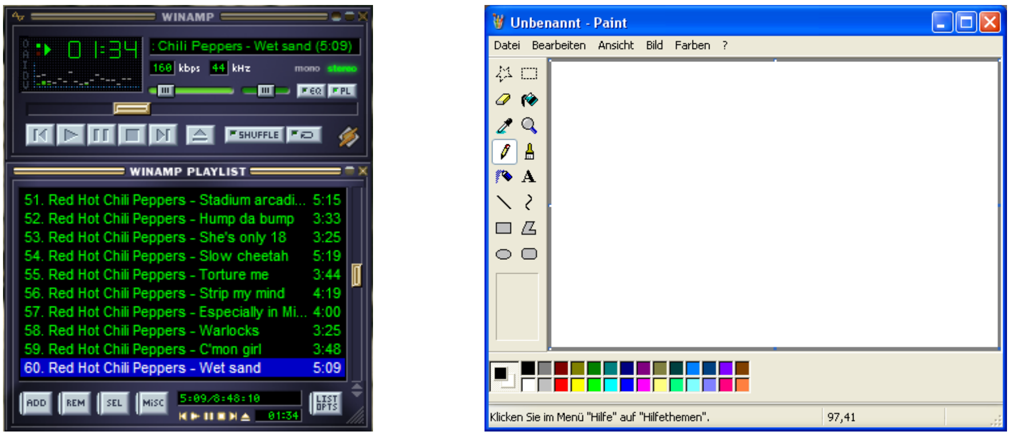


Figure 50: Left: Winamp user interface, Right: MS Paint user interface

Most of the participants started in selecting or dragging windows and desktop icons. They further dragged files from one folder to another folder, or dropped music files into Winamps’ user interface to play them. When interacting with Winamp, they paused and restarted the playback in selecting the corresponding buttons on Winamps' user interface or changed the volume of the speaker in using the slider for volume control. When users interacted with MS Paint, they draw sketches or wrote words.

The performed tasks can be found in many WIMP interfaces, where selecting buttons (Winamp) or GUI elements in general (selecting a brush in MS paint, selecting a color in MS paint, selecting a window), specifying a path (drawing sketches or writing words in MS paint, changing the position of a slider in Winamp, moving windows or files) are tasks that users commonly perform during interaction. Therefore we assume that the tasks our participants perform reflect a typical application scenario for our hand gesture interaction techniques when used to interact with traditional WIMP interfaces.

Multimodal Interaction with "NipMap"

We had 10 participants using our hand gesture techniques for multimodal interaction with "Nipper", a system which facilitates multimodal interaction at LHRDs. "Nipper" has been envisioned and developed by a student of the University of Konstanz. "NipMap", also called "Nipper's map application" is the first application of the "Nipper" system, which can be operated using multimodal input, combining speech with either laser pointer or hand gesture input. "NipMap" thereby applies a "speak and point" concept, which combines speech input with pointing actions (pointing, selecting and dragging). "NipMap" displays an aerial picture of the container harbor of Rotterdam. The user can create geometrical objects (rectangles and ellipses) and text fields to mark or label ships, containers or dedicated areas in the container harbour [Fäh 2008]. Previous research has illustrated that hand gestures can be a valuable enhancement to speech input to provide a natural way of interaction [Bolt 1980] [Lucente et al. 1998]. We use "NipMap" to explore the use of hand gestures in this promising area of multimodal interaction.

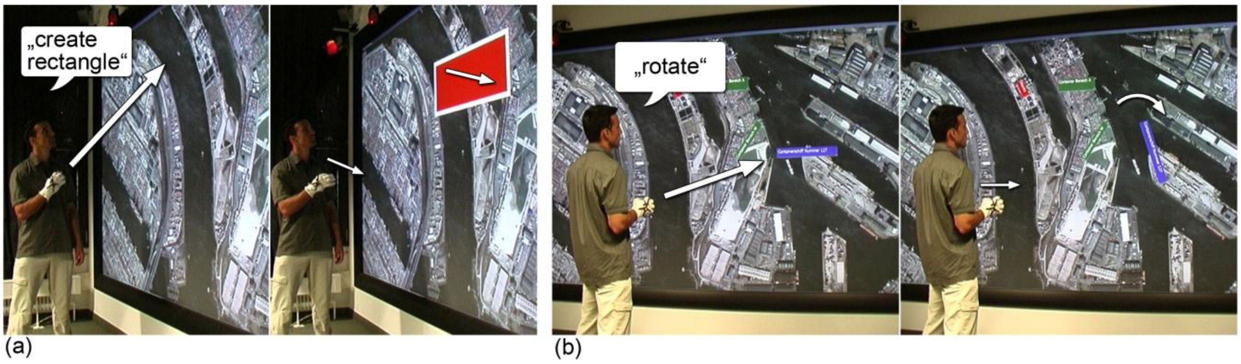


Figure 51: Examples for multimodal interaction with "NipMap". (a) Creating a rectangle. Left: Speech is used to issue the command to create a rectangle. Right: Following the spoken command, hand gesture dragging is used to specify the size of the object. (b) Rotating a text object. Left: Speech is used to issue the rotation command. Right: Following the spoken command, hand gesture dragging is used to specify the angle of rotation.

In Figure 51 two examples for multimodal interaction with "NipMap" are illustrated. Figure 51 (a) shows an example where a user creates a rectangle. To accomplish this, the user first issues a spoken command to indicate that a rectangle should be created. Following this the user can perform a dragging action with hand gesture input, to create the rectangle and specify the location and size of the rectangle. In (b) the user rotates a text field. Therefore the user first issues a spoken command to indicate that a rotation should be performed. Following this the user can rotate the text field with a dragging action that specifies the angle of rotation.

As an alternative to our hand gestures, a laser pointer, developed by members of the Human-Computer Interaction group of the University of Konstanz [König et al. 2007a] has also been used for multimodal interaction.

To evaluate how efficiently and intuitive the multimodal interaction techniques can be used an evaluation study has been conducted. Another aim of the evaluation study was to get information on how well the speech recognition system adapts to different users and how much training is needed for the speech recognition system. In the following description of the evaluation study we concentrate on the aspects concerned with the interaction techniques and leave out the parts devoted to evaluate the speech recognition system (for a detailed description of the evaluation study see [Fäh 2008]).

For the evaluation study six participants were recruited from students and members of the University of Konstanz and the University of Zürich. To get an impression on the intuitiveness of "NipMap"'s user interface, participants first performed a small training task set after getting only a very basic introduction of "NipMap"'s multimodal interaction techniques and user interface elements. Following this, a comprehensive introduction of "NipMap" was given to the participants. Each participant then performed two task sets. The task sets contained similar tasks, but were performed with different input modalities: one task set has been performed with speech input and the laser pointer and the other task set with speech input and hand gestures. The presentation of the task sets and the order of the pointing devices were varied in an alternating manner. The device used for the first task set was also used for the previous small training task set.

Participants performed tasks such as creating geometrical objects and text fields, moving them, rotate them or changing other properties, for example their background or foreground colour (the task sets used can be found in Appendix B).

Before the participants used the hand gesture interaction techniques, they have been equipped with the data glove, followed by an explanation of the gestural interaction techniques. A short training allowed them to get accustomed to the "pinch gesture" and the pointing, selecting and dragging hand gesture interaction technique.

Additionally to evaluating the "Nipper" system, we used the evaluation sessions to observe the use of the hand gestures for interaction with "NipMap" by the participants. Those observations and our conclusions will be described in chapter 8.3, section "Pointing, Selecting and Dragging".

8.2.2 Zooming and Panning

Six participants, recruited from the members of the Human-Computer Interaction group at University of Konstanz and from persons working outside University were asked to use the zooming and panning hand gesture interaction techniques to interact with the ZoomWorld application (see Figure 52 on page 90).

Zoom World [RaskinCenter] is a flash prototype application of a zoomable user interface. Different information objects are located on a plane that extents infinitely in both directions. To navigate in ZoomWorlds' user interface panning and zooming can be used. Panning moves the whole plane and its' containing objects. A smooth continuous zooming, where the user can specify the zooming reference point and further adjust the zooming reference point while zooming, can be applied to scale the whole displayed content.

