INTERACTING WITH STROKE-BASED NON-PHOTOREALISTIC RENDERING ON LARGE DISPLAYS

JENS GRUBERT

Studienarbeit

Institute of Simulation and Graphics
Faculty of Computer Science
Otto-von-Guericke-University Magdeburg

Interactions Laboratory
Faculty of Computer Science
University of Calgary

February 2008

Supervisors:
Dr. Tobias Isenberg
Prof. Dr. Sheelagh Carpendale
Dipl.-Ing. Angela Brennecke

Jens Grubert: *Interacting with Stroke-Based Non-Photorealistic Rendering on Large Displays,* Studienarbeit, © February 2008

ABSTRACT

Since over two decades researchers work on approaches for generating non-photorealistic renditions, that are often inspired by artistic styles. Research in this field has focused on developing algorithms to automatically produce a final rendition. Recently more effort was placed in including the user into the *process* of creating a non-photorealistic rendition.

This thesis takes a step toward opening the process of interactively creating non-photorealistic paintings to a broad range of users, who do not need to have backgrounds in arts or computer science. Therefore, techniques have been designed that focus on an intuitive interaction with non-photorealistic paintings. The interaction techniques presented in this thesis have been specifically designed for the use on large, touch-sensitive displays. These displays can be set up to resemble the setup that artists use in physical painting environments and, therefore, can support an expressive way of image creation.

The interaction techniques based on hand postures that are presented in this thesis allow users to interact expressively and immediately with non-photorealistic renditions.

Two bimanual interaction techniques combine the advantages of interacting at a distance and up close with large displays. By providing a combination of both techniques the interface supports the seamless transition between indirect and direct interaction. All techniques allow users to create a wide variety of effects by working at different levels of precision and expressiveness.

To expand the expressive palette of tools that users can employ throughout the image creation process even further, ideas from digital painting systems were incorporated. User-sketched strokes combine advantages of these digital painting systems with strengths of stroke-based rendering. Users are enabled to sketch a wide variety of 2D shapes that can be used as rendering primitives. These primitives allow to construct non-photorealistic paintings, without the need to create a painting pixel-by-pixel.

ZUSAMMENFASSUNG

Seit über zwanzig Jahren entwickeln Forscher auf dem Feld des nicht-photorealistischen Renderings Methoden zur Generierung von Bildern und Animationen, die von verschiedenen Kunststilen beeinflusst sind. Die Forschung auf diesem Gebiet hat sich darauf konzentriert, Algorithmen zur Erstellung fertiger Bilder zu entwickeln. Seit kurzem haben Forscher den Wert, Nutzer in den *Erstellungsprozess* eines nicht-photorealistischen Bildes einzubeziehen, erkannt.

Diese Arbeit versucht diesen interaktiven Erstellungsprozess zur Generierung nicht-photorealistischer Bilder einem breiten Nutzerkreis zu öffnen, der weder über Kenntnisse der Informatik noch über einen künstlerischen Hintergrund verfügen muss. Dafür wurden Techniken entwickelt, die eine intuitive Interaktion mit nicht-photorealistischen Bildern ermöglichen sollen. Die in dieser Arbeit vorgestellten Interaktionstechniken wurden für den Einsatz an großformatigen, berührungssensitiv Bildschirmen entwickelt. Diese großformatigen Bildschirme ähneln realen Leinwänden und können daher eine expressive Art der Interaktion unterstützen.

Die in dieser Arbeit beschriebenen Interaktionstechniken, die auf Handposen basieren, sollen den Nutzern eine expressive und unmittelbare Interaktion mit nicht-photorealistischen Bildern ermöglichen.

Zwei beidhändige Interaktionstechniken kombinieren die Vorteile der Interaktion auf Distanz und an großen Bildschirmen und sollen das nahtlose Wechseln von indirekter zu direkter Interaktion ermöglichen. Die Techniken erlauben den Nutzern, durch bewusstes Arbeiten auf verschiedenen Detailgraden, eine Vielzahl künstlerischer Effekte zu erzielen.

Weiterhin werden Ideen aus dem Bereich der digitalen Zeichen- und Malsystemen verwendet, um den Nutzern eine breite Palette an Werkzeugen für die Erstellung von nicht-photorealistischen Bildern anzubieten. Vom Nutzer frei skizzierbare Liniengraphiken kombinieren Vorteile von digitalen Zeichen- und Malsystemem mit den Stärken des *stroke-based renderings*. Die frei skizzierbaren Liniengraphiken erlauben Nutzern vielfältige 2D Strukturen zu zeichnen. Diese Strukturen können daraufhin als Rendering-Primitive verwendet werden. Der Einsatz dieser Primitive erlaubt Nutzern ein nicht-photorealistisches Bild schnell zu erstellen, ohne es Pixel um Pixel malen zu müssen.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

Grubert, J., Carpendale, S., and Isenberg, T. December 2007. Interactive Stroke-Based NPR using Hand Postures on Large Displays. Tech. Rep. 2007-883-35, Department of Computer Science, University of Calgary, Canada.

Grubert, J., Hanckock, M., Carpendale, S., Tse, E., and Isenberg, T. December 2007. Interacting with Stroke-Based Rendering on a Wall Display. Tech. Rep. 2007-882-34, Department of Computer Science, University of Calgary, Canada.

Grubert, J., Carpendale, S., and Isenberg, T. 2008. Interactive Stroke-Based NPR using Hand Postures on Large Displays. In Short Papers at Eurographics 2008. Aire-la-Ville, Switzerland. In Press. of Calgary, Canada.

Isenberg, T., Everts, M., Grubert, J., and Carpendale, S. 2008. Interactive Exploratory Visualization of Vector Fields. Computer Graphics Forum Vol. 27. Submitted.

ACKNOWLEDGMENTS

During the course of this project I met many people, who gave me moral and technical support. All people in the Interactions Laboratory at the University of Calgary made my stay enjoyable and fascinating. I especially would like to thank the following people:

Tobias Isenberg, whose supervision, inspiration and constant feedback encouraged me to stay motivated throughout this project.

Sheelagh Carpendale invited me to Calgary and the Interactions Laboratory. Sheelagh also provided me with good feedback in crucial situations.

Mark Hancock gave me precise feedback, participated in interesting discussions, and helped me with statistics. He also put much effort into one paper.

Edward Tse joined in for good discussions and gave me insights into multimodal interaction techniques.

Min Xin, Cody Watts, Uta Hinrichs, Petra Neumann, Robin Arsenault, Mark Watson, Helen He, and Dane Bertram participated in informal experiments and pilot studies. In addition, I would like to thank all participants of the formal experiments for their time and invaluable feedback.

Cheng Guo participated in informal experiments but also gave me insights into Wii Remote gestures.

Fabricio Anastacio did a lot of pilot testing on interaction techniques and participated in fruitful discussions.

Martin Schwarz built the initial Interactive Canvas system and gave me supporting comments when I was investigating the code of the initial system.

André Miede provided the Classic Thesis LaTeXtemplate for this thesis.

CONTENTS

LIST OF TABLES

ACRONYMS

ANOVA Analysis of Variance

API Application Programming Interface

CAVE Cave Automatic Virtual Environment

DViT Digital Vision Touch

DOF degrees of freedom

dpi Dots per inch

fps Frames per second

HCI human-computer interaction

IR LEDs infrared light emitting diodes

MERL Mitsubishi Electric Research Laboratory

NPR non-photorealistic rendering

OpenGL Open Graphics Library

SBR stroke-based rendering

SDK Software Development Kit

TSLD touch-sensitive large display

Wiimote Nintendo Wii Remote

WIMP Windows, Icons, Menus, and Pointers

WYSIWYG What You See Is What You Get

1 INTRODUCTION

Non-photorealistic rendering denotes the area of scientific endeavor dealing with the computer generation of images and animations that seem to be made “by hand” [Strothotte and Schlechtweg, 2002]. Many non-photorealistic rendering (NPR) approaches have been inspired by art and resemble different artistic styles, while others have created novel styles. However, most approaches concentrated primarily on the final outcome of a image creation process, a non-photorealistic rendition, rather than on the process itself.

1.1 MOTIVATION

Recently, research indicated that it is useful to integrate users's input into the process of creating a non-photorealistic rendition (e. g., [Deussen et al., 2000; Kalnins et al., 2002; Schwarz et al., 2007]), to better match users's expectations about the outcome of this process and to allow them to actively steer this process. Instead of dealing with the development of new styles for the use in NPR, this thesis concentrates on how *process* of image creation can be opened to a broad range of users.

This thesis is based on the thought that intuitive interaction techniques make a strength of stroke-based NPR approaches accessible to a broad range of users, namely the ability to resemble artistic styles without the need to create an image manually pixel-by-pixel. Users, who neither have backgrounds in arts nor in computer science should be enabled to intuitively create artistic non-photorealistic renditions. The goal of this thesis is, therefore, to provide intuitive interaction techniques that invite a broad range of users to experiment with explorative and expressive creation of non-photorealistic paintings.

In order to accomplish this goal, the techniques presented in this thesis specifically target the use of touch-sensitive large displays (TSLDs). These displays not only become more and more affordable but lend themselves well for the purpose of creating non-photorealistic paintings. Their increasing size and resolution provides much space for displaying large paintings with fine structures. Vertical displays even resemble the setup often found in artists's studios who work with canvases on scaffolds. Interaction techniques that are based on 2D mouse input and that are designed for todays desktop systems are often not suitable for interaction at TSLDs. TSLDs allow immediate interaction by directly touching the display. However, this form of direct interaction, challenges interaction designers, as adoption of interaction techniques, that where developed specifically for desktop systems, such as Windows, Icons, Menus, and Pointers (WIMP) based interaction, proves to be difficult in many situations [Czerwinski et al., 2006].

1.2 CONTRIBUTIONS

The main goal of this work is to allow a broad range of users to intuitively create stroke-based non-photorealistic renditions.

To accomplish this goal techniques in three domains are investigated. Interaction techniques for the use on TSLDs, that target to allow a broad range of users to expressively create non-photorealistic paintings, have been developed during this project. The interaction techniques are meant to be used straightforward, without the need to read lengthy instructions manuals, and only require commercially available products such as SMART Technologies Smartboards®, and the Wiimote. They allow users to interact up close on a large display and at a distance. Users can choose the interaction that seems appropriate and can change it at will. A minimalistic user interface leaves the majority of the screen space for the actual painting and reduces the need to first learn complex interaction techniques.

Stroke-based NPR techniques allow users to quickly abstract from an input image and do not require them to paint a whole rendition pixel-by-pixel.

The ability to create a wide variety of 2D shapes as in digital painting systems is used to allow users to define their very own shapes that are used as NPR primitives.

1.3 LIMITATIONS

The interaction techniques presented in this thesis are designed to provide direct access to frequently used functions, that are needed to create non-photorealistic renditions. They should be used by broad range of users and invite them for an expressive image creation.

However, the techniques do not allow access to a large number of functions. The technique based on hand postures provides access to four functions in a given context, the bimanual techniques to thirteen functions.

The bimanual interaction techniques allow indirect and direct interaction and have been developed to combine the advantages of interaction at a distance and up close with TSLDs. While the remote pointing technique for interaction at a distance provides the convenience to sit and to have an overview over the whole display, it does not invite users for expressive interaction to the extend as the direct-touch techniques.

It, therefore, would be desirable to integrate the concepts of expressive interaction and seamless transition between interaction at a distance and up close into one approach.

The techniques are used in contexts that do not require high precision during interaction. If high precision, e. g., for selecting small objects, is required other interaction techniques may be more suitable.

Furthermore, the interaction techniques have been specifically designed to leverage the advantages of TSLDs. The use of the direct-touch techniques is limited on regular desktop systems.

1.4 ORGANIZATION

The thesis is structured as follows:

CHAPTER 2 gives an overview of related work and helps to position this thesis among research domains. It introduces concepts and notions that help to understand the subsequent chapters. A special focus is placed on interaction on large displays.

CHAPTER 3 discusses concepts applied throughout this thesis. The underlying design process is presented, with a special focus on an informal observational study, that laid foundations for the techniques described in Chapter 4. Afterward, it introduces the notions of *expressive interaction* that guided the design of a interaction technique based on hand postures and the development of user-sketched strokes. The concept of *seamless transition* between interaction at a distance and up close motivates the development of bimanual interaction techniques for TSLDs. Finally, it is described how various *degrees of precision* invite users for a playful exploration of stroke-based NPR on TSLDs.

CHAPTER 4 describes how a interaction technique based on hand postures and bimanual interaction techniques for direct and indirect interaction with TSLDs have been realized. Afterward, it is presented how they can be used to interact with non-photorealistic renditions. The development of user-sketched strokes and its underlying data structure are described at the end of the chapter.

CHAPTER 5 presents a user study about the bimanual interaction techniques. The techniques are evaluated in terms of speed and accuracy. Afterward, observations on how people used these techniques to create an entire painting are described.

CHAPTER 6 summarizes the work presented in this thesis and completes with possible directions for future work.

2 BACKGROUND

In order to put this thesis into the context of the research domains human-computer interaction (HCI), NPR, and digital painting systems, this chapter presents an overview of related work. It introduces terms needed for understanding the following chapters.

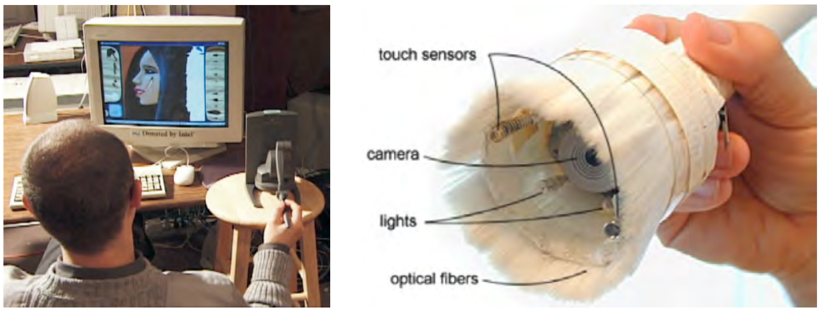
2.1 INTERACTION WITH DIGITAL PAINTING SYSTEMS

Systems for creating digital paintings from scratch exists in a wide variety ranging from simple applications like Microsoft Paint® to complex programs like Adobe Photoshop® that have a high learning curve and are generally not straightforward to use. Though most digital painting systems were developed for desktop use with mouse, keyboard, or stylus input (for an overview see e. g., [Smith, 2001]), considerable work has also been carried out for providing alternative user input. These alternative user interfaces were a source of inspiration during the design of interaction techniques that are described in this thesis.

Baxter et al. [2001] introduced the DAB system, a natural interface that resembles traditional artists's tools rather than providing a complex WIMP user interface for creating digital paintings. A physically based deformable 3D brush is steered by a SensAble Dekstop Phantom (see Figure 1(a)). Their system provides haptic feedback and a bi-directional, two layer paint model. This allows users to produce paint strokes intuitively. This system avoids a complex user interface and is built for desktop systems.

Lang et al. [2003] uses a physical paint brush as interaction device. It is equipped with 6 degrees of freedom (DOF) trackers and is leveraged to paint directly on a tabletop display. A separate color mixing area allows for selecting primary colors and mixing them. Afterward the virtual paint is distributed across the digital canvas. This way an immersive painting experience is created that allows for skill transfer of users with real world painting experience and encourages expressive paintings.