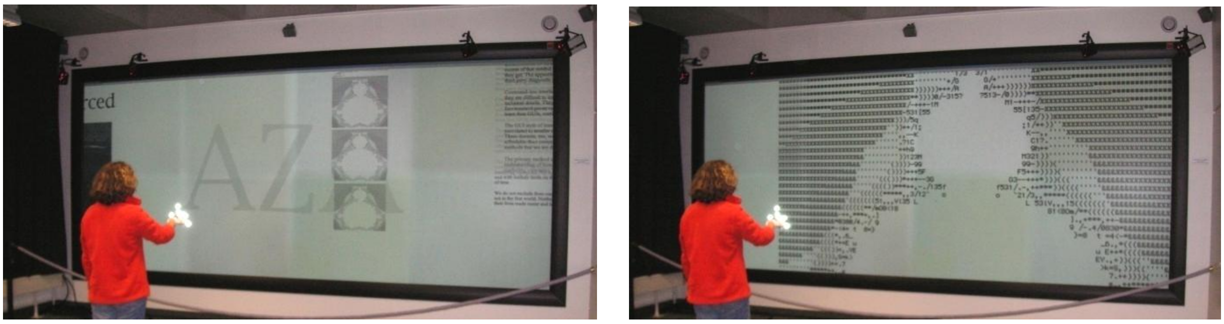


Figure 52: A user interacting with ZoomWorld, a zoomable user interface prototype application running on the Powerwall of the University of Konstanz

ZoomWorld was originally intended to be used with a keyboard and a mouse. To use our proposed zooming and panning hand gesture interaction techniques for navigation, we mapped the outcome of our zooming and panning hand gesture interaction technique to the input expected by ZoomWorld and thereby substituted the traditional input devices (keyboard and mouse) with our hand gesture input.

8.3 Findings

Whitey and the designed hand gesture interaction techniques for pointing, selecting, dragging, panning and zooming have been used by different persons with a number of applications. The materials used and the tasks performed have been described in the previous sections.

In the following sections we will describe our observations, based on feedback and observations on the use of our hand gesture interaction by 14 different persons and own experience of the author. Some persons performed all the different tasks, while others only performed parts of them. Therefore the groups of participants for each application area described above are overlapping groups.

8.3.1 Pointing, Selecting and Dragging

We found that the hand gesture interaction techniques for pointing, selecting and dragging were very easily learned and well accepted by our participants. Explaining and learning the gestures typically took 1-2 minutes. Furthermore, we observed that participants could combine the techniques fluidly during interaction. We already knew that the pointing and selecting techniques have been easily learned and performed by participants of the formal evaluation study of hand gesture performance using a commercial data glove solution (see chapter 5.5). However, we can confirm this observation for uses of the gestures in a setting with higher external validity and performed with Whitey.

Some users mentioned that they liked the acoustic feedback given when the “pinch gesture” (used for selecting and dragging) was first detected. Besides supporting the user, the acoustic feedback was also very helpful for the observer. First it was a good mechanism to initially explain the “pinch gesture” and when it is recognized by the system. Participants then usually first experimented with the “pinch gesture” while relying on the “click” sound, before starting to use the interaction technique with an application. Secondly, the acoustic feedback is a good mechanism to instantly detect when the gesture has been recognized without having to look at the participants fingers, or also detect if the gesture has not been recognized. Such knowledge was usually helpful when reasoning about why a task could not be performed and if the origin is found in the application or the input device (for instance markers which were occluded from the cameras point of view).

We found that most users were tense when they started to use the hand gesture interaction techniques, but became more relaxed with practice. Some started with completely outstretched arms, a posture which becomes tiring quickly, but changed to a more relaxed posture where the arm was bend when gathering more confidence with the hand gesture interaction technique.

According to the author's own experience, holding the hand with its back facing the right wall was more comfortable than holding the back of the hand facing the ceiling. However, not all participants used this more relaxing posture, even if it was suggested to them. The reason for that might be that they still lacked confidence in the gesture recognition system, and were tense and anxious about trying to "get it right.

This observed behaviour and the use of an outstretched arm when first using the hand gesture interaction techniques suggests that a certain amount of practice, which may be different depending on the user, is needed before users are comfortable with hand gesture interaction and start to be more relaxed when interacting.

8.3.2 Zooming and Panning

Our participants quickly learned the hand gesture interaction techniques for pointing, selecting and panning (pointing and selection is a prerequisite for performing the panning or zooming technique and is therefore explicitly mentioned). The time needed to explain and learn the gestures typically took 1-2 minutes. This is in line with our observations for the pointing, selecting and dragging techniques, where dragging technique requires the same actions by the user, with the difference that for panning the user does not have to exactly select an object but can select any part at the display for panning.

Learning the zooming technique also took participants little time. However, it took them longer than the other techniques (typically 3-5 minutes). We think this is due to the following reasons:

- Missing physical support when performing the "extended index gesture". The "extended index gesture" provides no feel able feedback as it does the "pinch gesture" through the contact of the thumb and index finger tip. It may be harder to judge and remember exactly "how much" the index finger has to be outstretched without any physical support. Beside acoustic feedback participants seemed to rely also on physical feedback when performing the gesture. Besides the feedback the contact of the two fingertips also stops finger movement, whereas outstretching the index finger has to be consciously stopped by the user. Thereby the "pinch gesture" itself provides physical guidance for the user.
- Lack of feedback on the current zooming state while users moved within the neutral area. In particular large neutral area left participants uncertain if they correctly entered the zooming state. We were able to compensate this in reducing the neutral area to a small range of approximately 2 to 4 cm. With the smaller neutral area participants could faster perceive visual feedback in observing the change in scale of the displayed content. Vogel & Balakrishnan [2005] provided additional visual feedback for their clicking gesture technique to indicate when a gesture is detected respectively no longer detected. Similar to them we could enhance our zooming technique with additional visual feedback. This visual feedback could reflect the position of the user in relation to the zooming areas and might be a way to support users during the zooming interaction.

> Especially when the users' hand moves in the neutral area where no visual feedback is provided yet (by the application).

Once users were familiar with the panning and zooming techniques, they could switch easily between the two interaction techniques and fluently navigate within the ZoomWorld application. Some participants mentioned that they liked interacting with our hand gesture interaction techniques and found them suitable for the tasks.

Considering the zooming interaction we could further observe that users tried to stop zooming by moving back into the neutral area, instead of just releasing the "extended index gesture". The same behavior was evident, when users wanted to reverse the zoom direction: they maintained the "extended index gesture" and moved their hand out of the active zoom area, across the neutral area into the opposite zoom area, instead of releasing and readapting the "extended index gesture", which would be a faster way for changing the zoom direction. With practice our participants changed this behaviour and released the "extended index gesture" for stopping the zooming action or changing the zooming direction.

However, it seems that the observed initial behaviour is a more natural way for stopping the zoom or changing the zooming direction. To better support this behaviour and provide a more natural way of interaction, the behaviour of the zooming areas could be altered. Instead of keeping them at a fixed position, based on the position when the user first entered the zooming state, the areas could be dynamically adapted to the current user position. Once a user has entered a zooming area, the opposite area boundary could follow the user in his movement and would therefore always be positioned closely behind, respectively before the actual user position, independently where the zooming state was entered. Users could then stop zooming more quickly or change the zooming direction more quickly using movement of their hand while maintaining the "extended index gesture".

Similar to the tense behavior of users who started with our hand gesture interaction techniques for pointing, selecting and dragging, most users were also tense when they had no previous experience with our hand gesture interaction techniques and were requested to start with our techniques for panning and zooming. This strengthens our assumption that it takes a certain amount of practice before users become confident and more relaxed using hand gestures for interaction.

8.3.3 On Whitey

The participants of our informal user study used either the large or small version of the data glove (see Figure 49 right on page 86 for the two data gloves). The attachment of the marker to a textile glove, should allow the use of Whitey without an initial calibration session for adjusting the system to the users hand. Thresholds needed for the finger classification and gesture recognition have to be measured once for each data glove and should fit each user wearing the glove. We observed that all participants could use those initially determined thresholds for interacting with our hand gesture interaction techniques. However, one participant wanted the gesture recognition to be less restrictive, which we achieved in altering the thresholds used for gesture recognition. We did not observe any obscure behaviour of the gesture recognition process, therefore we assume that once the finger classification and the gesture recognition is adjusted to a particular data glove it can be used by most users without the need to change the underlying thresholds.

However, what we've observed is that it disturbs users when the gesture recognition fails during tasks where a gesture is maintained for a longer time, for example when dragging an object with the “pinch gesture”. This can for instance happen if the user moves his hand in a way that markers become occluded from the cameras point of view. We could compensate this in activating the “gesture memory” option of the applied gesture recognition (see chapter 4.2) which allows the reuse of a gesture for a certain amount of time, even if no data of the finger position is delivered. Reusing a gesture for 8.333 ms (500 frames if the optical tracking system developed by the company A.R.T. is used) was found to provide the impression of a fluent interaction without causing confusion about a non-performed gesture being recognized.

For most of the participants we used the two finger version of the data glove, as the gestures we used for interaction could be recognized with the thumb and index finger marker.

Our main motivation for designing Whitey was to improve users’ range of motion for interaction compared to the restriction of user movements we observed during our formal evaluation study of hand gesture performance (see chapter 5.5.2). With the commercial finger tracking solution user movements have been restricted to a very narrow area located three meters away and in front of the middle of the display (with a set up of six cameras for the optical tracking system). We observed that participants had a larger area for hand gesture interaction when using Whitey, even with a smaller amount of cameras used for the optical tracking system (four cameras instead of six). Exact values for the area in which our participants could interact are hard to determine, as they not only depend on the data glove and the set up of the cameras but also on the individual user and his hand postures during interaction. The following values should therefore be viewed as a “rough” guess. When facing the middle of the display, our participants could approximately move as far away as 4 meters and as close as 0.5 meter from the display. When positioned as close as 0.5 meters from the display, users could approximately move from the middle of the display until they were 1 meter away from the display border. With a distance of 4 meter they could approximately move from the middle of the display until they were 2 meters away from the display border. Even if the values are only a rough guess, it could be clearly observed that with Whitey participants had a much larger area in which they could move freely while interacting. However, as mentioned above, finger markers which were occluded from the cameras point of view could impede user interaction even if users were in the field of view of the cameras.

8.3.4 Enhancements for our hand gesture interaction techniques

We observed that jitter, caused by natural hand tremor and tracking inaccuracies, made it hard to select small objects with our absolute pointing technique. To accommodate for the jitter, we applied a Kalman filter[9] on the display cursor position derived from the orientation and position of the palm of the hand. Kalman filter have been used by researchers (e.g. by [König et al. 2007b] [Oh & Stuerzlinger 2002] [Frolov et al. 2002]) to compensate jitter of the cursor when interacting with a laser pointer from a distance, where natural hand tremor also influences the display cursor. We combined dynamic and static Kalman filter to keep the cursor more stable during slow movements to ease target selection, while smoothening the path of the cursor during fast movements (for a detailed description of the applied filter see [König et al. 2007b]). We found that this Kalman filter is a valuable enhancement to the hand gesture interaction techniques, as it seemed to significantly ease the task of selecting objects and leads to smooth movements while dragging objects across the display.

[9] The Kalman filter has been developed by members of the Human-Computer Interaction group to be combined with different input devices such as a laser pointer or hand gestures.

We also observed that performing the “pinch gesture” used for selecting or dragging tasks could lead to a slight repositioning of the display cursor. This unintentional change of the display cursor position is due to correlated muscle movements in the palm of the hand which are captured by the target placed on the back of the hand. Those changes of the target result in a change of the derived display cursor position. To compensate for such unintentional changes of the cursor position we applied a so called “EasyClick” filter[10], taking the output cursor position of the Kalman filter as the input cursor position. *“[...] the EasyClick filter analyzes the previous movement path and estimates the intended position by heuristics combined with temporal and spatial interpolation”* [König 2008]. We found that the “EasyClick” filter was especially helpful when small objects should be selected and that it is, similar to the Kalman filter a valuable enhancement for the hand gesture interaction techniques.

8.4 Summary and Conclusion

With the aim to gain initial user feedback and first experience on the usability of our extended set of hand gesture interaction techniques (consisting of techniques to support pointing, selecting, panning, dragging and zooming tasks) and Whitey, we conducted an informal user study. Our 14 participants performed common tasks in applications with a WIMP interface, ZoomWorld the prototype application of a zoomable user interface and “NipMap” a multimodal drawing application.

We found that participants quickly learned our hand gesture interaction techniques and further combined them fluidly. We perceived positive feedback from our participants, which seemed to enjoy using our hand gesture interaction techniques. For our zooming technique we identified improvements in order to provide a more natural interaction and better support users during interaction. We observed that participants tend to be tense at the beginning and become more relaxed with practice, which suggests that although hand gestures might be quick to learn, a certain amount of practice is needed before users interact in a relaxed manner.

Our observations suggest that the “pinch gesture” might be easier to learn and perform than the “extended index finger gesture”. We assume that the contact of the thumb and index finger with the pinch gesture might be the reason therefore, as this contact provides implicit feedback and physical support for stopping the movement of entering the gesture. If our assumption can be confirmed in a formal evaluation study, this might be a valuable guidance for choosing suitable hand gestures for hand gesture interaction techniques.

Whitey could be used by our participants without any further adjustment, although the thresholds for our gesture recognition which we combined with Whitey had to be adjusted for one of our participants. Therefore we conclude that Whitey provides a suitable solution for tracking hand movements, at least in our setting and the used hand gesture interaction techniques.

We enhanced our hand gesture interaction techniques with a Kalman filter to accommodate for jitter caused by natural hand tremor and tracking inaccuracy. Further did we apply an EasyClick filter which compensates an unintentional slight repositioning of the cursor position while performing the “pinch gesture” used for pointing and dragging. This seemed to ease selection especially of particular small objects. Those additional filters are a valuable enhancement for

[10] The “EasyClick” filter has been developed by members of the Human-Computer Interaction group to be combined with different input devices such as a laser pointer or hand gestures.

our hand gesture interaction techniques, as they seem to improve the precision of pointing and selection tasks.

Our observations are promising and the next step should be to implement the identified improvements for our zooming technique and to conduct a formal evaluation study in order to confirm and strengthen our observations and conclusions.

9. Summary and Outlook

In this chapter we summarize our work, present our conclusions and provide an outlook for future work.

9.1 Summary and Conclusion

Large high-resolution displays require input devices that give users the freedom to move freely and interact from any point and distance. As an input device fulfilling this requirement, we introduced hand gestures.

Based on previous findings of Vogel & Balakrishnan [2005] and Kendon [2004] we identified suitable hand gesture interaction techniques for "pointing" and "selecting". To underline the analogy to real-world interaction, we provided proactive tactile feedback to the finger tips to enhance the selection task.

Our pointing technique, based on a "palm pointing gesture" combined with an absolute mapping of hand movements to cursor position, leads to an intuitive way of interaction which further allows moving the cursor fast to any location on the display. As a mechanism to signal a selection, which is *"A classic problem in device-free interaction [...]"* [Vogel & Balakrishnan 2005] we use a "pinch gesture", which is fast to perform for the user with little ambiguity from the users point of view [Wilson 2006] and can be fluidly combined with our pointing gesture.

To assess the usability of our proposed hand gesture techniques for pointing and selecting, and the influence of additional tactile feedback and movement direction we conducted a comparative evaluation study based on the ISO 9241-9. The 20 participants performed horizontal and vertical one-directional tapping tasks with hand gesture input with and without tactile feedback in front of the Powerwall of the University of Konstanz, a large high-resolution display (5.20x 2.15 m). We further asked participants to rate the non-tactile version of our hand gesture interaction technique and compare the tactile version against it. For tracking hand and movements and provide tactile feedback, participants were equipped with a commercial data glove solution. For gesture recognition we applied an algorithm, based on geometric gesture models and state dependent comparison of thresholds.