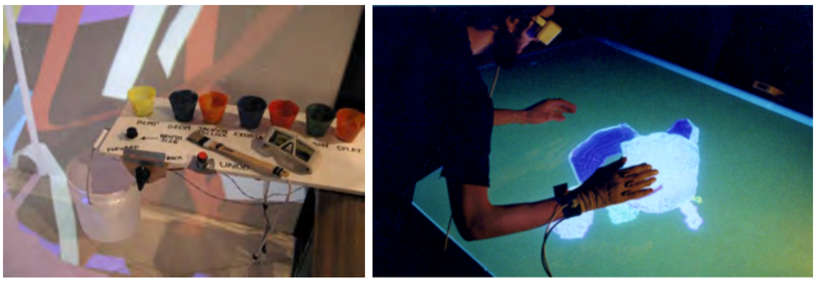
I/O BRUSH is another example for building an intuitive user interface for a digital painting application [Ryokai et al., 2004]. The system allows users to easily create paintings with textures captured from real world objects. A physical brush (shown in Figure 1(b)) is equipped with a video camera, light, and touch sensors. Used outside the digital canvas the brush can pick up textures with its integrated camera. On the canvas these real world textures are used to create abstract objects. A preliminary study with young children (age 4-5) has shown that the children explored their physical surroundings with the brush and made connections between real world surfaces and their abstract appearance on the digital canvas. Even though I/O BRUSH was developed on small displays it is suitable for use on large displays.



(a) (b)

Figure 1: Parts of the DAB [Baxter et al., 2001] and I/O BRUSH [Ryokai et al., 2004] user interfaces: a SensAble Phantom with haptic feedback for desktop use (a) and a physical brush equipped with camera, light, and touch sensors for capturing real world textures (b).

In CavePainting [Keefe et al., 2001] artists can work in a 2.4 ×2.4m ×2.4m Cave Automatic Virtual Environment (CAVE) to create 3D paintings. Physical props like a brush, a paint bucket, and color pickers (shown in Figure 2(a)) are tracked in 3D space and used to create and modify 3D shapes. This way an immersive and expressive painting experience can be achieved. Unfortunately this system is bound to the rather expensive CAVE systems and puts the burden to wear data gloves and 3D goggles on the artist.



(a) (b)

Figure 2: The table interface used in the 3D CavePainting system [Keefe et al., 2001] (a) and a user interacting with the Responsive Workbench [Cutler et al., 1997] in the Surface Drawing [Schkolne et al., 2001] system (b).

A similar approach is used in the Surface Drawing environment, in which surfaces are created by moving a glove-equipped hand through 3D space [Schkolne et al., 2001]. A 3D view is provided through the semi-immersive 3D display Responsive Workbench [Cutler et al., 1997] and thus is an actual large display system (see Figure 2(b)). The system supports free exploration of 3D space and uses the so-called Cookie

Cutter algorithm for creating, joining and erasing of surface meshes (see [Schkolne et al., 2001] for details). Even though the main focus is on 3D surface modeling this approach could be an interesting basis for a digital painting system.

Recently Lee et al. [2006] introduced Interactive 3D Fluid Jet Painting to create digital paintings in a Jackson Pollock Style. A 3D viscous fluid jets model is used to generate streams of virtual paint that are distributed across a canvas. The user interacts with a DiamondTouch table [Dietz and Leigh, 2001] to indicate the position of a virtual paint bucket out of which the paint jets are poured out (see Figure 3). As the available screen space and resolution on the DiamondTouch table is limited (resolution: 1024 × 768 pixels, width: ca. 81 cm, depth: ca. 51 cm) the final painting is projected on a wall mounted large display for an immersive experience.

Figure 3: In 3D fluid Jet Painting [Lee et al., 2006] the user interacts with a virtual paint bucket on a Diamond-Touch table. The resulting image is projected on a wall mounted display.

In contrast to digital painting systems, with which user create paintings from scratch, many NPR approaches take an input model, such as a 2D image, to abstract from it using NPR primitives, such as strokes. NPR approaches that are relevant for this thesis are described next.

2.2 NON-PHOTOREALISTIC RENDERING

In the field of NPR much research was carried out that focuses on resembling different artistic styles, such as stippling, hatching, watercolor paintings [Gooch and Gooch, 2001; Strothotte and Schlechtweg, 2002]. The areas that are closely related to this thesis are stroke-based rendering (SBR) and interaction with NPR.

2.2.1 *Stroke-Based Rendering*

SBR encompasses approaches for the automatic creation of non-photorealistic renditions by placing strokes, discrete elements that are usually bigger than a pixel, on 2D and 3D models. In contrast to digital painting systems often these underlying models, such as images or 3D scenes, are used to guide the artistic process. Strokes being simulated are, e. g., pencil strokes, pen-and-ink strokes used for stippling, brush strokes for watercolor or oil painting, or mosaic tiles. A stroke is represented in a data structure that can be rendered in the image plane. Hertzmann gives an comprehensive overview of SBR in [Hertzmann, 2003].

Given an input model SBR approaches usually place strokes automatically on that model according to certain goals, such as the resemblance of the input model with a specific visual style or the abstraction from that model. Typically, there is a large number of parameters available to steer the degree of approximation and abstraction, for example primitive type, size, orientation, position. Many parameters are set before the generation of the final image.

As strokes often are coarser structures than the ones that are present in the input model a SBR approximates the input model. While the automatic placement according to model features (such as gradients in 2D images or silhouette edges on 3D models) allows for fast application of visual styles it takes away much of the artistic freedom that is involved when applying these styles by hand.

The large number of parameters to influence the visual style can result in lengthy adjustment-rendering loops as the actual rendering process behaves like a black box for the user [Schwarz et al., 2007]. Furthermore many approaches provide only a limited set of stroke shapes that can not be extended by users.

An early but influential approach for the field of SBR was Paint By Numbers by Haeberli [1990]. Strokes are painted with the mouse on an input image and their color is determined by the color of that underlying image. Other parameters, such as position, shape, style, orientation and size are controlled through user interaction with the mouse and keyboard. The system allows the application of various artistic styles, e. g., pointillism (shown in Figure 4), painterly rendering or decorative mosaics.

Hertzmann [1998] introduced a multi layer approach into SBR. Strokes with different parametrization are rendered into layers as B-Splines. Different style parameters are used to match different levels of detail in the source image and to generate different styles, namely "Impressionist" (shown in Figure 5(a)), "Expressionist", "Colorist Wash", and "Pointillist".

The previous approach was extended in Paint By Relaxation [Hertzmann, 2001]. A relaxation algorithm combined with various search heuristics are employed to automatically place brush strokes. In this approach users specify what style they want through parameterization not how to apply it. Because the parameterization of the algorithms

Figure 4: Pointillism style applied to a flower image in Haeberli's Paint By Numbers [Haeberli, 1990].

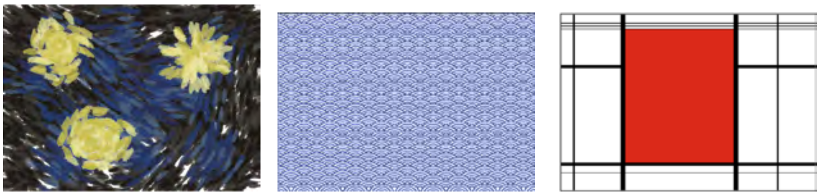
are not obvious Hertzmann still allows users to place single strokes by hand again.

Representing individual rendering primitives as agents in a multi-agent framework that create imagery through user input is explored by Mason et al. [Mason and Carpendale, 2001a,b; Mason et al., 2005; Mason, 2006]. Goals were to lay back the control over the image creation process into the hands of the artist and to extend the set of tools that can be used for expressive image creation. A rendering primitive exists as agent in an agent-space and renders itself into a canvas-space. Agents with the same properties are grouped in tribes, coalitions are formed when agents discover that the fulfillment of their goals are compatible with the goals of other agents. With her multi-agent framework Mason created images with styles such as the Japanese Seigaiha style, painterly rendering or Piet Mondrian's style (see Figure 6). Unfortunately interacting with the rendering primitives requires the user to learn an element-based language.

Schlechtweg et al. [2005] presented a different multi-agent system for rendering 2D images in artistic styles such as hatching, stippling, or mosaics (see Figure 5(b)). Each stroke is represented as a RenderBot. A RenderBot is a reactive, behavior-based, autonomous agent, that renders a non-photorealistic image in cooperation with other agents. Besides a source image additional G-Buffers [Saito and Takahashi, 1990] can be used to influence the behavior of the RenderBots. Several artistic styles can be achieved by using various RenderBot classes that differ in their physical behavior and their way of painting. This behavior can

(a) (b)

Figure 5: Multiple layers of strokes are used to create a non-photorealistic rendition (a) in [Hertzmann, 1998]. With the RenderBots system [Schlechtweg et al., 2005] different visual styles (b) such as painterly rendering (top) and mosaics (bottom) can be created.



(a) Painterly rendering style. (b) Seigaiha Style (c) Piet Mondrian's style.

Figure 6: Three different artistic styles that were generated inside Mason's multi-agent framework [Mason, 2006]: Painterly rendering stlye (a), the Japanese Seigaiha style (b), and Piet Mondrian's style (c).

be adjusted through setting several parameters at runtime in a WIMP based interface. As the user interface presents a large number of parameters at once it can be difficult to adjust in a way that a desired visual style is achieved. NPR approaches that lie a focus on user interaction during are presented next.

2.2.2 *Interaction with Non-Photorealistic Rendering*

While most work in NPR concentrated on image creation at real-time frame rates using predefined sets of parameters, over time researchers realized the importance of user interaction in the creation phase of a non-photorealistic rendition to produce expressive results. The main approaches that are relevant to this thesis are the ones described in [Kalnins et al., 2002; Kalnins, 2004] and [Schwarz et al., 2007; Schwarz, 2007].

In [Kalnins et al., 2002; Kalnins, 2004] an approach was introduced, that allows users to interactively paint strokes on 3D models. The What You See Is What You Get (WYSIWYG) system allows users to select brush and paper styles and to draw strokes over a 3D model from different viewpoints in realtime (an example image is shown in Figure 7(a)). As the viewpoint changes the number and placement of strokes are adapted by the system. The WYSIWYG system also allows to automatically extract feature strokes out of 3D models and to re-stylize them. As the creation and stylization of strokes happens in real time Kalnins's approach helps the user to achieve desired styles more direct and faster than with previous approaches.

Schwarz et al. [Schwarz et al., 2007; Schwarz, 2007] tried to further expand interaction possibilities and to offer novel and responsive painting techniques for NPR. Like Mason, Schwarz et al. focused on the interactive use of rendering primitives. Their Interactive Canvas provides algorithmic support during image creation in an interactive process. The system avoids the global parameter tweaking in adjust-rendering cycles as in many previous approaches. Instead it leverages the power of local tools that provide spatially explicit computational assistance to interactively manipulate a large number of primitives at once. Being a hybrid approach the Interactive Canvas combines the advantages of visually rich but inflexible pixel graphics with visually less rich but more flexible vector graphics (an example image is shown in 7b). All tools are accessible through a virtual painting palette (shown in 9b) that avoids the need to enter numerical values. As an underlying structure the University of Calgary Tabletop Framework, that is described in [Isenberg et al., 2006; Miede, 2006], is used to retain responsible interaction with a large number of primitives, even on high-resolution displays. This way the approach is suitable for interaction on desktop systems and on a TSLD. On a TSLD bimanual interaction is supported but demands more concentration from the user while interacting. Similar to previous approaches only a fixed, predefined set of rendering primitives can be created and modified. This, consequently, limits the creative freedom of artists who want to create their very own marks. In the Interactive Canvas system the palette provides access to most tools, that are used for creating and modifying rendering primitives, without the need to enter numerical values. However, the system essentially provides a WIMP interface, as many tools are accessed through pie menus and sliders. It could benefit from alternative interaction techniques for large displays that can help to realize an immersive painting experience for users. One goal of this thesis is to provide alternative interaction techniques for the Interactive Canvas paradigm that allow users to create expressive paintings without the need to think about the interface.

2.3 INTERACTION WITH LARGE DISPLAYS

Large, high-resolution displays are often used for collaborative work in small teams or as information presentation systems in public spaces (e. g., [Elrod et al., 1992; Johanson et al., 2002]). Despite a number of usability issues, that were recently summarized by Czerwinski et al. [2006], their large size makes them a good choice if information has to be shared between participants that are co-located [Agrawala et al., 1997]. There are several advantages that make them interesting for in-

(a) (b)

Figure 7: An artist drew silhouettes, creases, and several forms of hatching to create a non-photorealistic rendition from an underlying 3D scene (a) in [Kalnins, 2004]. Bold strokes were used to abtract from a photograph of cherries (b) in the Interactive Canvas system [Schwarz et al., 2007].

teraction in a digital painting setting. Their large-size combined with high resolutions allows for precise depiction of fine structures, such as thin strokes. Furthermore, interaction at a distance with remote pointing devices or up close with multiple fingers, hand gestures and hand postures interaction, is possible. This makes such large displays interesting to explore the space of alternative user interfaces for creating non-photorealistic renditions.

2.3.1 *Remote Pointing*

Ray casting is a commonly used technique for pointing to distant objects on large displays. Remote pointing devices include e. g., laser pointer [Dan R. Olsen and Nielsen, 2001; Peck, 2001; Myers et al., 2002; Parker et al., 2005], data gloves [Tse et al., 2007a], wand tracker [Ciger et al., 2003], and hand tracker [Vogel and Balakrishnan, 2005].

Dan R. Olsen and Nielsen [2001] described the use of a laser pointer in group settings for indirect interaction on large vertical displays. A series of interactive techniques using a laser pointer and a camera for point recognition were developed for list navigation, selection, and text entry. Tests suggested that interaction based on 2D mouse input is clearly faster than the techniques based on laser pointers. Along with the unfamiliarity with laser pointer based interaction, the performance of the laser pointer technique was, nonetheless, surprisingly good relative to other interactive techniques.

In the same year Peck [2001] conducted an user study with ten participants to derive useful parameters for laser pointer interaction, such as the the dwell time of a laser point, target acquisition time, and wiggle size, that occurs due to hand jitter. Coon found that the physical design of the laser pointer impacts the performance and that widgets designed for laser pointer interaction must be fairly big if standing 3m or farther away from a wall display.

Myers et al. [2002] compared the use of different laser pointer form factors (conventional, pen, pda-mount, toy gun) that are shown in Figure 8 to mouse and direct-touch interaction on a Wall mounted Smartboard. Direct-touch was found to be the fastest and most accurate technique, followed by the mouse and, finally, laser pointers. The authors's studies indicate why conventional interaction techniques designed for a mouse are doomed to fail when used with a laser pointer. The beam is too unsteady due to natural hand jitter, users cannot turn the beam on or off where they want, and there is no mouse button. However, the authors found that it is convenient not to walk up to touch a display but to point to things across the room using a laser pointer.

Figure 8: Laser pointer devices used in a comparison between remote pointing, direct-touch and 2D mouse input [Myers et al., 2002] (from left to right): conventional laser poitner, pen-like laser pointer, pda-mounted, and toy gun mounted laser pointer.