We found that the non-tactile version of our hand gesture interaction technique was very well received by participants. Additionally, the resulting effective index of performance, of up to 3 bits/s appears promising and suggests that our techniques are adequate for interaction with large high-resolution displays.

Contrary to previous research, we cannot confirm a beneficial effect of additional tactile feedback on user performance in terms of effective index of performance. Subjective user preferences were mixed, without a clear tendency for either one. Therefore we suggest that tactile feedback can be used to enhance our proposed hand gesture interaction technique for selection, but does not necessary have to. Considering the mixed results on the effects of tactile feedback, we think that further research activities are needed to identify the factors influencing the usefulness of additional tactile feedback. As a start we provided a classification in proactive and retroactive feedback, but there are clearly other influencing factors at play.

Considering movement direction, we found a significant difference in favour of the horizontal target alignment compared to the vertical one in terms of the effective index of performance. Differences in physical limb movements are considered to for being the main reason, while the combination of hand gestures used might also be an influencing factor. To compensate for the influence of the hand gestures used, we proposed to apply a filter which reverses the unintentional slight repositioning of the display cursor while the selection gesture is performed. On the influence of physical limb movements on user performance, we suggest further research activities. The correlation of movement direction to user performance might not be limited to hand gesture input, but could also be evident for other input devices held in mid-air, and therefore be a more general issue for interaction at large high-resolution display. We further made recommendations, how the influence of approaching direction for point and selection tasks could be reflected in the design of graphical user interfaces of applications running on large high-resolution displays intended to be used with hand gesture input.

During the previously described comparative evaluation study we observed that the commercial data glove solution we used limited user mobility and allowed participants to interact only from a stationary position. To overcome this limitation, but nevertheless still better address known issues of hygienic of other commercial data glove solutions, we designed Whitey. Whitey is a solution for tracking hand movements, which combines an optical tracking system and a fixed arrangement of passive markers and a target attached to a textile glove. To associate the tracked position data to the finger they originate from, it is accompanied by a finger classification algorithm, based on biomechanical constraints and heuristic knowledge of finger movements. Although we implemented Whitey using the optical tracking system developed by the company A.R.T. and a textile glove, Whitey is not limited to this setting. It can also be realized in combination with other tracking solutions or without a textile glove.

We pointed out that physical navigation, which increases with display size, has its limitations and drawbacks, and that techniques for virtual navigation can compensate for those limitations. Based on previous findings of Kendon [2004] and human behavior when interacting with their real-world surroundings, we identified suitable hand gesture interaction techniques for “dragging”, “panning” and “zooming” tasks, which are commonly used for virtual navigation.

Similar to our selection technique, we used the “pinch gesture” for dragging and panning gesture, which allows for a fast switch between those techniques supporting a fluid interaction. For zooming we used the “extended index gesture” and combined it with movements of the hand towards or away from the display.

We gained initial user feedback and first experience on the usability of our extended hand gesture interaction set a Whitey in an informal user study, which took place at the Powerwall of the University of Konstanz. 14 participants performed common tasks in applications with a WIMP interface they interacted with a zoomable user interface and a multimodal drawing application. To track hand movements we used Whitey, with two different sized gloves to accommodate different hand sizes. For gesture recognition we applied the same algorithm as for the formal evaluation study, which based on geometrical gesture models and state dependent comparison of thresholds.

We received positive feedback from our participants. They all learned our hand gesture interaction techniques fast. However, we observed that they were tense at the beginning but became more relaxed and comfortable with practice. For our “zooming” technique we identified improvements which could lead to a more natural interaction and a better user support during interaction. Our observations suggest that physical support implicitly provided by hand gestures (contact between the tip of the thumb and index finger for our pinch gesture) might improve interaction, as it can function as implicit feedback and as physical guidance of user movements.

Almost all our participants could instantly use and interact with Whitey and the thresholds defined for the gesture classification. We observed that enhancing our hand gesture interaction techniques with filters which compensate for jitter (Kalman filter), and a slight cursor repositioning (EasyClick filter) when performing the selection gesture, eased the selection in particular of small target. Therefore we suggest combining our hand gesture interaction techniques with those two filters to improve precision for pointing and selection tasks.

The results of our evaluation studies (a controlled experiment with 20 participants and an informal user study with 14 participants), feedback from users and our own experience suggest that our hand gesture interaction techniques are an adequate and valuable technique for interaction with large high-resolution displays.

9.2 Outlook

The results of our informal user study are encouraging, so the next step will be to implement the suggested improvements. A formal evaluation study could then be conducted to confirm and strengthen our findings.

So far our set of hand gesture interaction techniques consists only of one-handed gestural techniques. However, many of the tasks we perform with our hands in the real-world are bi-manual. *"We steer our car with one hand while changing gears with the other. We hold a ruler or drafting machine with one hand and use a pencil in the other. All of these tasks employ the performance of everday motor skills that have potential in human-computer interaction, but are largely ignored by computer systems"* [Buxton 1994]. To utilize this potential, we would like to explore two-handed gestural technique for distant interaction at LHRDs.

We also think that combining hand gesture with other input modalities is promising. We have already combined our hand gesture point and selection techniques with speech input to interact with "NipMap", a drawing application. Feedback we received from users and our own experience with this kind of multimodal input are encouraging. Similar to Baudel & Beaudouin-Lafon [1993] we think that therefore speech can be used for tasks without a natural mapping to hand movements. Whereas hand gestures could be applied for tasks highly suitable for hand movements, such as rotating an object or indicating a spatial location.

Furthermore, we consider investigating on the effect of input device movement direction on user performance and its underlying reasons important. The number of large high-resolution displays is expected to increase [Vogel & Balakrishnan 2005], which corresponds to a rising need for mobile input devices and suitable interaction techniques. A better insight into what sorts of movements are easier for users to perform and possess a higher degree of reliability can lead to valuable guidelines for designing user interfaces and interaction techniques for large high-resolution displays.

Appendix

Appendix A

Documents used for the Evaluation of Hand Gesture Performance (in german)

Pre-Test Questionnaire

Pre-Test Fragebogen

Herzlichen Dank, dass Sie sich bereit erklärt haben an dieser Untersuchung teilzunehmen. Bevor wir anfangen, benötigen wir von Ihnen noch einige Angaben zu Ihrer Person und Ihrer bisherigen Erfahrung mit Computern.
Wir möchten Ihnen hiermit noch einmal versichern, dass alle Daten vertraulich behandelt werden.

Zur Person

Alter: ________

Geschlecht:

☐ männlich
☐ weiblich

Momentante Tätigkeit: ____
(bei Studium auch den Studiengang bitte nennen)

Computer/Internet - Erfahrung

Besitzen Sie momentan einen eigenen Computer?

Ja ☐ Nein ☐

1

Wie viele Stunden verbringen Sie pro Tag an einem Computer?

0 – 1 Stunde ☐
1 – 2 Stunden ☐
2 – 3 Stunden ☐
Mehr als 3 Stunden ☐

Haben Sie schon einmal einen Handschuh für Virtual-Reality Anwendungen ausprobiert?

Ja ☐ Nein ☐

Welche Eingabegeräte benutzen Sie/haben Sie bereits benutzt?

____

Haben Sie bereits einmal mit einem Beamer (Projektor) oder der Powerwall gearbeitet?