3D input devices were also investigated as solutions to distant pointing [MacKenzie and Jusoh, 2001; Zhai, 1998]. Hand-held isometric input devices and isotonic flying mice were used for remote pointing. An isometric input device does not require the movement of the device itself and, therefore, results in less fatigue than a freely held isotonic device. Depending on the device design the transition between distant pointing and direct-touch interaction may be complicated when interacting with TSLDs.

Alternative input techniques for remote pointing include body, hand and eye tracking. Vogel and Balakrishnan [2004] introduced an approach where coarse grained 1D pointing is accomplished through body position. This way a context for fine-grained actions is provided, but 2D control is difficult with this approach. Nickel and Stiefelhagen [2003] used methods from computer vision to determine pointing direction using the head-to-hand line of sight and the orientation head and forearm. Head orientation was found to be important for determining pointing direction. Eye tracking for controlling a cursor on large displays was investigated as early as 1981 by Bolt [1981] or more recently by Skaburskis et al. [2004]. Saccade (fast eye) movements and difficulties in finding a suitable click mechanism make it difficult to use eye gaze alone for precise pointing and selection. However, when combined with a regular pointing device, eye tracking can be used for coarse contextual pointing as in "Manual and gaze input cascaded (MAGIC) pointing" by Zhai et al. [1999].

2.3.2 *Direct-Touch*

Touching a screen with the finger or a pen is an effective way to interact when the user is up close to a display. Many techniques (e. g., [Rekimoto, 1997; Vernier et al., 2002; Wu and Balakrishnan, 2003; Kruger et al., 2005]) were designed for multi user, collaborative environments and recently evaluated by Nacenta et al. [2005, 2007].

Besides using a single finger or a pen, hand gestures can be used to widen the interaction possibilities. Wu and Balakrishnan [2003] presented a variety of multi-finger and whole hand gestural interaction techniques for tabletop displays that leverage and extend types of actions, that people perform on physical tabletops. Wu et al. use the DiamondTouch table from Mitsubishi Electric Research Laboratory (MERL) as primary hardware platform [Dietz and Leigh, 2001]. The gesture set included tap, double tap, quickly sliding the finger away or toward oneself (flick and catch), vertical hand, horizontal hand, tilted hand, two vertical and two corner shaped hands. A furniture layout application was used to demonstrate the power of those gestures. Initial user feedback suggested that learning these techniques does not require much time.

Tse et al. widely explored the use of hand gestures combined with speech input in single and multi user tabletop environments (e. g., [Tse et al., 2006a,b,c, 2007b; Tse, 2007; Tse et al., 2008]). Similar to previous research (e. g., [Bolt, 1980; Cohen, 2000]) Tse et al. leveraged the suitability of speech for issuing commands (e. g., "move", "built", "fly to") and gestures for indicating actions, such as pointing to a location on the display. The authors developed multimodal interaction techniques using speech commands and gestures that create awareness and are meaningful to other users in a co-located multi user setting.

In contrast to hand gestures that are time dependent hand movements, hand postures are static hand signs. Vision-based hand posture detection has received considerable attention in recent years. As many techniques were specifically designed for 3D tracking, only few approaches are applicable for direct-touch interfaces. The one most relevant for this thesis is the one described by von Hardenberg and Bérard [2001]. Here, finger tracking and hand posture detection are used for interacting via bare hands on a front-projected direct-touch display. Hand postures are used, e. g., for supporting presentations or brainstorming. Even though the total number of recognizable hand postures is large, there are some restrictions. In particular, the hand posture detection and finger tracking in Hardenberg's approach relies on front-projected displays as additional hardware (IR LEDs, camera) is required behind the display, its accuracy depends on the projected image, and his approach requires additional image processing.

2.3.3 *Bimanual Interaction*

Benefits for using bimanual input in graphical user interface tasks were described as early as in 1986 by Buxton and Myers [1986] and motivated the development of bimanual interaction techniques as described in Section 4.3. The studies by Buxton and Myers showed benefits for using the non-dominant hand for reference changes such as scrolling while using the dominant hand for precise selection.

One year later Guiard [1987] described a theoretical model for understanding the nature of this bimanual action called the Kinematic Chain. The nondominant hand remains near the root of the kinematic chain and can be used for coarse hand movements while precise selection is achieved lower on the kinematic chain through the dominant hand.

The performance of mice vs. directly touching a large digital table for one- and two-handed interaction was compared by Forlines et al. [2007]. The results of their experiments indicate that users benefit from direct-touch input for bimanual tasks, even though the direct-touch modality did not lead to greater performance in terms of speed and accuracy for unimanual tasks. Forlines et al. suggest, that when considering other factors such as fatigue, spatial memory and awareness of others's actions in a multi-user setting, direct-touch may still be the favorable modality. Hinckley et al. [1997] found that performance was significantly reduced when the roles of the dominant hand and the non-dominant hand were reversed.

2.3.4 *Seamless Transition between Direct and Indirect Interaction*

Even though many approaches exist for interaction at a distance and interaction up close, researchers only recently began to investigate potential benefits of providing a seamless transition between these two types of interaction. The work that is presented next has been a motivation for investigating an approach for seamless transition between direct and indirect interaction (see Section 3.3 and Section 4.3).

The Tractor-Beam input device is a 3D tracked stylus that enables transition between indirect and direct interaction on a digital table [Parker et al., 2005]. When held against the table surface the device acts as an absolute stylus. When it is lifted and the pen tip faces the table the device acts as a laser point. Even if accurate selection of distant targets may be error prone and limit the scope of tasks that can be achieved on large displays, user comfort must also be considered when selecting an interaction technique. Parker et al.'s results from a user study clearly indicate that users preferred to use a pointing interaction style to select distant objects, even though it was slower for selecting far, small targets compared to the touch input technique.

Vogel and Balakrishnan [2005] explored freehand pointing and clicking interaction with a very large high resolution wall display. They designed, implemented and evaluated three techniques for gestural pointing and two for clicking. Due to the lack of kinesthetic feedback they use subtle auditory and visual feedback. With all their techniques the authors simulate single-button mouse clicks or touch screen interaction with a single finger. Vogel et al.'s approach allows easy transition from distant pointing to direct-touch as only the physical hand and no extra hand-held device is used for pointing. To achieve this independence of hand-held pointing devices a Vicon system (http://www.vicon.com/) is used for tracking a glove that is equipped with passive markers. To date these system are rather expensive and were not available during this project.

2.4 SUMMARY

Painting systems allow the user to create digital paintings from scratch in a variety of ways and, therefore, offer a large set of parameters in often cluttered WIMP interfaces. However, some approaches leverage alternative user interfaces to allow for a more direct painting experience.

Digital painting systems provide users with complete control over the shapes that they create. In contrast, many NPR approaches allow users only to select primitives out of a predefined set. Similar to digital painting systems NPR approaches often suffer from cluttered user interfaces and allow the creation of renditions only in lengthy adjustment-rendering loops.

A goal of this thesis is to investigate alternative interaction techniques for TSLDs in the context of NPR. They target on enabling a broad range of users to concentrate on the image creation process by avoiding complex user interfaces.

3 CONCEPTS

This chapter presents the conceptual foundations of this thesis. It should clarify why certain techniques were developed and how their design was guided. First, the iterative design process that was followed throughout the project is described. A special focus is placed on an informal observational study that laid the foundation for subsequent development. Afterward, the notions of *expressive interaction*, *seamless transition between indirect and direct interaction*, and *degrees of precision* are introduced and it is described how they relate to the techniques presented in Chapter 4.

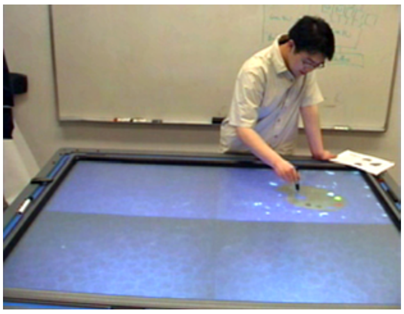
3.1 ITERATIVE DESIGN PROCESS

The development of the interaction techniques described in this thesis are based on the iterative design methodology as a cyclic process of design, testing, and development (as described, for example in [Mayhew, 1999; Preece et al., 1994]). The following section focuses on an informal observational study, that was carried out at the beginning of the project, as it laid the foundation for the further development of the project.

When artists explored the Interactive Canvas system [Schwarz et al., 2007] the novel approach of creating non-photorealistic renditions using rendering primitives was welcomed. However, they felt, that the existing interface limited their abilities to concentrate on the image creation process.

These comments motivated an informal observational study to evaluate the existing user interface of the Interactive Canvas system. Strengths and weaknesses of the existing system in the process of image creation should be revealed.

The study was conducted with three participants (two male, one female) on a Smartboard tabletop with a physical size of 146 cm × 110 cm and a resolution of 2800 × 2100 pixels. Two of the three participants had used the Interactive Canvas briefly before. One had an artistic background. Participants were introduced to the system through oral and written instructions. They had approximately 20 minutes time explore the system by creating an image of their choice. The participants were asked to think out aloud, and were videotaped while interacting. A quick manual was available as reference. Field notes were taken and it was registered which tools they used frequently. Afterward, they were interviewed and asked what they enjoyed during interaction and which aspects of the system they experienced as cumbersome. Figure 9 (a) shows one of the participants interacting with the palette as main widget.



(a)

(b)

Figure 9: A participant explores the Interactive Canvas during an informal observational study (a). Icons for orienting and moving primitives (highlighted with green rings) on the palette widget (b) were confused frequently by participants.

Observations of this study suggested that alternative interaction techniques for creating non-photorealistic renditions on TSLDs need to be investigated. Important observations are described next to motivate the design decisions that were made in subsequent project stages.

While exploring the Interactive Canvas participants had frequent problems to remember which tool they currently used. One participant said that it was "hard to remember which mode I am in". These problems occurred even though the currently active tool was indicated by a highlighted icon on the palette. This may be explained by the fact, that the palette was not always visible for the users while interacting on the large tabletop display. When participants wanted to access certain tools they had problems to identify the corresponding icon. For example, two out of the three participants mistook the icons for enlarging strokes and increasing the number of strokes several times. One said that the "icons are confusing". All of them confused the icons for orienting and moving the primitives several times (these icons are shown highlighted in Figure 9(b)). Even when the appropriate icons were identified, participants had problems in selecting them. This could be explained by inaccuracies in the underlying hardware position detection process, but could also be due to the relative small size of some icons. Participants also had problems to remember how some tools worked. One participant mentioned that there were "too many modes."

The participants could access over twenty functions through pie menus, buttons and sliders on the palette widget and further ones through keyboard shortcuts. They especially liked that colors could be selected and mixed easily. After all functions had been explored briefly by the participants, only some of them were used frequently. This could be explained by the limited time, that the participants had for exploring the system. Nonetheless, it showed the need to prioritize functions and make the most frequently used ones accessible as easily as possible.

During the exploration of the Interactive Canvas as well as in the following interviews, participants mentioned that they liked how a rendition can be quickly created using the available rendering primitives. However, one participant mentioned that she would like to use an extended set of primitives. The others requested a possibility to create their own shapes.

The observations that participants used only a small subset of functions and that they had problems to select them through clicking on menus, icons, and sliders indicated that they may be distracted by the existing user interface. Together with the participants's wish to use additional primitives these observations indicated the need for an adapted interaction approach, that avoids a WIMP user interface, that allows users to create and employ their own shapes, and, that consequently enables them to directly concentrate on the expressive image creation process.

Potential benefits of interacting at a distance with large displays, such as an overview of the whole display, led to the investigation of indirect interaction techniques. Interaction approaches found in games for the Nintendo Wii® motivated the investigation of the Wiimote as an input device for indirect interaction.

Remote pointing techniques with the Wiimote have been iteratively developed and techniques based on Wiimote gestures (see Section A.3) have been investigated throughout this project.

As some interaction research suggests, it is beneficial to allow users to interact with large displays at the distance the users seem appropriate but to change that distance at will. In order to provide a seamless transition between interaction at a distance and up close, concepts and techniques using speech and bimanual interaction techniques have been investigated in this project.

Developing a speech interface that could be used by a variety of people was hard to realize and finally aborted (see Section A.4). In several design-test-cycles two bimanual interaction techniques have been developed. The bimanual techniques allow users to access common tools in a consistent way. As these two techniques promised to enable users to interact at a distance and up close, they have been evaluated to find out in which situations users would favor one technique over the other.

3.2 EXPRESSIVE INTERACTION

A goal of this project was to enable users to effectively and expressively create stroke-based non-photorealistic renditions without the need to think about the interface itself. This motivated the development of user-sketched strokes as well as the design and implementation of hand postures for interacting with the strokes [Grubert et al., 2007a, 2008]. While user-sketched strokes extend the palette of rendering primitives, hand postures allow people to interact directly with non-photorealistic renditions on TSLDs and avoid a WIMP-based interface.

3.2.1 *User-Sketched Strokes*

In digital painting systems users can create a wide variety of 2D shapes with digital tools such as brushes or pens. However, with many applications such as Adobe Photoshop®, users have to be skilled in order to create elaborate structures. The once created structures are also often hard to reuse. The most common scheme is to copy and paste the region that should be reused in other region of a painting. With many NPR approaches (as described in Section 2.2) various visual styles, such as pointillism, hatching, or painterly rendering, can be obtained. To obtain these styles, often an input model, such as an image, is used to abstract from it using NPR primitives. Even though users are able to steer the rendering process in some approaches (e. g., [Kalnins et al., 2002; Schlechtweg et al., 2005; Schwarz et al., 2007]), they often cannot build their own shapes to be used in the rendering process (with [Kalnins et al., 2002] being an exception). To combine the advantages of freely definable shapes in digital painting systems with the ability to quickly abstract from input images with NPR primitives, user-sketched strokes were developed during this project. A goal was to allow unexperienced users to create smooth strokes quickly, to be used as rendering primitives, ranging from simple line segments to complex structures such as Chinese letters. This ability to sketch a variety of 2D shapes supports users in creating expressive renditions.

3.2.2 *Hand Postures for Expressive Interaction*

When examined closely, TSLDs can be used similar to the setup that finger painters employ, when creating real paintings on scaffolds. A large canvas provides sufficient space for creating expressive paintings. A single finger, multiple fingers, or even the whole hand can be used by the artist to transfer paint onto the canvas. Afterward, the paint can be further distributed and smeared through hand interaction.

In contrast to direct-touch interaction with a single finger or a pen, hand postures go beyond the mere indication of location. Therefore, hand postures allow users to explore a wider design space for interaction with stroke-based NPR on TSLDs.

In order to support expressive interaction, hand postures should be easy to remember and easy to use. Ease of use is supported by employing hand postures that can be formed by a broad range of users. Even though several dozen different hand postures can be formed by a skilled person, such as a piano player, only a small subset is easily formed and remembered by most people. Employing metaphors for the use of hand postures and mapping them to similar functionality in different contexts helps to reduce the cognitive effort needed to remember the hand postures and their associated functions.

Inspired by the finger painting approach mentioned above and given the technical constraints, that limit the set of recognizable hand postures (see Section 4.1), four different hand postures have been designed that work for a variety of users. These are one finger, two fingers, fist, and flat hand (as shown in Figure 10). Their implementation and use is described in Section 4.1.

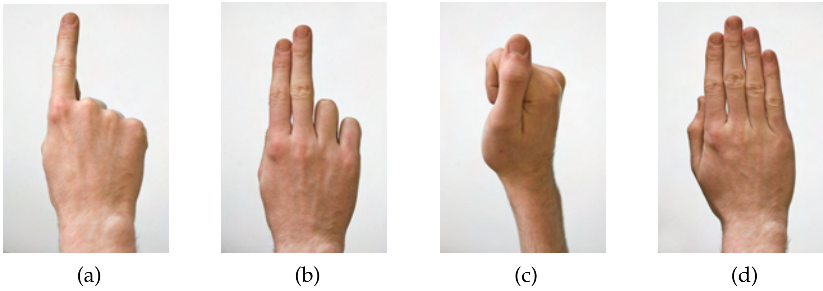


Figure 10: The four hand postures one finger (a), two fingers (b), fist (c), and flat hand (d).

Hand postures are meant to allow users to work expressively when standing up close to a TSLD. However, there are situations when interaction at a distance are appropriate and when a transition between direct and indirect interaction needs to be possible.

3.3 SEAMLESS TRANSITION BETWEEN INDIRECT AND DIRECT INTERACTION

TSLDs with ever increasing resolutions and sizes allow users to work up close with detailed information or to step back to gain an overview of the entire screen space. Both types of interactions have unique advantages. Interaction up close with direct-touch techniques allows users to precisely steer even small objects on the display. In the context of digital painting, standing close to the display permits interaction that is similar to the way artists interact with a physical canvas. Small wrist movements or even whole body movements can be employed to interact with the display and enrich the palette of expressive interaction. On the other hand, standing for a long time close to a display, for example to produce a detailed painting, can also lead to fatigue. In contrast, remote pointing allows the user to sit while interacting with a vertical display and, therefore, can be more comfortable over an extended period. Interacting at a distance also provides a better overview over the entire display.

Therefore, a goal of this thesis is to provide interaction techniques that combine the advantages of interacting up close and interacting at a distance by allowing a seamless transition between those contexts.

This transition can be supported by not requiring the user to change the input device [Vogel and Balakrishnan, 2005] or to allow a convenient switch between input devices. Both is possible with the bimanual interaction techniques developed in this project. Providing a consistent way to access functionality for indirect and direct interaction techniques also can support an easy transition between the two as the cognitive effort to access the same system functions with both techniques is reduced. Two bimanual interaction techniques have been developed in this project. One is designed for interaction up close and is based on direct-touch. The second one is an technique for interacting at a distance and uses the Wiimote for remote pointing. Both techniques allow to access system functionality in a consistent way with one hand while the other hand is used to indicate the location where a chosen action will be performed. When users interact at a distance they indicate location by remotely pointing with Wiimote. When interacting up close with a TSLD they can touch the display with either the Wiimote or their finger. Figure 11 shows an user interacting with both bimanual interaction techniques.

(a) (b)

Figure 11: Bimanual interaction techniques permit a seamless transition from remote pointing (a) to direct-touch (a).

3.4 DEGREES OF PRECISION

In traditional painting, artists work at various levels of precision. The choice of painting tools such as brushes of different sizes or of paint that is applied affects the achievable precision. In addition, the movement of the body also has an influence on precision because, it can range from small wrist motion to full-body motions. This is often deliberately used to achieve different effects by traditional painters.

Therefore, the interaction techniques presented in this thesis deliberately also support similar freedoms. These techniques provide different levels of precision that allow users to choose the technique that suits them and to switch between techniques if desired.

For example, using the developed bimanual interaction users can precisely distribute and modify strokes with direct-touch. However, while remote pointing is inferior to direct-touch with respect to accuracy and control [Myers et al., 2002], it allows for quick distribution of strokes in large areas. This resembles the artistic process during painting where a background layer is created using broad strokes that are often not as precisely distributed on the canvas as the subsequent smaller strokes. Another example is how users can explicitly employ various levels of precision that different hand postures offer them (also see Section 4.1). Precise distribution of strokes in a small area is possible with a one finger posture. While positioning the fist posture is less accurate due to the increased occluded display area, using this posture enables users to quickly fill a larger area with strokes.

Imprecise techniques are utilized throughout the system to invite users to a playful exploration of creating non-photorealistic renditions. For example, when strokes are distributed by users, they are placed and oriented randomly in an area of influence to obtain an appealing look, without the need for further modification. The system is also designed to forgive small inaccuracies. For example, when users sketch strokes, wiggly movements, due to hand and arm jitter, are tolerated and still create a smooth stroke. This general approach combines different techniques for introducing and tolerating imprecision to invite users to explore the system freely.

3.5 summary

An initial observational study laid the foundations for subsequent developments of this project. The study showed, that it is desirable to enable users to directly concentrate on the process of creating non-photorealistic renditions. Therefore, interaction techniques that target on allowing users to interact expressively and intuitively have been developed throughout this project. Expressive and intuitive interaction are supported in several ways. Hand postures allow users to access frequently used functions directly. User-sketched strokes enable people to create a wide variety of 2D shapes that can be used as NPR primitives.

Furthermore, the interaction techniques presented in this thesis focus on the use of TSLDs. These displays can be seen to resemble the setup that artists use when creating traditional paintings and, therefore, can support an intuitive painting process. Large displays have different advantages when users interact up close, e. g., the possibility to precisely steer even small objects, or when users interact at a distance, such as an overview of the whole display. Therefore, bimanual interaction techniques have been developed that allow users to interact at the distance they seem appropriate. The bimanual interaction techniques focus on providing a consistent way of accessing system functions. This way, a seamless transition of interacting at a distance and up close is supported.

The techniques developed during this project provide and tolerate various levels of precision. Users are invited to explore the non-photorealistic image creation process freely and to work at the level of expressiveness they seem appropriate.

4 REALIZATION

This chapter gives insights how the previous introduced concepts were used to realize interaction techniques and user-sketched strokes. Principals, classification and use of hand postures for interaction up close on TSLDs are described first. To allow for interaction at a distance remote pointing techniques have been investigated. They are presented afterwards. To combine the advantages of interaction at a distance and up close bimanual interaction techniques have been developed and are described in the subsequent section. Then insights into algorithms and data structures for creating user-sketched strokes are given. In order to better match users expectations when interacting with these strokes, instantaneous buffers have been employed. They are described in the last section of this chapter.

4.1 HAND POSTURES FOR DIRECT INTERACTION

As described in Chapter 2, direct interaction with TSLD is often realized through direct-touch or pen input, and more recently through multi-touch and hand posture interaction. While an investigation of multi-touch techniques was started during the project, it was finally aborted due to severe limitations in the recognition capabilities of the available hardware. Therefore, this section concentrates on the use of hand postures for direct interaction with large displays. In the following an overview is given about principals of hand posture detection, development and evaluation of an initial set of hand postures and adaption of this initial set to work with a wide spectrum of users.

4.1.1 *Principals of DViT Touch Recognition*

Smart Technologies Digital Vision Touch (DViT) equipped Smartboards have been used for posture detection(a tabletop and a wall display). One advantage of the DViT technology is that it can be used on various kinds of large displays, such as front-projected, rear-projected, LCD, and plasma displays, regardless of whether they are mounted vertically or horizontally. In order to provide touch recognition for large displays a frame is mounted on top of the display. This frame is equipped with one camera in each of its four corners and infrared LEDs along its edges. The position of up to two touching objects as well as approximate width and height of their bounding boxes can be accessed through an Application Programming Interface (API). The position of a touching object is determined through triangulation. For a more detailed introduction into the triangulation of positions see Section A.1. Simply put, an angle is derived for each camera and each boundary point, by determining the position of the first and last infrared LED inside the frame's edges that is obscured by the touching object (see Figure 12). As each camera recognizes two angles, the coordinates of two boundary points of an touching object can be computed with two opposing cameras (see Figure 12). The remaining two cameras can be used to determine the position of a second object or to make the recognition of the first object more stable. Given the two (respectively four) boundary points width and height of the bounding rectangle can be computed.

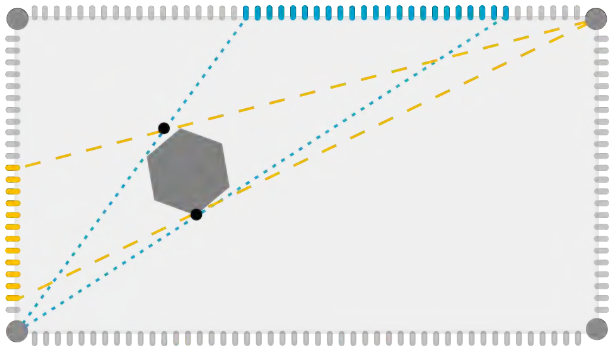


Figure 12: With each camera two angles of an touching object can be recognized. The angles are used for determining the position of object boundary points through triangulation.

4.1.2 *Hand Posture Classification*

Width and height information are used to compute the area of a bounding rectangle of an touching object. This area in turn is used for posture classification.

On the first glance the number of detectable postures is limited only by the number of closed areas that can be formed with a hand. However, physical, intra-personal, and inter-personal factors as well as instabilities in the hardware recognition process severely limit the number of recognizable hand postures. A lower and upper bound for detectable areas is given by the smallest and biggest area that can be formed with a hand or a pen on a flat surface (e. g., the of a tip for the smallest area; a flat hand for the largest area). One person can touch a surface with the same body part in different ways. For example, an index finger can be put onto the surface in a rather acute angle or a flat angle which in turn changes the touched surface area. Inter-personal differences occur in, e. g., hand shapes and hand sizes. Instabilities in the hardware recognition process also have a negative impact. Position and area values are sampled once during each rendering pass. Even though these values are stable in their averages they vary significantly on a value to value basis. To overcome this problem the first few area values are averaged as soon as a touch-down event is registered. Experiments showed that averaging the first four samples is sufficient to classify a posture reliably. Another problem arises when postures are created or released. For example, when the hand is lifted from the surface, the sampled values are getting smaller, in turn all hand postures that have a smaller area than the current one would be recognized as well if hand postures would be recognized in each rendering frame. To avoid this problem, the hand posture is registered after the first four area values are averaged and does not change until a touch release event occurs. Finally, the DViT technology does not deliver constant width and height values for the same posture on all parts of the screen. The widths and heights of the bounding rectangle are most accurate in the corners of the Smartboard but increase toward the center of the screen. Therefore, these values are weighted by the position to ensure constant posture recognition across the entire screen. The weights are smallest at the display center as the detected width and height values are biggest at the center. The width and height values decrease toward the edges and corners of the display. Therefore, the weights are increased in these areas.

4.1.3 *Experiment*

In order to provide stable hand posture classification among different users, the initial set of hand postures (as introduced in Section 3.2) was evaluated in an informal experiment. Five people of different physiognomy (two women, three men, age ranging from 22 to 27 years) participated in this experiment. Even though the exact dimensions of the participants's fingers and hands were not measured, care was taken that a wide variety of hand shapes and sizes, ranging from small hands with short fingers to large hands with long fingers, was covered.

Each user placed their dominant hand at 17 positions across a Smartboard tabletop display (see Figure 13). At each position, the participants touched the screen with each of the initial seven hand postures in two variations. Variations in the placement of hand occur across the screen as was shown for a single finger in [Forlines et al., 2007]. The 17 positions and two variations of placing a hand posture were chosen to account for this. 500 samples of area values were taken at each position. Average, minimum and maximum area values were recorded and used to determine reliable area bounds for the postures.

It turned out that only four out of the seven initial hand postures could be discriminated consistently across participants, namely one finger, two fingers, fist, and flat hand (as shown in Figure 10). For these four hand postures average minimum and maximum area values that work across different users were derived. These values were not computed in screen space but in physical space (in inch^2 and m^2) and so can easily be used among Smartboards with different physical sizes and different input and output resolutions. For a single user, all seven hand postures are still usable if they would run a short calibration application before using hand postures for direct interaction.

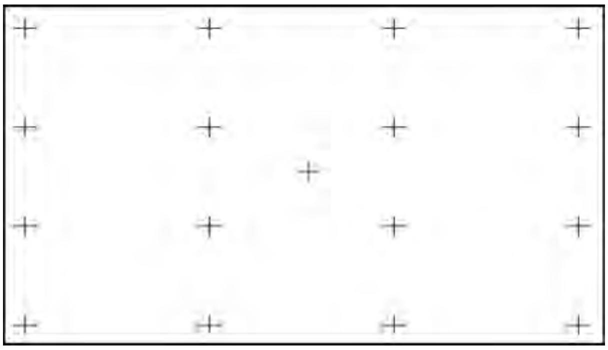


Figure 13: Crosses indicate target points where participants should touch with hand postures during an experiment for classifying postures for a variety of users.

4.1.4 *Use of Hand Postures*

The limited set of hand postures that can be easily discriminated presents the challenge that not more that few functions can be accessed in a given context. However, the set of tools being present in the initial Interactive Canvas system [Schwarz et al., 2007] was larger than the one that can be supported with four hand postures. To address this problem, frequently used functions have been identified and made accessible through hand postures. Additionally, the number contexts in which hand postures can be used was increased.

When used to interact with user-sketched strokes hand postures should support the sketching of strokes as well as the use of these strokes on the canvas by distributing and modifying them. To allow this distinction, a pen tray that is attached on the bottom of a Smartboard is used. It can detect, if up to four pens are lifted from the tray and, therefore, provides four contexts in which hand postures could be used to access up to sixteen functions. While remembering sixteen different functions that are accessed by the same hand postures may be hard, providing one additional set of hand postures by lifting one pen seems to be appropriate. By lifting the pen out of the tray (see Figure 14(a)) the user also can switch naturally between a distribution context, in which strokes are applied to the canvas and modified, to a sketching context, in which the user can sketch its own strokes that are used afterward.

In the sketching context strokes are sketched with the pen, a single finger, two fingers, or the fist (see Figure 10). A flat hand is used to erase the strokes partially or completely. This is inspired by the way a chalk board can be cleared with the hand. Afterward, they are distributed with a single finger or the fist. While the use of one finger allows to precisely distribute a few strokes in a small area the fist is used to fill a wider area with more strokes. The two finger posture is used to align distributed strokes and, therefore, allows to emphasise certain image structures. Finally, the flat hand posture is used to erase strokes (see Figure 14(b)). Instead of disappearing suddenly the strokes grad-

(a) (b)

Figure 14: The pen tray (a) is used to switch between a sketching and distribution context. A user gradually fades out strokes with a flat hand in the distribution context (b).

ually fade out if the flat hand dwells over them. This behavior also allows to obtain various levels of transparency and therefore enables the blending of strokes that are stacked on top of each other. Figure 15 shows strokes with various width in the sketching and distribution context.

(a) (b)

Figure 15: Strokes with variying thickness can be sketched with pen, one finger, two fingers, fist (a). Strokes are rendered to texture and distributed with the one finger posture or the fist posture (b). The different colors of the strokes on the left side are derived from an underlying color image.

4.2 Interaction at a distance

As described in Section 3.3 when users interact with large displays at a distance they gain a better overview of the screen area and have the possibility to find a comfortable seating position. Interaction techniques that were developed for interaction at a distance with large displays are described afterward. While the direct interaction techniques that are described in Section 4.1 are suitable for horizontal and vertical displays, the remote pointing techniques described in this section are specifically meant to be used on vertical displays. Wiimote gestures for issuing commands, such as creating or erasing strokes were designed and some aspects have been implemented (see Section A.3).