Ja ☐ Nein ☐

2

ISO 9241-9 Dependent rating scale (German Version)

Fragebogen zu vergleichender Bewertung

	Kein taktiles Feedback							Taktikles Feedback		
	Am schlechtesten						Am besten	Schlechter	Gleich	Besser
	1	2	3	4	5	6	7	-1	0	1
1 Betätigungskraft										
2 Gleichmäßigkeit bei der Nutzung										
3 Anstrengung bei der Nutzung										
4 Genauigkeit										
5 Benutzungs-geschwindigkeit										
11 Allgemeine Zufriedenheit										
12 Benutzung insgesamt										

	Kein taktiles Feedback							Taktiles Feedback		
	Extrem						Keine	Schlechter	Gleich	Besser
	1	2	3	4	5	6	7	-1	0	1
6. Ermüdung der Finger										
7. Ermüdung des Handgelenks										
8. Ermüdung des Arms										
9. Ermüdung der Schulter										
10. Ermüdung des Nackens										

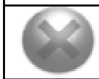
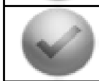
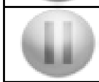
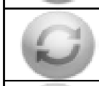
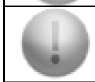
Appendix B

Tasks for the Evaluation of “Nipper”, a multimodal drawing application (in german)

Training Task Set

Bedienung des Programms Nipper

Das Programm „Nipper“ gibt Ihnen die Möglichkeit, Objekte zu zeichnen und zu verändern. Dies können Sie über die Spracherkennung in Kombination mit einem Laserpointer oder auch Handgesten.
Die Spracherkennung wird mit dem Befehl „**Computer**“ aktiviert. Ist die Spracherkennung aktiv (d.h. kein rotes Icon angezeigt) können die Befehle direkt ausgesprochen werden. Es muss nicht mit „Computer“ begonnen werden.
Ist das Gesprochene nicht an den Computer gerichtet, schaltet sich die Spracherkennung nach 10 Sekunden automatisch aus und es wird das rote Icon angezeigt. Der Befehl „Computer“ aktiviert die Spracherkennung wieder.
Das Pausensymbol zeigt an, dass die Spracherkennung die Eingabe verarbeitet.

Icon	Funktion
	Spracherkennung deaktiviert (permanent angezeigt)
	Spracherkennung aktiv (blinkt auf und wird ausgeblendet)
	Spracherkennung beschäftigt
	Keine Verbindung zum Client
	Erkennung nicht Ausreichend (blinkt auf und wird ausgeblendet)

Mit dem Befehl „**Hilfe**“ wird das Hilfemenü aufgerufen. Darin befinden sich die verfügbaren Werkzeuge zur Ansicht, diese können durch Aussprechen ihrer Bezeichnung – z.B. „Rechteck“ ausgewählt werden. Das derzeit aktive Werkzeug ist mit dem Icon ersichtlich z.B. ein T für das Textwerkzeug. Werkzeuge können auch ohne das Hilfemenü direkt über die Sprache ausgewählt werden.
Eigenschaften von Objekten können entweder ausschließlich über die Sprache angegeben werden, z.B. „Hintergrundfarbe rot“ oder aus Menüs ausgewählt werden. So öffnet sich zum Beispiel bei dem Befehl „Hintergrundfarbe“ die Farbauswahl.
Soll etwas aus einem Menü ausgewählt werden ist das ausschließlich mit Sprachbefehlen möglich, Klicks können nicht zur Auswahl verwendet werden.

Übungsaufgaben

Bitte lesen Sie jede Aufgabe laut vor, bevor Sie mit Ihr beginnen.
Aktivieren Sie anschließend die Sprachsteuerung mit dem Befehl „**Computer**".
Nachdem Sie eine Aufgabe bearbeitet haben, deaktivieren Sie bitte die Sprachsteuerung mit dem Befehl „**nicht mehr zuhören**"

1. Rufen Sie das Hilfemenü auf und suchen Sie den Befehl für „Kreis". Zeigen Sie auf das entsprechende Icon.

2. Zeichnen Sie einen Kreis

3. Ändern Sie die Hintergrundfarbe des Kreises

4. Ändern Sie die Linie in eine dicke blaue Linie

5. Zeichnen Sie ein rotes Rechteck mit einer grünen 5 Pixel Linie und einem leichten Schatten

Task Set A

Bitte lesen Sie jede Aufgabe laut vor, bevor Sie mit Ihr beginnen.
Aktivieren Sie anschließend die Sprachsteuerung mit dem Befehl „**Computer**".
Nachdem Sie eine Aufgabe bearbeitet haben, deaktivieren Sie bitte die Sprachsteuerung mit dem Befehl „**nicht mehr zuhören**"

1. Zeichnen Sie einen **kleinen, gelben Kreis,** mit einer **blauen 2 Pixel Linie** auf der **linken Hälfte** der Powerwall.

2. Zeichnen Sie ein **kleines, weisses Rechteck** mit **schwarzer 18 Pixel Linie** und **gerundeten Ecken** mit einem **Radius von 14 Pixel** in der **Mitte** der Powerwall.

3. Zeichnen Sie ein **dunkles Rechteck** mit einer **hellen, dünnen Linie** und einem **leichten Schatten** auf der **linken Hälfte** der Powerwall.

4. Erstellen Sie ein **Textfeld** mit **hellem, halb transparentem Hintergrund** und **dunkler Schrift** mit dem Text **„Hafenbecken"**.

5. **Verschieben** Sie das **dunkle Rechteck** auf die **rechte Seite** der Powerwall und **runden Sie die Ecken stark ab**.

6. **Vergrössern** Sie das **weisse Rechteck** an der **oberen rechten Ecke** und ändern Sie die **Linienbreite auf 15 Pixel**.

7. **Drehen** Sie das **Textfeld nach rechts** und **verändern Sie die Schriftfarbe**.

Task Set B

Bitte lesen Sie jede Aufgabe laut vor, bevor Sie mit Ihr beginnen.
Aktivieren Sie anschließend die Sprachsteuerung mit dem Befehl „**Computer**".
Nachdem Sie eine Aufgabe bearbeitet haben, deaktivieren Sie bitte die Sprachsteuerung mit dem Befehl „**nicht mehr zuhören**"

1. Zeichnen Sie einen **kleinen, roten Kreis**, mit einer **grünen 4 Pixel Linie** auf der **rechten Hälfte** der Powerwall.
2. Zeichnen Sie ein **kleines, blaues Rechteck** mit **weisser 22 Pixel Linie** und **gerundeten Ecken** mit einem **Radius von 10 Pixel** in der **Mitte** der Powerwall.
3. Zeichnen Sie ein **helles Rechteck** mit einer **dunklen, dünnen Linie** und einem **leichten Schatten** auf der **rechten Hälfte** der Powerwall.
4. Erstellen Sie ein **Textfeld** mit **dunklem, halb transparentem Hintergrund** und **heller Schrift** mit dem Text **„Tankschiff"**.
5. **Verschieben** Sie das **helle Rechteck** auf die **linke Seite** der Powerwall und **runden Sie die Ecken stark ab**.
6. **Vergrössern** Sie das **blaue Rechteck** an der **oberen rechten Ecke** und ändern Sie die **Linienbreite auf 15 Pixel**.
7. **Drehen** Sie das **Textfeld nach links** und **verändern Sie die Schriftfarbe**.

Bibliography

[5DT 2007] 5DT. 5DT Data Glove 14 Ultra. 2007. URL: http://www.5dt.com/products/pdataglove14.html (last accessed on Jan. 15, 2009)

[Accot & Zhai 2002] J. Accot and S. Zhai. More than dotting the I's – foundations for crossing-based interfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 73–80, New York, NY, USA, 2002. ACM.

[Adams et al. 2008] N. Adams, M. Witkowski, and R. Spence. The inspection of very large images by eye-gaze control. In *AVI '08: Proceedings of the working conference on Advanced visual interfaces*, pages 111–118, New York, NY, USA, 2008. ACM.

[Akamatsu & MacKenzie 1996] M. Akamatsu and I. S. MacKenzie. Movement characteristics using a mouse with tactile and force feedback. *Int. J. Hum.-Comput. Stud.*, 45(4):483–493, 1996.

[Apple 2009] Apple. iPhone 3G. 2009. URL: http://www.apple.com/de/iphone (last accessed on Jan. 15, 2009)

[ART 2005] A.R.T. Advanced Realtime Tracking GmbH. A.R.T. Fingertracking, Users Manual. August 2005. A.R.T. GmbH

[ART 2006] A.R.T. Advanced Realtime Tracking GmbH. DTrack Technical Appendix. October 2006. A.R.T. GmbH

[ART a] Advanced realtime tracking GmbH. A.R.T. Fingertracking System. URL: http://www.ar-tracking.de/fileadmin/redakteure/Flyer_und_Logos/Flyer_Fingertracking.pdf (last accessed on Jan. 15, 2009)

[ART b] Advanced realtime tracking GmbH. ARTtrack3: Technical Data. URL: http://www.ar-tracking.de/Technical-Data.120.0.html (last accessed on Jan. 15, 2009)

[ART c] Advanced realtime tracking GmbH. Markers: Marker Comparison. URL: http://www.ar-tracking.de/Marker-comparison.145.0.html (last accessed on Jan. 15, 2009)

[ART d] Advanced realtime tracking GmbH. Your experts for infrared optical tracking systems. URL: http://www.ar-tracking.de/ (last accessed on Jan. 15,2009)

[Balakrishnan & MacKenzie 1997] R. Balakrishnan and I. S. MacKenzie. Performance differences in the fingers, wrist, and forearm in computer input control. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 303–310, New York, NY, USA, 1997. ACM.

[Ball & North 2005] R. Ball and C. North. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1196–1199, New York, NY, USA, 2005. ACM.

[Ball et al. 2005] R. Ball, M. Varghese, A. Sabri, E. Cox, C. Fierer, M. Peterson, B. Carstensen, and C. North. Evaluating the benefits of tiled displays for navigating maps. In *IASTED*

International Conference on Human-Computer Interaction (IASTED-HCI 2005), pages 66–71, 2005.

[Ball et al. 2007] R. Ball, C. North, and D. A. Bowman. Move to improve: promoting physical navigation to increase user performance with large displays. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 191–200, New York, NY, USA, 2007. ACM.

[Barco] Barco. Control Rooms, References.
URL: http://www.barco.com/controlrooms/en/references/ (last accessed on Jan. 15, 2009)

[Barco 2008] Barco. Newseum. 2008.
URL: http://www.barco.com/entertainment/en/references/references.asp?ref=3574 (last accessed on Jan. 15, 2009)

[Barrett & Krueger 1994] J. Barrett and H. Krueger. Performance effects of reduced proprioceptive feedback on touch typists and casual users in a typing task. *Behaviour & Information Technology*, 13(6):373–381, 1994.

[Baudel & Beaudouin-Lafon 1993] T. Baudel and M. Beaudouin-Lafon. Charade: remote control of objects using free-hand gestures. *Commun. ACM*, 36(7):28–35, 1993.

[Baudisch et al. 2003] P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In *Proceedings of Interact*, pages 57–64, 2003.

[Bezerianos & Balakrishnan 2005] A. Bezerianos and R. Balakrishnan. The vacuum: facilitating the manipulation of distant objects. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 361–370, New York, NY, USA, 2005. ACM.

[Bieg 2008] J. Bieg. *Laserpointer and Eye Gaze Interaction - Design and Evaluation*. mastersthesis, University of Konstanz, Mar 2008.

[Bolt 1980] R. A. Bolt. “Put-That-There”: Voice and gesture at the graphics interface. In *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262–270, New York, NY, USA, 1980. ACM.

[Brewster et al. 2007] S. Brewster, F. Chohan, and L. Brown. Tactile feedback for mobile interactions. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 159–162, New York, NY, USA, 2007. ACM.

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

[Burdea 2000] G. Burdea. Haptics issues in virtual environments. In *CGI '00: Proceedings of the International Conference on Computer Graphics*, pages 295–302, Washington, DC, USA, 2000. IEEE.

[Buxton 1994] W. Buxton. Human skills in interface design. In L.W. MacDonald & J. Vince (Eds.), *Interacting with virtual environments*, pages 1–12. New York, Wiley, 1994.

[Buxton et al. 2000] W. Buxton, G. Fitzmaurice, R. Balakrishnan, and G. Kurtenbach. Large displays in automotive design. *IEEE Comput. Graph. Appl.*, 20(4):68-75, 2000.

[Cao et al. 2008] X. Cao, M. Massimi, and R. Balakrishnan. Flashlight jigsaw: an exploratory study of an ad-hoc multi-player game on public displays. In *CSCW'08: Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 77–86, New York, NY, USA, 2008. ACM.

[Casiez et al. 2008] G. Casiez, D. Vogel, R. Balakrishnan, and A. Cockburn. The impact of control-display gain on user performance in pointing tasks. *Human Computer Interaction*, 23(3):215–250, 2008.

[Dennerlein et al. 2000] J. T. Dennerlein, D. B. Martin, and C. Hasser. Force-feedback improves performance for steering and combined steering-targeting tasks. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 423–429, New York, NY, USA, 2000. ACM.

[Dexigner 2007] Dexigner. Ford's New Powerwall Capability Accelerates Product Design and Quality. 2007. URL: http://www.dexigner.com/design_news/fords-new-powerwall-capability-accelerates-product-design.html (last accessed on Jan. 15, 2009)

[Dhawale et al. 2006] P. Dhawale, M. Masoodian, and B. Rogers. Bare-hand 3d gesture input to interactive systems. In *CHINZ '06: Proceedings of the 7th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction*, pages 25–32, New York, NY, USA, 2006. ACM.

[Erol et al. 2007] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1-2):52–73, 2007. Elsevier Science Inc.

[eyevis 2008] Eyevis. Control Rooms and Operating Centres. 2008. URL: http://www.eyevis.de/eyevis%20HP%20NEU%20Okt%2007_EN/index.php?action=kontroll (last accessed on Jan. 15, 2009)

[Fäh 2008] S. Fäh. *Sprachsteuerung als Basis multimodaler Interaktion auf grossen, hochaufl¨osenden Displays*. bachelorsthesis, University of Konstanz, Oct 2008.

[Flanagan & Johansson 2002] J. R. Flanagan and R. S. Johansson. Hand movements. In *Encyclopedia of the Human Brain*, volume 2, pages 399–414. Academic Press, 2002.

[Foehrenbach et al. 2008] S. Foehrenbach, W. A. König, J. Gerken, and H. Reiterer. Natural interaction with hand gestures and tactile feedback for large, high-res displays. In *MITH'08: Workshop on Multimodal Interaction Through Haptic Feedback, held in conjunction with AVI'08: International Working Conference on Advanced Visual Interfaces*, May 2008. Workshop Research Paper.

[Forlines & Balakrishnan 2008] C. Forlines and R. Balakrishnan. Evaluating tactile feedback and direct vs. indirect stylus input in pointing and crossing selection tasks. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1563–1572, New York, NY, USA, 2008. ACM.

[Freeman & Weissman 1995] W. T. Freeman and C. D. Weissman. Television control by hand gestures. In *IEEE Intl. Workshop on Automatic Face and Gesture Recognition*. IEEE, 1995.

[Frolov et al. 2002] M.G.P. Frolov and S. Matveyev and S. Klimenko. Using kalman filter for natural hand tremor smoothing during interaction with the projector screen. In *Workshop Proceedings VEonPC2002*, pages 94-101, 2002.

[Furnas & Bederson 1995] G. W. Furnas and B.B. Bederson. Space-scale diagrams: understanding multiscale interfaces. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241, New York, NY, USA, 1995. ACM.

[Goldstein 2002] E. B. Goldstein. *Wahrnehmungspsychologie*. Spektrum Akademischer Verlag, 2te dt. Auflage, 2002.

[Google 2008] Google. Google Earth. 2008. URL: http://earth.google.com/ (last accessed on Jan. 15, 2009)

[Guimbretière et al. 2001] F. Guimbretière, M. Stone, and T. Winograd. Fluid interaction with high-resolution wall-size displays. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 21–30, New York, NY, USA, 2001. ACM.

[Häger-Ross & Schieber 2000] C. Häger-Ross and M. H. Schieber. Quantifying the independence of human finger movements: Comparisons of digits, hands, and movement frequencies. *Journal of Neuroscience*, 20(22):8542–8550, 2000.