4.2.1 *Remote Pointing*

Remote pointing is often used with the help of raycasting techniques, but alternative input techniques can be used as well (see Section 2.3.1). The Nintendo Nunchuk, as an elastic rate control device, and the Wiimote, based on isotonic position control, have been investigated for remote pointing. Both devices were chosen because they provide an inexpensive way to equip large displays with robust remote pointing capabilities. In order to generate a click event several buttons on the Nunchuk and the Wiimote could be used. While the Nunchuk provides two buttons that can be pressed the Wiimote provides seven possible buttons to be pressed. On the Wiimote the B button was found to generate the least shake when while pressing it (see Figure 17(b)). To generate less shake is important for gaining a high as possible pointing precision with the controller. That pressing the B button generates the least shake can be explained by the fact that the direction of the finger movement when pressing it is parallel to the pointing direction. When pressing the other buttons that are on top of the Wiimote the press direction is perpendicular to the pointing direction and, therefore, the press movement generates more shake.

Elastic Rate Control vs. Isotonic Position Control

First the Nunchuk with integrated joystick as elastic input device was investigated for positioning a cursor on large displays. The device provides an analog joystick. This joystick provides rate control, its displacement is mapped onto the velocity of the cursor on the screen. Potential advantages of an elastic device, based on rate control, compared to a device, based on isotonic position control, are e. g., reduced fatigue during operation as well as smoother and more stable positioning of the cursor [Zhai, 1998]. However, while testing the Nintendo Nunchuk in informal experiments participants complained about the "indirect" feeling in front of a large wall display and the relative slow movement of the cursor. If the control-to-display ratio was modified to support faster cursor movements it was hard to achieve satisfying target acquisition accuracy. A possible explanation is that only small deflection is possible with the Nunchuk joystick. Mapping the magnitude of joystick deflection to a reasonable fast and yet precise movement was not successful.

In a second step remote pointing with the Wiimote was implemented. As a device that is based on isotonic position control, the position of the controller pointing toward the screen is mapped onto the cursor position. On the one hand, potential drawbacks exists, compared to an elastic rate control based device, e. g., more fatigue, a limited movement range, and more difficult coordination of controller position. On the other hand there are advantages such as faster target acquisition times that are closer to target acquisition times of direct-touch as well as easiness of learning compared to isometric and elastic rate control devices [Zhai, 1998]. Finally these advantages, positive user feedback and the possibility to use the Nunchuk for bimanual interaction (see Section 4.3) were decisive when finally choosing the Wiimote over the Nunchuk as remote pointing device in this project.

Challenges of Wii Remote Use for Remote Pointing on Large Displays

The Wiimote was designed specifically for use in an entertainment environment on regular-sized TV screens. There are challenges, when using the controller for remote pointing on large displays. Two single infrared light emitting diodes (IR LEDs) (or arrays of them) are placed below or above a display in a known distance. An integrated infrared camera is integrated into the front of the Wiimote. This enables 2D position tracking of the controller relative to the IR LEDs through triangulation (see Section A.1) and, hence, positioning of a cursor on screen. The viewing angle of 45° limits the range of tilt and side offset of the Wiimote where tracking of the IR LEDs is possible (see Figure 16). While the maximal side offset is half the distance of the Wiimote to the IR LEDs it quickly becomes smaller as the controller is tilted. The limited resolution of the camera allows the use of the controller only up to approximately 4 meters with the use of standard IR LEDs that are used e. g., in Nintendo's Wii's "sensor bar", which is a bar with five IR LEDs on each end. The exact resolution is unknown but the radii of the IR LEDs only can be approximately determined unit-less within a range from 1-16. The Wiimote could be used conveniently at the wall display that was used during this project. However, for larger displays more powerful IR sources should be provided to support the use of the Wiimote IR sensing capabilities at larger distances. For remote pointing

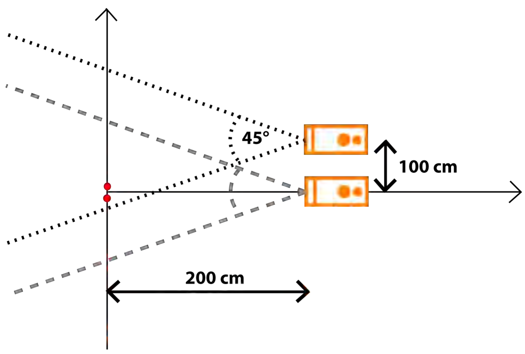


Figure 16: At a pointing distance of 2 meters the maximal lateral displacement is 1 meter if the Wiimote is held orthogonal to the screen.

on horizontal displays the Wiimote can be cumbersome to use. This is due to uncomfortable arm and hand positions that result from pointing directly on a horizontal display (which usually is not on ground level, but most likely resembles a table).

Due to the increased fatigue compared to the interaction based on rate control with the Nunchuk, user tend to relax their arms by pointing off screen after a task is accomplished. Two variants of how the cursor behaves in this situation have been investigated. The cursor can be reset in a default position on screen (e. g., the top left corner of the screen). As an alternative it remains at its last known screen position. Depending on the ratio of the cursor-to-display size it might be favorable to reset the cursor to a default location when this ratio becomes small (i. e., the relative size of the display compared to the cursor size is large). This can be explained by reduced search times for the last cursor position and the memory effect that occurs when the cursor is reset to the same position several times. Even though informal tests suggested this, no reliable values for this ratio could be determined that could be used for decision finding in other projects. Finally the resetting to a default corner position was chosen.

4.3 BIMANUAL INTERACTION

As described in Section 3.3 it is desirable to combine the strengths of interacting both at a distance and up close. After the available functions in the initial system were prioritized, different interaction techniques for providing a seamless transition between indirect and direct interaction were designed, in particular using speech input for issuing commands (e. g."create strokes", "move strokes") and bimanual interaction. After several design iterations two bimanual interaction techniques evolved that provide a consistent user interface to the most common functions. Direct-touch allows users to interact up cloase and remote pointing with the Wiimote is used for interacting at a distance. As the Wiimote can be easily put into a pocket or used for touching the screen when up close it enables a seamless transition between indirect and direct interaction. Figure 17 and Figure 11(b) show both techniques in use.

(a) (b) (c)

Figure 17: While a user is seated (a) she points toward the screen with the Wiimote in her dominant hand (b). She indicates which effect the dominant hand interaction an action with the Nunchuk in her non-dominant hand (c).

In this setting, the non-dominant hand controls a Nintendo Nunchuk to indicate what effect an interaction with the dominant hand will have, regardless whether direct-touch or remote pointing is used for indicating position with the dominant hand. The Nunchuk device provides a small joystick that can be controlled with the thumb and two buttons (button C and button Z) that can be triggered with the index finger. The joystick is positioned inside an octagonal shape and, thus, provides eight discrete positions on the perimeter that can be acquired easily as well as an additional center resting position. These nine positions are used to access nine primary functions. The constantly visible tool menu shown in Figure 19(a) provides visual feedback about the currently active function. This tool menu is located at the current cursor position as shown in Figure 18. An alternative configuration was investigated by placing the icon menu in a corner of the screen. Initial user tests showed that this led to overview problems when users interacted through direct-touch and, thus, when they were standing close to the display. A study described in Section 5.1 showed that the tool menu that is constantly visible at the current cursor position does not distract while interacting with the canvas.

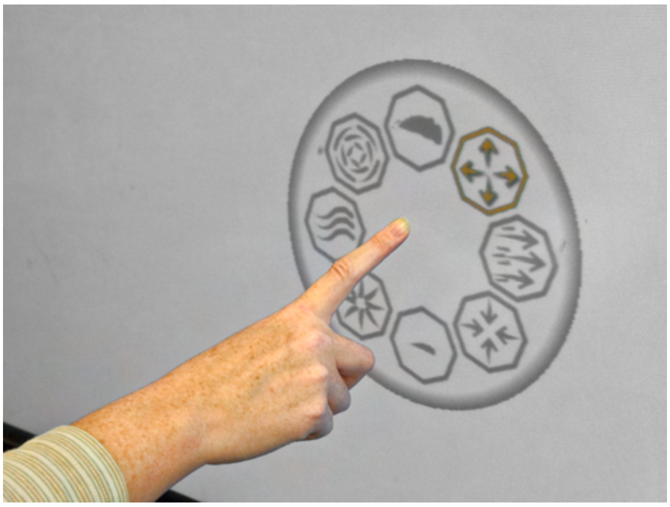
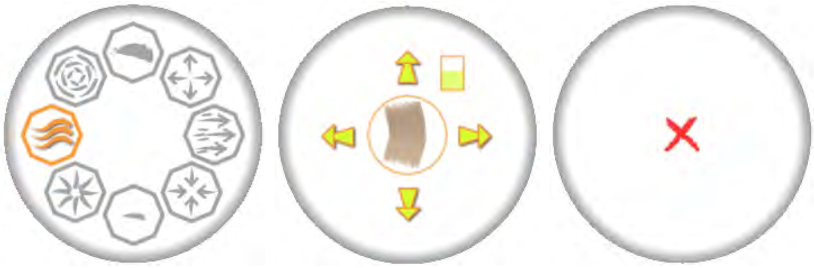


Figure 18: When touching the screen the tools menu appears at the touch location. The repel tool is selected with the Nunchuk.

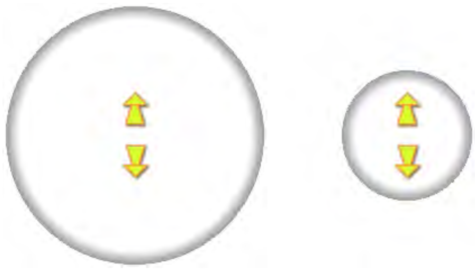
To activate and maintain eight main functions users have to move into and hold the joystick at one of the eight directions throughout dominant hand interaction. They can also press and hold the C and Z buttons to activate additional functions as shown in Figure 19. While a higher cognitive load may be involved, compared to techniques, such as clicking a mouse button, that do not require users to actively hold a state, it was shown before that forcing the user to maintain the position prevents mode errors [Sellen et al., 1992]. The functions are orienting primitives in a linear, circular, or star shape, enlarging and shrinking objects, moving primitives along the path of the tool, an to repel, or attract them. When the joystick is in the default center resting position, primitives are created when the display is touched or the



(a) (b) (c)

Figure 19: Tool menus that can be accessed by using the Nuchnuk joystick and buttons. Eight primary functions are accessed by using the Nunchuk joystick, the align linear function is selected (a). The stroke selection menu is accessed by holding button C (b). The erase function is accessed by pressing button C and Z simultaneously (c).

Wiimote engaged with the B Button. Primitives are erased by holding the C and Z buttons and interacting with the dominant hand. The area that is affected by interacting with the dominant hand may be changed by holding the Z button and moving the joystick up or down (see Figure 20). Feedback in form of a circle showing the area of influence is given.



(a) (b)

Figure 20: The resizing of the area of influence can be accomplished by pressing button Z and simultaneously moving the Nunchuk joystick up (a) or down (b).

Different primitive types can be selected by holding button C and moving the joystick left or right (see Figure 19(b)). The number of primitives that are created or erased in one time step is adjusted by holding button C and moving the joystick up or down. The exact number of created or erased primitives is also dependent on the affected area in order to ensure a consistent primitive density.

More functions, such as color mixing can be accessed through an additional menu system that has to be invoked explicitly, e. g., by clicking outside the screen (as there is a touch sensitive frame outside the visible screen area). This menu system was designed (see Section A.2) but not implemented due to time constraints.

4.4 USER-SKETCHED STROKES

An informal observational study (see Section 3.1) revealed that users asked for means to create their own primitives to work with. A stroke data structure for achieving this goal was developed. Interaction with those strokes is a two step process as described in Section 3.2. First, strokes are created in a *sketching phase*. In the following *distribution phase*, they can be used just like other pre-defined strokes; i. e., they can be distributed and modified on the canvas. In the following the sketching phase and the stroke data structure for stroke sketches are described in detail.

In the sketching phase, a stroke is represented by using a data structure that maintains a list of control point with their associated parameters, such as width or saturation. In contrast to predefined textures, this data structure allows users to modify and change their strokes at any time while they are still in the sketching phase. In order to sketch a stroke, the user indicates the path of the new stroke by using pointing. Stroke control points and widths are generated by a modified Dynadraw algorithm [Haeberli, 1989].

In Dynadraw, one creates smooth, calligraphic strokes using a brush and spring model. The brush which does the actual painting is connected to the current cursor position through a damped spring. As the user changes the cursor position the brush updates its position, and a stroke point with associated parameters is generated at the position of the brush rather than at the cursor's position. Several brush parameters including mass, velocity and friction influence the behavior of the brush and, therefore, the appearance of the resulting strokes. These parameters can be modified according to an intended appearance. For each hand posture, lower and upper bounds for the stroke width, and the mass of the virtual brush are adopted as shown in Table 1 and Figure 15(a).

Posture	Min. Stroke Width	Max. Stroke Width	Brush Mass
Pen	5 px	15 px	0.5 px
One Finger	10 px	30 px	1.0
Two Fingers	20 px	50 px	2.0
Fist	35 px	80 px	2.0

Table 1: Parameters used for adopting the stroke appearance according to an employed posture.

In order to render a stroke its data structure holds vertex and index information that are used to create an OpenGL quad strip. Alternatively the stroke also can be rendered as a series of points, to achieve a different rendering style. Vertex positions for a given control point cp that are part of the OpenGL quad strip are derived as shown in Listing 4.1.

```
for each cp
{
        if (cp != firstCP && cp && lastCP)
        {
                compute vector vp between cp and predecessor;
                compute vector vs between cp and successor;
                compute unit bivector bv between vp and vs;
                v0.coordinates = cp.coordinates + (bv / width/2).
                    coordinates;
                v1.coordinates = cp.coordinates - (bv / width/2).
                    coordinates;
        }
        ...
}
```

Listing 4.1: Computation of Vertex Positions.

To deal with the special cases of the first and last control point, an angle of 180° between the current control point and the successor (at the beginning of a stroke) or the predecessor (for the last control point) is assumed.

Strokes created with the original Dynadraw algorithm start and end with circles that are wider than intermediate stroke segments. This is due to the speed of the mouse cursor that is reversely mapped to the stroke width (the faster the mouse movement, the smaller the stroke width). As this speed is zero at the beginning, and (generally) lower at the end of the stroke, than at intermediate segments, the circles at these endpoints are bigger. The algorithm was modified to generate cone-shaped ends by blending a small default width at the end points into the widths that are computed for intermediate points. The blending is described by the transfer function:

$$tv(cns) = 1 - sin(\frac{\pi * cncp}{2 * tncp})$$

cns: current number of control points.

$tncp$: maximum number of control points that are considered for blending.

An adjusted width cwa at an intermediate control point can then be computed through weighted linear interpolation between the initial width iw at the first control point and the width cw at the current control point:

$$cwa = (1 - tv(cns)) * cw + tv(cns) * iw$$

Figure 21 shows the schematic structure of a stroke and Figure 15 shows actual strokes rendered with the system that were created using different hand postures.