[Hardenberg & Bérard 2001] C. von Hardenberg and F. Bérard. Bare-hand human-computer interaction. In *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–8, New York, NY, USA, 2001. ACM.

[Harper et al. 2008] R. Harper, T. Rodden, Y. Rogers, and A. Sellen, editors. *Being Human: Human-Computer Interaction in the year 2020*. Microsoft Research Ltd., 7 J J Thomson Avenue, Cambridge, CB3 0FB, England. 2008.

[Hinckley 2003] K. Hinckley. Input technologies and techniques. In *The Human-Computer Interaction Handbook. Fundamentals, Evolving Technologies and Emerging Applications*, pages 151–168. Lawrence Erlbaum Associates, 2003.

[Hoggan et al. 2008] E. Hoggan, S. A. Brewster, and J. Johnston. Investigating the effectiveness of tactile feedback for mobile touchscreens. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1573–1582, New York, NY, USA, 2008. ACM.

[IBM 1998] IBM Research. DreamSpace: natural interaction. URL: http://www.research.ibm.com/natural/dreamspace/ (last accessed on Jan. 15, 2009)

[ict] ict. ict Steglos FlatScreen MIS-4230 mit Burn-In-Compensation. x. URL: http://www.steglos-flatscreen.de/Technische_Daten_ict_Steglos_FlatScreen_MIS-4230.1110.0.html (last accessed on Jan. 15, 2009)

[Immersion 2008] Immersion. Wireless Data Glove: The CyberGlove® II System. 2008. URL: http://www.immersion.com/3d/products/cyber_glove.php (last accessed on Jan. 15, 2009)

[ISO 9241-9 2000] ISO 9241: Ergonomic requirements for office work with visual display terminals (VDTs) - Part9: Requirements for non-keyboard input devices, 2000.

[ISO 9241-410 2006] ISO 9241: Ergonomics of human-system interaction – Part 410: Design criteria for physical input devices, 2006.

[Jones & Lederman 2006] L. A. Jones and S. J. Lederman. *Human Hand Function*. Oxford University Press, 2006.

[Kandel et al. 2000] E. R. Kandel, J. H. Schwartz, and T. M. Jessel. *Principles of neural science*. McGraw-Hill, fourth edition, 2000.

[Kendon 2004] A. Kendon. *Gesture. Visible Action as Utterance*. Cambridge University Press, 2004.

[Khan et al. 2004] A. Khan, G. Fitzmaurice, D. Almeida, N. Burtnyk, and G. Kurtenbach. A remote control interface for large displays. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 127–136, New York, NY, USA, 2004. ACM.

[Kjeldsen & Kender 1996] R. Kjeldsen and J. Kender. Toward the use of gesture in traditional user interfaces. In *FG '96: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*, page 151, Washington, DC, USA, 1996. IEEE Computer Society.

[König 2006] W. A. König. *Referenzmodell und Machbarkeitsstudie für ein neues Zoomable User Interface Paradigma*. mastersthesis, University of Konstanz, Jun 2006.

[König 2008] W. A. König. *Novel Input Devices for Large, High-Resolution Displays. Desing, Interaction and Evaluation.* Phd Dissertation Proposal, University of Konstanz, 2008. URL: http://hci.uni-konstanz.de/intehrdis/koenig_PhD_Proposal_2008.pdf (last accessed on February, 05 2009)

[König et al. 2007a] W. A. König, H.-J. Bieg, and H. Reiterer. Laserpointer-Interaktion für grosse, hochauflösende Displays. In *Mensch & Computer 2007: Interaktion im Plural. 7te Konferenz für interaktive und kooperative Medien*, pages 69–78. Oldenbourg Verlag, 2007.

[König et al. 2007b] W. A. König, H.-J. Bieg, T. Schmidt, and H. Reiterer. Position-independent interaction for large high-resolution displays. In *IHCI'07: Proceedings of IADIS International Conference on Interfaces and Human Computer Interaction 2007*, pages 117–125. IADIS Press, Jul 2007.

[König et al. 2008] W. A. König, J. Böttger, N. Völzow, and H. Reiterer. Laserpointer-interaction between art and science. In *IUI'08: Proceedings of the 13th international conference on Intelligent User Interfaces*, pages 423 – 424. ACM Press, Jan 2008. Demonstration Session.

[Kurtenbach & Hulteen 1990] G. Kurtenbach and E.A. Hulteen. Gestures in Human-Computer Interaction. In B. Laurel (Ed), *The Art of Human-Computer Interface design*, pages 309–317. Addison-Wesley, 1990.

[Lenman et al. 2002] S. Lenman, L. Bretzner, and B. Thuresson. Using marking menus to develop command sets for computer vision based hand gesture interfaces. In *NordiCHI '02: Proceedings of the second Nordic conference on Human-computer interaction*, pages 239–242, New York, NY, USA, 2002. ACM.

[Letessier & Bérard 2004] J. Letessier and F. Bérard. Visual tracking of bare fingers for interactive surfaces. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 119–122, New York, NY, USA, 2004. ACM.

[Leung et al. 2007] R. Leung, K. MacLean, J., M.B. Bertelsen, and M. Saubhasik. Evaluation of haptically augmented touchscreen gui elements under cognitive load. In *ICMI '07: Proceedings of the 9th international conference on Multimodal Interfaces*, pages 374–381, New York, NY, USA, 2004. ACM.

[Lucente et al. 1998] M. Lucente, G.-J. Zwart, and A. D. George. Visualization space: A testbed for deviceless multimodal user interface. In *Intelligent Environments 98, AAAI Spring Symposium Series*, pages 87–92, 1998.

[MacKenzie & Jusoh 2001] I. S. MacKenzie and S. Jusoh. An evaluation of two input devices for remote pointing. In *EHCI '01: Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction*, pages 235–250, London, UK, 2001. Springer-Verlag.

[MacKenzie & Oniszczak 1998] I. S. MacKenzie and A. Oniszczak. A comparison of three selection techniques for touchpads. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 336–343, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[McGuffin & Balakrishnan 2002] M. McGuffin and R. Balakrishnan. Acquisition of expanding targets. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 57–64, New York, NY, USA, 2002. ACM.

[Microsoft 2008] Microsoft. Microsoft Surface. 2008. URL: http://www.microsoft.com/surface/index.html (last accessed on Jan. 15, 2009)

[Mutschler et al. 2007] E. Mutschler, H.-G. Schaible, and P. Vaupel. *Anatomie Physiologie Pathophysiologie des Menschen*. Wissenschaftliche Verlagsgesellschaft mbH Stuttgart, 2007.

[Ni et al. 2006] T. Ni, S. Schmidt, O. G. Staadt, M. A. Livingston, R. Ball, and R. May. A survey of large high-resolution display technologies, techniques, and applications. In *VR '06: Proceedings of the IEEE conference on Virtual Reality,* pages 223-236, Washington, DC, USA, 2006. IEEE Computer Society.

[Ni et al. 2008] T. Ni, R. McMahan, and D. Bowman. rapMenu: remote menu selection using freehand gestural input. In *Proceedings of IEEE Symposium on 3D User Interfaces 2008,* pages 55–58, Reno, Nevada. 2008

[Nickel & Stiefelhagen 2003] K. Nickel and R. Stiefelhagen. Pointing gesture recognition based on 3d-tracking of face, hands and head orientation. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, pages 140–146, New York, NY, USA, 2003. ACM.

[Nissan 2007] Nissan. Nissan's New Design Center. 2007. URL: http://www.nissan-global.com/EN/DESIGN/REPORT/070308/index.html (last accessed on Jan. 15, 2009)

[Norman 2002] D. A. Norman. *The Design of Everyday Things*. Perseus Books Group, 2002.

[Nullsoft] Nullsoft. Winamp. URL: http://de.winamp.com/ (last accessed on Jan. 15, 2009)

[Oh & Stuerzlinger 2002] Laser pointers as collaborative pointing devices. In *Graphics Interface*, pages 141–149, 2002.

[Parrot & Harrison 1980] J. Parrot and D. Harrison. Surgically dividing pianists' hands [letter to the editor]. *Journal of Hand Surgery*, 5(6):619, 1980.

[Perceptive Pixel 2008] Perceptive Pixel. 2008. URL: http://www.perceptivepixel.com/ (last accessed on Jan. 15, 2009)

[Polhemus 2008] Polhemus. Patriot Wireless. URL: http://www.polhemus.com/polhemus_editor/assets/PATRIOT%20WIRELESS%20brochure.pdf (last accessed on Jan. 15, 2009)

[Poupyrev & Maruyama 2003] I. Poupyrev and S. Maruyama. Tactile interfaces for small touch screens. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 217–220, New York, NY, USA, 2003. ACM.

[Poupyrev et al. 2002] I. Poupyrev, S. Maruyama, and J. Rekimoto. Ambient touch: designing tactile interfaces for handheld devices. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 51–60, New York, NY, USA, 2002. ACM.

[Raskin 2000] J. Raskin. *The humane interface: new directions for designing interactive systems.* ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000.

[RaskinCenter] Raskin Center. Demos. URL: http://rchi.raskincenter.org/index.php?title=Demos (last accessed on Jan. 15, 2009)

[Robertson et al. 2005] G. Robertson, M. Czerwinski, P. Baudisch, B. Meyers, D. Robbins, G. Smith, and D. Tan. The large-display user experience. *IEEE Comput. Graph. Appl.*, 25(4):44–51, 2005.

[Rosenbaum 1991] D. A. Rosenbaum. *Human motor control.* Academic Press, San Diego, 1991.

[Roudaut et al. 2008] A. Roudaut, S. Huot, and E. Lecolinet. TapTap and Magstick: improving one-handed target acquisition on small touch-screens. In *AVI '08: Proceedings of the working conference on Advanced visual interfaces*, pages 146–153, New York, NY, USA, 2008. ACM.

[Rubine 1992] D. H. Rubine. *The automatic recognition of gestures*. phdthesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1992.

[Schapira & Sharma 2001] E. Schapira and R. Sharma. Experimental evaluation of vision and speech based multimodal interfaces. In *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–9, New York, NY, USA, 2001. ACM.

[Scheibe et al. 2007] R. Scheibe, M. Moehring, and B. Froehlich. Tactile feedback at the finger tips for improved direct interaction in immersive environments. *In Proceedings of the IEEE Symposium on 3D User Interfaces*, pages 123–130, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[Segen & Kumar 1998] J. Segen and S. Kumar. Gesture VR: Vision-based 3d hand interace for spatial interaction. In *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia*, pages 455–464, New York, NY, USA, 1998. ACM.

[Segen & Kumar 2000] J. Segen and S. Kumar. Look ma, no mouse! *Commun. ACM*, 43(7):102–109, 2000.

[Serge 1997] N. Serge. On the speed of different senses and nerve transmission by hirsch (1862). *Psychological Research*, 59(4):261–268, 1997.

[Shupp et al. 2006] L. Shupp, R. Ball, B. Yost, J. Booker, and C. North. Evaluation of viewport size and curvature of large, high-resolution displays. In *GI '06: Proceedings of Graphics Interface 2006*, pages 123–130, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.

[Sips et al. 2006] M. Sips, J. Schneidewind, D. A. Keim, and H. Schumann. Scalable pixel-based visual interfaces: Challenges and solutions. In *Information Visualisation, International Conference on*, pages 32–38, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[Soukoreff & MacKenzie 2004] R. W. Soukoreff and I. S. MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *Int. J. Hum.-Comput. Stud.*, 61(6):751–789, 2004.

[Spiegel 2005] Spiegel online. Riesen-monitor, Perfekt sehen mit der Power Wall. 2005. URL: http://www.spiegel.de/netzwelt/tech/0,1518,355609,00.html (last accessed on Jan. 15, 2009)

[Starner et al. 1998] T. Starner, A. Pentland, and J. Weaver. Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12): 1371–1375, Washington, DC, USA, 1998. IEEE Computer Society.

[Sturman 1992] D. J. Sturman. *Whole-Hand Input*. phdthesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1992.

[TechnologyReview 2002] Technology Review. GM's Internet Overhaul. 2002. URL: http://www.technologyreview.com/web/12970/page1/ (last accessed on Jan. 15, 2009)

[Tözeren 2000] A. Tözeren. *Human Body Dynamics, Classical Mechanics and Human Movement*. Springer, New York, NY, USA, 2000.

[Tortora & Derrickson 2006] G. Tortora and B. Derrickson. *Anatomie und Physiologie*. WILEY-VCH Verlag, 2006.

[Tse et al. 2007] E. Tse, M. Hancock, and S. Greenberg. Speech-filtered bubble ray: improving target acquisition on display walls. In *ICMI '07: Proceedings of the 9th international conference on Multimodal interfaces*, pages 307–314, New York, NY, USA, 2007. ACM.

[Tsukada & Yasumura 2002] K. Tsukada and M. Yasumura. Ubi-finger: Gesture input device for mobile use. In *Proceedings of APCHI 2002*, volume 1, pages 388–400, 2002.

[Vicon 2009] Vicon. Performance Pack. 2009. URL: http://www.vicon.com/products/performancepack.html (last accessed on Jan. 15, 2009)

[Vicon 2009 b] Vicon. Motion Capture Systems from Vicon. 2009. URL: www.vicon.com (last accessed on Jan. 15, 2009)

[Vogel & Balakrishnan 2004] D. Vogel and R. Balakrishnan. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 137-146, 2004. ACM.

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

[Ware 2004] C. Ware. *Information visualization: perception for design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2004.

[Wilson 2000] A. D. Wilson. Robust computer vision-based detection of pinching for one and two-handed gesture input. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 255–258, New York, NY, USA, 2006. ACM.

[Wilson & Bobick 2000] A.D. Wilson, and A.F. Bobick. Realtime Online Adaptive Gesture Recognition. *ICPR '00: Proceedings of the International Conference on Pattern Recognition*, pages 1270, Washington, DC, USA, 2000. IEEE Computer Society.

[Wilson 2006] A. D. Wilson. Robust computer vision-based detection of pinching for one and two-handed gesture input. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 255–258, New York, NY, USA, 2006. ACM.

[Wilson 2007] A. D. Wilson. Sensor- and Recognition-Based Input for Interaction. In A. Sears & J.A. Jacko (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, pages 177–200. Lawrence Erlbaum Associates, 2007.

[Wired 2008] Wired. Super Tuesday's Gadgets: Touch Screen Monitors, New Notebooks, and 3D Graphics Drive TV Coverage. 2008.
URL: http://blog.wired.com/gadgets/2008/02/monitors-notebo.html (last accessed on Jan. 15, 2009)

[Worldwind 2006] Nasa. World Wind. URL: http://worldwind.arc.nasa.gov/ (last accessed on Feb, 08 2009)

[Wu & Balakrishnan 2003] M.Wu and R. Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multiuser tabletop displays. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 193–202, New York, NY, USA, 2003. ACM.

[XIST 2006] XIST DataGlove. DataGlove Beschreibung.
URL: http://www.nodna.com/x-ist/index.php?cat=c8_DataGlove-Beschreibung.html (last accessed on Jan, 15 2009)

[Yost et al. 2007] B. Yost, Y. Haciahmetoglu, and C. North. Beyond visual acuity: the perceptual scalability of information visualizations for large displays. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 101–110, New York, NY, USA, 2007. ACM.

[ZKM 2008] ZKM. PanoramaFestival. 2008. URL: http://zkm.de/panoramafestival/index.php (last accessed on Jan. 15, 2009)

[Zobl et al. 2003] M. Zobl, M. Geiger, B. Schuller, M. Lang, and G. Rigoll. A real-time system for hand gesture controlled operation of in-car devices. In *ICME '03: Proceedings of the*

2003 International Conference on Multimedia and Expo - Volume 3 (ICME '03), pages 541–544, Washington, DC, USA, 2003. IEEE Computer Society.

List of Figures

List of Tables