Depending on the width at the control points and the angle between adjacent line segments self intersection of the OpenGL polygon may occur. Because the stroke finally is rendered into a texture with a relatively low resolution (e. g., 512x512px) artifacts are rarely visible. If high quality output is required, approaches as discussed in [Zander

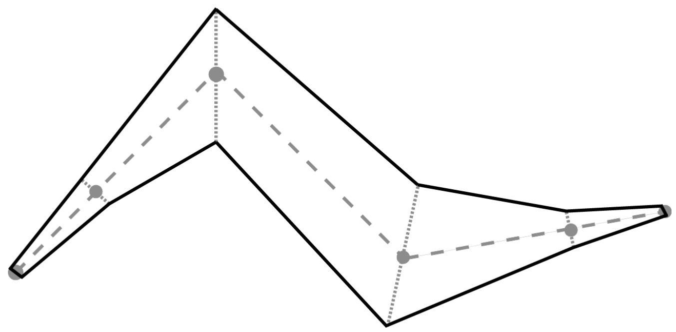


Figure 21: Stroke control points (black dots), vectors between control points (dashed gray lines), bisectors (dotted gray lines), and boundary lines of associated OpenGL quad strip.

et al., 2004] may be integrated to improve results and avoid the mentioned artifacts.

After a stroke segment has been created in the sketching phase, it can be erased completely or in parts. The user indicates erasing either by selecting the erase option on the Nunchuk joystick and pointing toward the area to be erased or uses the flat hand as erase posture if using direct-touch. If the beginning or the end of a stroke is erased, only the indices of the remaining OpenGL vertices, that are maintained in the stroke data structure, have to be updated to ensure a proper rendering. If a stroke is split in the middle by an erase action two new instances of the stroke data structures are created that contain the unaffected control points and vertices. The old stroke is erased afterward.

If users are satisfied with the stroke shape, they terminate the sketching phase to use the stroke as a rendering primitive. For this purpose the stroke is rendered into a texture, and from now on, behaves like other rendering primitives.

4.5 INSTANTANEOUS BUFFERS

Like the initial Interactive Canvas [Schwarz et al., 2007] this system is build upon the University of Calgary Tabletop Framework [Miede, 2006] and uses i-buffers (2D arrays) for responsive interaction on large displays. Information such as color, size, or orientation of primitives are stored into i-buffers through spatial tools. Primitives query these information and then decide how to render themselves. Advantages of using this approach is improved interaction responsiveness and the ability to manipulate a large number of primitives at once without the need for individual selection.

In the Interactive Canvas system primitive properties, such as size or orientation were stored in persistent buffers that exist for the entire run-time of the system. Persistent buffers represent actual object properties and are filled with default values. As primitives are moved across the display their properties constantly update. A side effect is that tedious adjustments of a primitive in one region can easily be lost if it is moved just slightly across the screen. This effect of implicit property changes was reported as confusing in the informal observational study.

To avoid this behavior instantaneous buffers replace the persistent buffers (with the color buffer being the only exception) in the current system. Instantaneous buffers store property changes that are reset after every rendering step; the actual properties are stored inside the primitives. This leads to a more comprehensible behavior of primitives as they are moved across the screen.

An example is the erasing of primitives. Each primitive has a opacity property that is maintained regardless of position changes. As the user applies an erasing tool, e. g., with a flat hand posture, the opacity of affected primitives is decreased. If the opacity of a primitive is below a certain threshold it pushes itself into a erase queue and is deleted afterward (see Figure 22).

Figure 22: A row of spiral strokes is faded out and finally erased.

4.6 SUMMARY

Several interaction techniques have been iteratively developed throughout this project. They target at enabling users to interact expressively and intuitively with non-photorealistic renditions. Remote pointing techniques have been investigated to allow users to interact with large displays at a distance. The Wiimote that offers isotonic position control was favored for remote pointing over the Nunchuk that is based on rate control. The Wiimote promises faster target acquisition times and easiness of learning compared to isometric and elastic rate control devices such as the Nunchuk.

For interaction up close, a interaction technique based on hand postures was designed and implemented. This technique works for a broad range of users and allows them to directly access a set of four functions. More functions can be accessed by providing different context through the Smartboard pen tray. If a single user runs a posture calibration before interacting with an application, up to seven hand postures can be used.

Interacting at a distance with TSLDs has advantages such as gaining an overview of the whole display. Interaction up close also has its strengths, e. g., the possibility to precisely steer even small objects on the screen. Two bimanual interaction techniques have been developed that allow users to interact at the distance they seem appropriate. With both techniques users choose an action with one hand by using the Nunchuk joystick or its buttons. The other hand is used to indicate the location where the action will be carried out. This way, the bimanual interaction techniques provide a consistent user interface for accessing system functions and, therefore, support a seamless transition of interacting at a distance and up close.

User-sketched strokes have been developed to extend the set of expressive tools that users can employ when creating non-photorealistic renditions. A stroke data structure was employed that enables users to sketch a variety of 2D shapes, to modify and erase them. When users have finished to sketch a stroke, this stroke can be used as a regular rendering primitive.

In order to better match users expectations when interacting with these rendering primitives, the existing buffer approach was modified; instantaneous buffers replace persistent buffers. Rendering primitives can now be moved across the Interactive Canvas without changing their properties, such as size or orientation.

5 EVALUATION OF BIMANUAL INTERACTION TECHNIQES

This chapter describes the evaluation of the bimanual interaction techniques for interacting at a distance and interacting up close (as described in [Grubert et al., 2007b]).

5.1 INTRODUCTION

During the design of the bimanual interaction techniques it was not clear which technique, direct-touch or remote pointing, people would prefer to develop their digital paintings with. Both techniques provide a mapping to the available functionality and both have unique advantages. While direct-touch interaction allows users to work closer to the painting and, therefore, supports a more direct approach to painting than remote pointing, remote pointing allows users to sit comfortably at a distance and to gain a better overview of the entire painting than it is possible with direct-touch.

Therefore, a user study was performed to evaluate both interaction techniques for creating NPR paintings. The performance of the techniques was evaluated when users were doing simple tasks. In addition, the ways how people would use the interaction techniques when creating an entire painting was investigated.

5.1.1 *Participants*

Sixteen paid participants from the area of a local university (seven male, nine female) were recruited. Ages ranged from 21 to 66 years ($M = 30.3$ years, $Mdn = 28$ years, $SD = 10.2$ years). Five had used a Wiimote before, all were right-handed (with one claiming to have converted from ambidextrous at birth), and seven had an artistic background.

5.1.2 *Apparatus*

Participants performed the experiment at a plasma wall display with a resolution of 1360 × 768 pixels and a display area of 135 cm × 76 cm, mounted so that its bottom was 106 cm off of the ground. Direct-touch input was provided through SmartBoard DViT technology and remote pointing through a Wiimote and Nunchuk. For the direct-touch condition, participants were asked to stand directly in front of the display with the Nunchuk in one hand. For the remote pointing condition, participants were asked to sit in a chair that was 46 cm high and placed 165 cm in front of the display (eye-to-display distance approx. 195 cm), with the Nunchuk in one hand and the Wiimote in the other. The infrared LED markers used to detect the Wiimote's position were placed at the bottom of the screen.

5.1.3 *Procedure & Design*

The user study consisted of two phases. In the first phase, performance of each technique was measured as the participants performed controlled painting tasks. In the second phase, participants' behaviour when they were given the freedom to choose how to interact was observed.

Phase I: Controlled Tasks

In the first phase of the experiment, participants were asked to perform the following four tasks:

- create strokes (create),
- align strokes horizontally (align),
- orient strokes in a star shape (star), and
- repel strokes in a circular pattern (repel).

These four tasks can be invoked with the Nunchuk joystick using the center rest position (create), the left position (align), the bottom-left position (star), and the top-right position (repel). While holding the correct position, the participant then touched the display (in the direct-touch condition) or pointed with the Wiimote and pressed the B button (in the remote-pointing condition) to invoke the action.

Each participant was asked to perform five blocks of trials for each of the two techniques. Each block consisted of 20 trials (5 repetitions of each task) for a total of 200 trials (2 techniques $\times$ 5 blocks $\times$ 20 trials). For each technique, participants began with a practice block of 8 trials (2 trials per task) and were reminded that they could take a break after each block. For each trial, the instruction (i. e., which task to perform) was displayed at the top of the screen and a target area was displayed in the center. The participant was asked to perform the described task inside the target area as quickly as they could, but to affect the area outside the boundary as little as possible.

For the create and align tasks, the target area was a long horizontal oval (Figure 23 and Figure 24) and for the star and repel tasks, the target area was a circle (Figure 25). For each trial, distractor strokes were distributed randomly outside the target area. For the align, star and repel tasks, a more dense concentration of strokes was distributed in an area with double the height of the target area (centred at the same location), providing a dense set of strokes on which to perform the task. The participant indicated that they were finished each trial by pressing the Z button on the Nunchuk, which also started the next trial in the block.

The area affected by a touch in the direct-touch condition and by the Wiimote in the remote pointing condition was a circle the same height

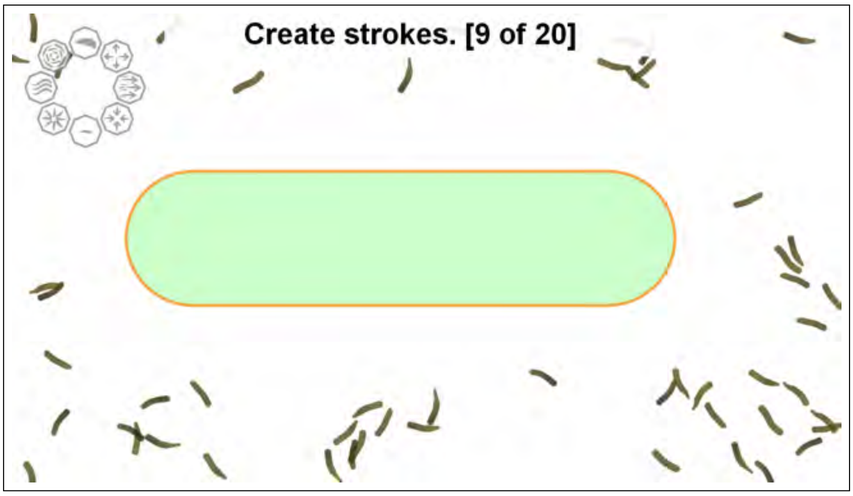


Figure 23: Sample screen for the create task. The instructions and the number of the current task and the total number of tasks is displayed on top of the screen. A green oval denotes the target area. Distractor strokes are distributed across the canvas. The tool menu is in its default position in the top left corner of the screen.

as the target area. Thus, the ideal movement was to draw a straight line in the create and align tasks and to acquire and dwell on a target at the center of the circle in the star and repel tasks.

Phase II: Painting

In addition to the tasks from Phase I of the experiment, the participant was introduced to the following additional functionality. They could also:

- orient strokes radially,
- move strokes inward—like a black hole,
- move strokes along a line,
- make strokes larger,
- make strokes smaller,
- adjust the size of the area of influence,
- alter the stroke type,
- adjust the rate at which strokes were created or erased, and
- erase strokes.

Participants were shown four examples of paintings created with the system along with the template images used to create them. They were then asked to create their own painting, based on one of four photographs (Figure 26), using the provided tools. This photograph

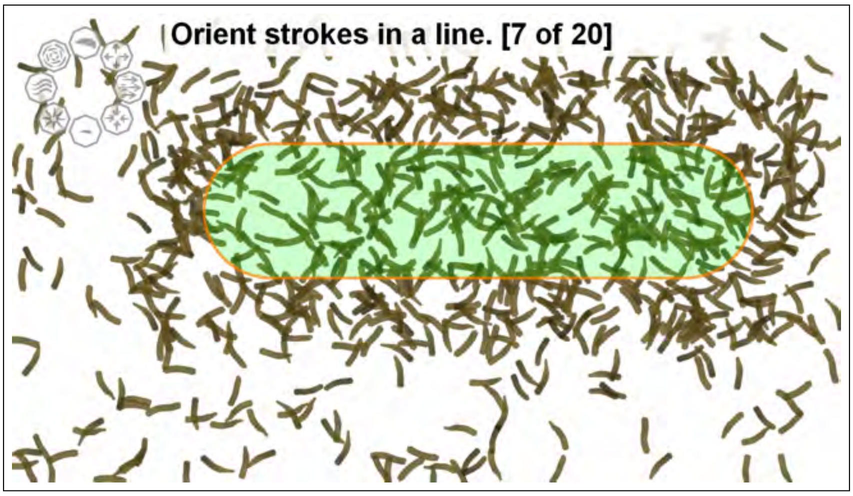


Figure 24: Sample screen for the orient align task.

was used to automatically determine the colour of each stroke based on its position in the canvas; participants were thus only required to alter the number, size, orientation, position, and type of the strokes.

5.1.4 *Hypotheses & Focus*

The first phase of the experiment was designed as a hypothesis test to compare direct-touch to remote pointing, specifically in the context of the tasks. How participants learned to use the devices over time was also of interest. The usual null hypotheses was associated with the factorial design.

The second phase of the experiment was designed to provide the opportunity to observe the system in use. Focus was on the following aspects of this interaction:

- the choice of interaction technique,
- whether participants would switch between interaction techniques,
- what tools the participants would choose to use,
- whether participants would rate certain aspects of the system particularly enjoyable or frustrating, and
- whether participants would enjoy working with the system in general.

5.1.5 *Data Collection*

Participants were videotaped and the experiment was followed by an informal interview in which they were asked to comment on ease of use, problems encountered, and overall opinions for each of the techniques. Timing data and input device coordinates (for both direct-

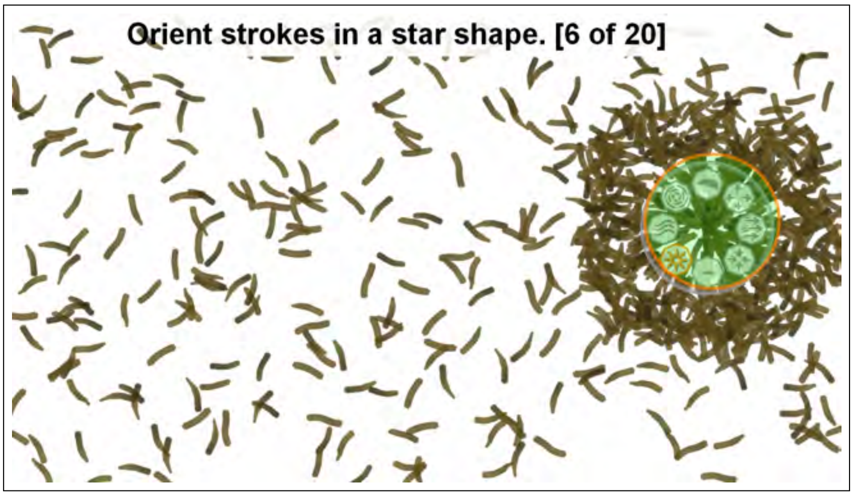


Figure 25: Sample screen for the orient radial task. The tool menu with selected orient radial icon is positioned in the circle target area as the user performs the action.

touch and the Wiimote) as well as the final size, orientation, and position of each stroke were logged.

5.2 RESULTS & DISCUSSION

In the first phase of the experiment, performance was of primary interest, so that in the second phase of the experiment the focus could be on observing behavior. Thus, results for speed and accuracy are only presented for the first phase. Data were analyzed using a within-participants analysis of variance for the following three factors:

- block (1–5),
- task (create, align, star, repel), and
- device (direct-touch, remote pointing).

Significance values for all post-hoc pairwise comparisons were adjusted using the Bonferroni correction.

5.2.1 *Speed*

Task completion times (TCT) for each trial were analyzed. TCT includes several components including: the time to read the instruction, the time to react, the time to select the appropriate action with the Nunchuk, the time to acquire the target, and the time to perform the action. Separately the time to perform the action was analyzed, but report only results for TCTs are reported, as the effects, interactions and mean differences were similar.

The main effects and interactions are summarized in Table 2. The main effect of *device* shows that participants were significantly faster

Figure 26: Template images, one of which each participant was asked to interpret in the painting phase.

Factor	F-score	p-value
device	$F(1, 13) = 8.7$	$p = .01$
block	$F(4, 52) = 46.3$	$p < .001$
task	$F(3, 39) = 5.8$	$p < .01$
device × block	$F(4, 52) = 0.6$	$p = .66$
device × **task**	$F(3, 39) = 5.2$	$p < .01$
block × task	$F(12, 156) = 1.4$	$p = .16$
device × **block** × **task**	$F(12, 156) = 2.4$	$p < .01$

Table 2: ANOVA results for task completion times.

with direct-touch ($M = 4.20$ s, $SE = 0.33$ s) than with remote pointing ($M = 5.32$ s, $SE = 0.49$ s). The main effect of *block* reflects an expected learning effect; pairwise comparisons showed that participants were significantly slower in block one than all future blocks ($p < .05$), and significantly slower in block two than blocks three and five ($p < .05$), but that differences between blocks three to five were not significant ($p > .05$). For the main effect of *task*, post-hoc tests showed that participants were significantly slower in the align task than in the star ($p < .01$) and repel ($p < .01$) tasks, but no other pair of tasks were significantly different ($p > .05$). It could be that the align task was slower because, although quick movement were observed for this task, participants sometimes needed to "correct" the result with a second or third pass.

The interaction between *device and task* (see Figure 27(a)) shows that the difference in performance for the align task depends on which device the participant used. That is, for remote pointing, the align task was significantly slower than the star ($p < .01$) and repel ($p = .01$) tasks and no other pairs were different (similar to the main effect of task), but no task pairs were different for the directtouch condition

Task	Block				
	1	2	3	4	5
create	p = .19	**p < .01**	**p < .01**	**p < .01**	**p < .001**
align	p = .06	p = .05	**p < .01**	**p = .03**	**p < .01**
star	p = .20	**p = .02**	p = .17	p = .09	p = .62
repel	**p < .01**	**p = .02**	p = .49	**p < .01**	p = .35

Table 3: Pairwise significant differences between devices.

($p > .05$). The *three-way interaction* further illustrates these differences (see Figure 28). In addition, Table 3

shows the pairwise significant differences between devices for each task and each block. All mean differences show that direct-touch was faster than remote pointing. These differences suggest that, for tasks requiring movement along a line (create and align), the improvement over time was greater for direct-touch, but for tasks requiring only pointing and dwelling (star and repel), the improvement over time was greater for remote pointing. Note also in the latter case that the remote pointing improved to be not significantly different than direct-touch by the final block.

5.2.2 *Accuracy*

Two measures of were analyzed accuracy: average distance (D_{avg}) and coordination (C). The average distance is a measure of how closely the participant matched the optimal trajectory. The coordination measure is the same as that used by Zhai and Milgram to measure coordination in six degree of freedom input devices [Zhai and Milgram, 1998], but cannot be calculated for the *star* and *repel* tasks, since the optimal path length is zero. For both of these measures, a lower value indicates

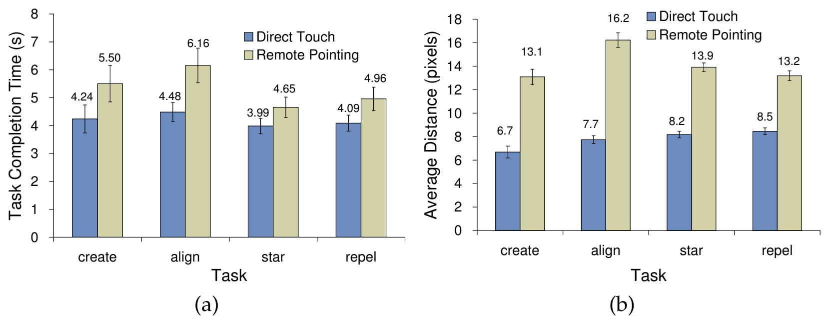


Figure 27: Interaction between device and task (a). Average distances by task for both devices (b).

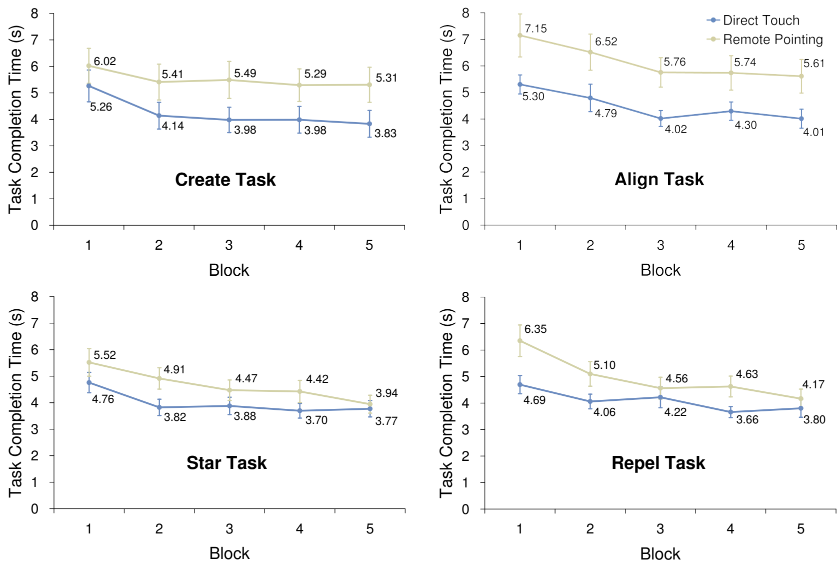


Figure 28: Three-way interaction between device, block, and task.

higher accuracy with either the participant's finger or the Wiimote These two measures are defined as follows:

$$D_{avg} = \frac{1}{|P|} \sum_{p \in P} \text{distance}(p, L_{opt})$$

$$C = \frac{\text{length}(P) - \text{length}(L_{opt})}{\text{length}(L_{opt})}$$

Where P is the set of points defining the path traversed during a trial (the touched points in the direct-touch condition and the points traversed while holding the B button in the indirect condition) and L_{opt} is the optimal path for a trial (a line in the create and align tasks, a single point in the star and repel tasks).

Distance

There was a significant main effect of *device* ($F(1, 14) = 172.3$, $p < .001$). Participants activated points closer to the optimal path with the direct-touch technique ($M = 7.8$ pixels, $SE = 0.4$ pixels) than with the remote pointing technique ($M = 14.1$ pixels, $SE = 0.6$ pixels). There was also a significant interaction between *device and task* ($F(3, 42) = 3.9$, $p = .01$). Post-hoc comparisons showed that, for the direct-touch condition, the average distance to the optimal line was significantly closer for the create task than for the align task ($p = .01$), but no other pair of tasks was significantly different for either technique ($p > .05$). This isolated difference may be due to the fact that the create task invited more precision than the align task (which both required movement along a line), which was only achievable using the direct-touch device (see Figure 27(b)). There were no other main effects or interactions ($p > .05$).

Coordination

There was a significant main effect of *device* ($F(1, 14) = 8.6$, $p = .01$). Participants were more coordinated when using direct-touch ($M = 0.25$, $SE = 0.02$) than when using remote pointing ($M = 0.51$, $SE = 0.11$). There was also a significant main effect of *task* ($F(1, 14) = 14.8$, $p < .01$) as participants were more coordinated in the create task ($M = 0.24$, $SE = 0.03$) than in the align task ($M = 0.53$, $SE = 0.10$). There was also a significant interaction between *device and task* ($F(1, 14) = 8.6$, $p = .01$). Post-hoc analysis showed that, in the align task, participants were significantly more coordinated with direct-touch than with remote pointing ($p = .01$), but in the create task, this difference was not significant ($p = .16$). There were no other significant main effects or interactions ($p > .05$).

With the lack of significant differences involving the block factor, it appears that coordination is not affected by learning or fatigue (within the hour-long time frame of our study). The results also suggest that coordination difficulties with remote pointing depend on the task. That is, in the align task, participants were less coordinated with remote pointing, but not so in the create task.

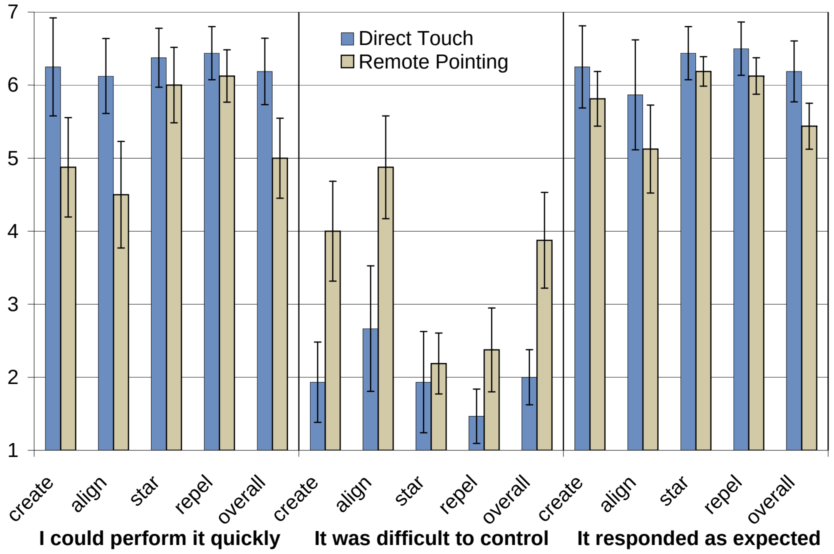


Figure 29: Participant responses for speed, control, and expectation (1 = strongly disagree, 7 = strongly agree).

5.2.3 *Questionnaire Data*

A seven-point Likert scale [Lickert, 1932] for questions about speed, control, and expectation (Figure 29) was used. Participants were also asked to state which device they preferred (Figure 30) for each task (if any). Participants agreed that both devices were fast and responded as expected. This agreement was slightly stronger in the create and align tasks and overall for direct-touch. Participants disagreed with direct-touch being difficult to control in all tasks. For remote pointing, they disagreed with this statement for the star and repel tasks, but agreed for the align task and were neutral for the create task and overall. Participants showed a clear preference for direct-touch, particularly for the create and align tasks. Note that participants were asked specifically about both speed and control, but often commented that they preferred remote pointing for other reasons.

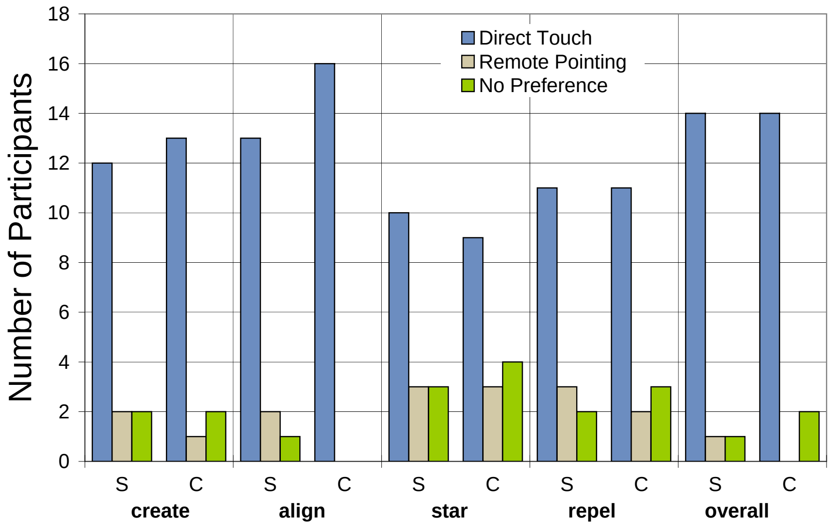


Figure 30: Participant preferences for speed and control.

5.3 OVERALL DISCUSSION

Consistent with previous findings, e. g., [Dan R. Olsen and Nielsen, 2001; Peck, 2001; Myers et al., 2002; Parker et al., 2005], the results suggest that direct-touch is faster and more accurate than remote pointing. These results alone suggest that direct-touch is a better design choice than remote pointing; however, observations during the study point to a variety of other factors that may make remote pointing or a combination of both a better choice in practice. The importance of these other factors is reflected by the fact that, in the second phase, only seven participants chose to use direct-touch, while four chose to use remote pointing, and five switched between the two.

5.3.1 *Display Distance*

The distance to the display played a large role in participants' preferences, as well as in their decisions about which device to use in Phase II of the study. Because direct-touch requires a person to stand at close proximity to a large display, it can be difficult to obtain an overview of the entire canvas. For example, when using direct-touch in Phase I of the study, some participants reported that the lack of overview made it difficult to complete the task. One participant reported that standing in front of the display felt like "standing too close to a computer screen" and also reported that he was feeling heat emitted by the plasma display.

Participants used several strategies to deal with this lack of overview. Many people stepped back and forth between trials to transfer from reading instructions to interacting with the display. Other people stood still and used wide head movements to read instructions. In Phase II, participants continued to use broad head movements when using direct-touch to look up a desired operation on the menu in the screen's top left corner (This default location was chosen to prevent people from "losing" the menu after stepping back and shifting their gaze, as it was done for remote pointing with the Wiimote as describe in Section 4.2.1). In contrast, in the remote-pointing condition, people were able to see the whole display without moving their head. Several participants reported this as a major benefit over the direct-touch technique. The proximity to the display also introduces the difficulty of reach. Many participants had to make sideways steps to reach remote areas of the screen in both phases of the study. When participants sat and used remote pointing, their movement was typically constrained to small arm and wrist movements.

5.3.2 *Control Space*

Both interaction techniques are also characterized by differences in their control spaces. While broad arm movements covering the entirety of the wall display were common for direct-touch interaction, small movements of the Wiimote achieved similar results. The broad movements of the direct-touch interaction paired with the direct physical contact allowed participants to achieve high precision in their actions and good control over the changes they applied. They reported that it "feels like a physical connection" and is "more accurate" and provides "more control." However, participants also mentioned that their arm got tired after a while due to the repeated arm movements that they used for accomplishing, in particular, the create and align tasks. In contrast, participants used small movements in the remote pointing condition. These fairly small movements of the Wiimote resulted in big actions on the screen and, thus, induced larger errors in pointing and dragging/moving. Participants who started Phase I with remote pointing reported a noticeable gain in precision and control when they switched to direct-touch, especially during the create and align tasks. Some noted that the direct technique "felt more accurate." On the other hand, participants also reported that the Wiimote "becomes extension of your hand" after a while and feels like an "extension of your body" or even that the remote pointing feels like playing a video game.

5.3.3 *Forgiving Imprecision*

No main effects or interactions involving the block factor in either average distance nor coordination were observed. However, a speed improvement over time occurred. These results suggest that participants tended to sacrifice accuracy for speed. Behavior that was consistent with these results was observed. For example, many participants seemed to be less and less careful to keep their actions constrained to the target area as the blocks progressed. Some participants would blatantly choose to not completely fill the target area or ignore when their movement line was not straight, despite the initial instruction to stay within the target boundary.

This behavior may be partly due to the fact that the painting application is very tolerant of inaccuracy. The application is forgiving both on the small scale and on the large. For example, when creating strokes, the exact location of each stroke is constrained to be within the area of influence, but the strokes are randomly distributed at each time step. Also, for any of the actions provided, an action by the user will affect any stroke whose center is in the area of influence, and so parts of the stroke may rotate outside of this area. These small-scale inaccuracies may have encouraged participants to favor speed in the first phase. On the large scale, the application allows the creation of non-photorealistic images and, specifically, invites abstraction and expressiveness (for some example results from Phase II see Figure 31). Small errors or inaccuracies are, therefore, not noticeable or even desired as part of the artistic exploration process or as part of the intended effect. Alternatively, errors can also be corrected easily and without penalty by erasing or painting over previous strokes. Consequently, in the second phase, people tended to be satisfied with a final image that reflected their intended or unintended abstraction and expressiveness. Against an initial hypothesis many participants chose to use remote pointing in the second phase, despite its obvious performance drawbacks. However, because the application was forgiving, participants may have recognized that they could sacrifice accuracy to achieve speeds close to those of direct-touch, and therefore leverage some of remote pointing's other benefits. Some participants also commented that using the Wiimote was more fun.

5.3.4 *Handedness*

An initial assumption was that people would prefer to use the Nunchuk in their non-dominant hand and to touch or point with their dominant hand. Previous research has shown that the non-dominant hand is best suited to both actions that do not require precise movement and actions that use small motions with the thumb or wrist [Kabbash et al., 1993]. Because the Nunchuk interaction primarily required only a ballistic movement to select one of the eight corners on the joystick, and this motion was activated with the thumb, this mapping is consistent with this literature. However, in the direct-touch condition, seven par-

Figure 31: Four example results that participants created in Phase II within approximately 15 minutes.

ticipants chose to hold the Nunchuk in their dominant (right) hand and to interact with the display with their non-dominant (left) hand. Furthermore, this did not seem to adversely affect their performance. This choice may be again explained by the forgiving nature of the application. Because the actions required by direct-touch are not precise by nature and because the device offers more control than the Wiimote, participants may have decided that the Nunchuk interaction required their dominant-hand abilities. One of the participants who chose to interact in direct-touch this way commented that he made this choice because he wanted more control of the Nunchuk menu.

5.4 SUMMARY OF THE EVALUATION

Findings can be summarized as follows:

- Direct-touch was shown to be faster and more precise than remote pointing.
- With remote pointing, people are able to achieve speeds similar to direct-touch by sacrificing accuracy.
- Applications that are tolerant of imprecision or invite exploration may alleviate some of the disadvantages of otherwise less efficient interaction methods.
- People had mixed preferences for remote pointing and direct-touch. In general, no correlation between preference and performance was observed. Some preferred direct-touch for its improved performance, but some preferred remote pointing for the ability to have an overview and for less fatiguing movement. Others preferred to switch between the two techniques to achieve different levels of precision and control at different times.

Both bimanual interaction techniques were shown to be suitable for the Interactive Canvas. This redundancy allows people to choose the appropriate tool for the appropriate task and to switch between them in a seamless manner. For example, when creating strokes to fill the canvas, a person can sit and view the entire screen at once and avoid the need to reach across the entire display, but when controlled motion is required to (e. g.) align strokes, a person can stand and interact directly with the canvas.

In general, the bimanual interaction techniques are a step toward providing more freedom to create and interact with non-photorealistic rendering.

6 SUMMARY AND FUTURE WORK

The goal of this thesis is to allow a broad range of users to intuitively create stroke-based non-photorealistic renditions. The Interactive Canvas [Schwarz et al., 2007] was a basis for achieving this goal, as it allows users to interactively steer the non-photorealistic rendering process.

Starting from user comments about the initial Interactive Canvas system, an informal observational study was performed to identify key strengths and drawbacks of its user interface. Based on the study results, techniques for interacting with stroke-based non-photorealistic renditions on TSLDs, were iteratively designed and implemented.

In particular, an interaction technique based on hand postures was combined with the possibility for users to sketch their own stroke primitives. This combination allows users to interact directly and expressively with stroke-based NPR on TSLDs. An example picture is shown in Figure 32.

Furthermore, two bimanual interaction techniques for direct-touch and remote pointing were designed and implemented. They provide a consistent user interface and allow users to both choose the interaction distance at will and transition between different distances seamlessly.

A user study was conducted to compare the performance and user preference of these two interaction techniques. The study revealed that even though direct-touch was superior in terms of accuracy and performance, the majority of users did not favor one technique over the other, and some frequently switched between the techniques. This suggests that the combination of both techniques supports a seamless transition between interaction at a distance and up close.

In addition, the existing buffer approach was modified (instantaneous buffers instead of persistent buffers were employed). Rendering primitives can now be moved across the Interactive Canvas without changing their properties, such as size or orientation. Informal test suggested that users anticipate this new behavior of rendering primitives.

While several interaction techniques have been developed during this project, none of them provides access to a large number of functions. To address this issue, the functions present in the initial system have been prioritized, so that the most frequently used ones can be accessed easily with the newly developed interaction techniques. However, it remains an open question how to enable access to a large number of functions and to avoid a complex interface at the same time. Providing easy access to color mixing seems to be especially urgent as this significantly extends the users's possibilities to create expressive paintings. The color mixing, that was present in the initial Interactive

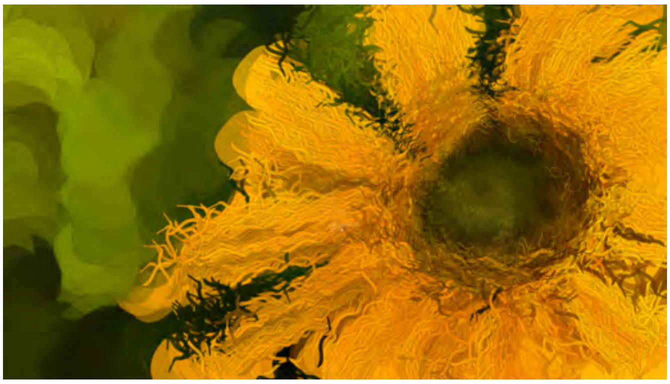


Figure 32: An example picture created with hand postures and user-sketched strokes.

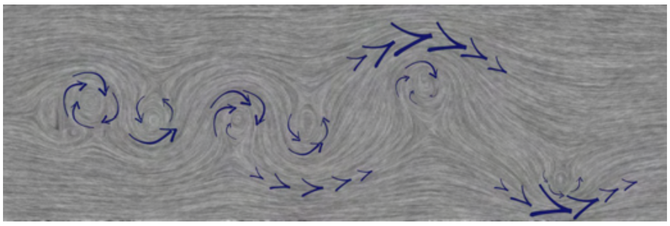
Canvas system, cannot be accessed with the techniques presented in this thesis.

To allow expressive interaction at a distance that goes beyond mere pointing and selecting, Wiimote gestures could be employed. One challenge in this context is the detection of complex gestures. Also, it is not clear how to enable a seamless transition between expressive interaction at a distance and up close using, for example, Wiimote gestures and hand postures.

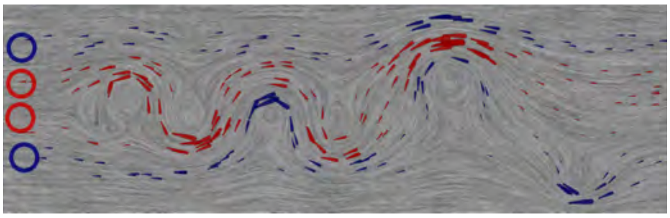
Initial tests showed that hand postures by themselves enable interaction with non-photorealistic rendition in an expressive way. However, a formal user study still should be conducted to proof this.

Figure 33: A user erases glyphs inside a flow visualization.

In addition, hand postures also could be investigated for other applications which only require a small set of functionalities. An initial example for such an application is flow visualization where hand postures can be employed for interaction with the visualization [Isenberg et al., 2008]. Figure 33 shows an example for a person interacting with



(a)



(b)

Figure 34: Annotation of a Line Integral Convolution image of a vortex simulation with three hand-sketched glyphs (a). Colored glyph sources can be used to identify trends in the vector data (b).

this application, and Figure 34 shows results produced with this application.

Speech commands could be considered as an alternative to command selection via the Nunchuk. Therefore, speech algorithms that recognize the voices of various users without the need for prior voice calibration should be employed. Algorithms that are included in the Dragon Naturally Speaking Software Development Kit (SDK)® promises this, but was unavailable during the project.

To further extend the expressiveness of interaction one could also employ dynamic primitives, that can generate geometry algorithmically, such as Graftals [Kowalski et al., 1999; Markosian et al., 2000; Kaplan et al., 2000]. Instead of adapting their level of detail to the viewing distance, these primitives could alter their appearance based on the existence or absence of neighboring primitives or based on user interaction.

APPENDIX A

A.1 TRIANGULATION

Given the distance $\overline{AB}$ between two known points (e. g., two fixed cameras or two LEDs in a sensorbar) and the angles α and β between them and a unknown third point C (e. g.a contact point on a tabletop surface or, Wiimote camera) the lengths between the known points and the unknown points, $\overline{AC}$ and $\overline{BC}$ can be computed:

$$\overline{AC} = \frac{\sin(\beta) * \overline{AB}}{180° - \alpha - \beta}.$$
$$\overline{BC} = \frac{\sin(\alpha) * \overline{AB}}{180° - \alpha - \beta}.$$

Given that length the cartesian coordinates x, y of point C can be computed:

$$x = \overline{AC} * \cos(\alpha) = \overline{BC} * \cos(\beta)$$
$$y = \overline{AC} * \sin(\alpha) = \overline{BC} * \sin(\beta).$$

Figure 35 shows the connection between the three points A, B, C and the angles α and β.

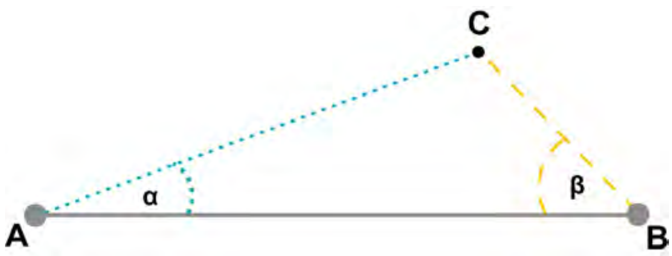
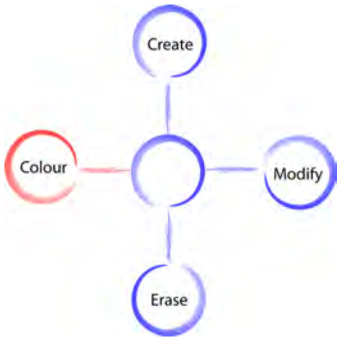


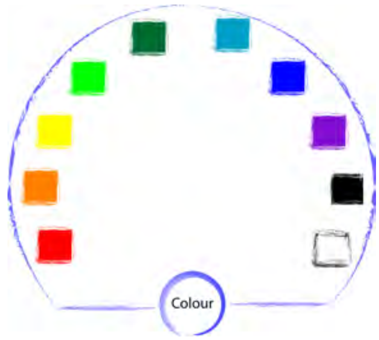
Figure 35: Coordinates of point C can be calculated through triangulation, if the distance $\overline{AB}$ and the two angles α and β are known.

A.2 GUI DESIGN

The interaction techniques described in this thesis provide direct access to frequently used functions. To access further functions a context menu, that could be invoked, e. g., by double click on or outside the canvas area, could be used. Figure 36 shows an initial design of such a context menu. Due to time constraints this menu was not fully implemented.



(a)



(b)

Figure 36: Design of a marking menu for accessing additional tools with the color option selected (a). The subsequent color selecting menu with color mixing area (b)

A.3 GESTURE RECOGNITION WITH THE NINTENDO WII REMOTE

In an early design stage capabilities of the Wiimote to detect linear acceleration in three orthogonal directions were used to implement gestures, i. e.time-dependent position and orientation changes of the Wiimote. They were used for triggering actions, such as, creating or erasing of strokes. While some gestures could be recognized quite easily, the recognition of other gestures became very challenging. In the following principals of retrieving acceleration values, some gestures and approaches for recognition of complex patterns are described.

The six DOF for a rigid body mean that it has three translation directions (x-, y-, and$z$-direction) in 3D space to specify its position, and three rotation angles (pitch, yaw, and roll) to determine its orientation. These six parameters could be used for determining static postures or time-varying gestures with the Wiimote. The Wiimote is equipped with three linear accelerometers that measure acceleration along the three pairwise orthogonal axes and that can be used to derive the instantaneous force that is imparted on the controller (see Figure 37). This force is $1g$ ($9.80665 \frac{m}{s^2}$) in upward direction when the controller is lying on a flat surface. In free fall this force is approximately zero.

However, the Wiimote reports these forces unit-less. In order to obtain values with a unit (e. g., in g), the controller first has to be calibrated. For this purpose, the controller is laid on a flat, level surface with one axis facing upwards. The g-force imposed on the controller is measured on this axis and the zero-points on the other two axes. This procedure is repeated for the other two axes.

Three pairs of x-, y-, and z-force values are obtained ((x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)). The zero points of each axis can be estimated to be:

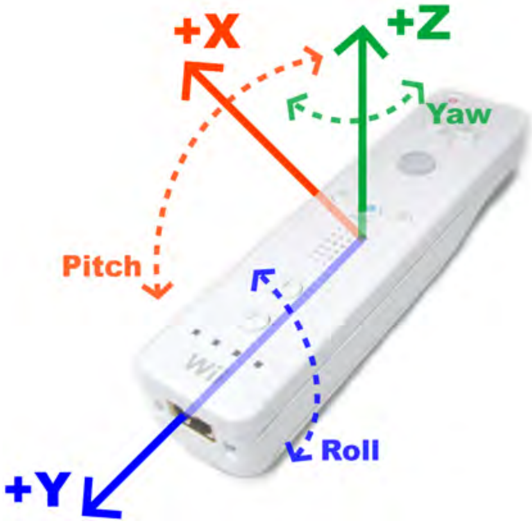


Figure 37: Coordinate system of the Wiimote.

$$x_0 = \frac{x_1 + x_2}{2}$$
$$y_0 = \frac{y_1 + y_3}{2}$$
$$z_0 = \frac{z_2 + z_3}{2}.$$

Given any unit-less integer value from the 3 sensors ($x_{raw}, y_{raw}, z_{raw}$) accelerometer values in g unit (x_g, y_g, z_g) can be computed.

$$x_g = \frac{x_{raw} - x_0}{x_3 - x_0}$$
$$y_g = \frac{y_{raw} - y_0}{y_2 - y_0}$$
$$z_g = \frac{z_{raw} - z_0}{z_1 - z_0}.$$

Roll and pitch values can be computed easily from the three raw values if they lie between $-1g$ and $1g$, i. e.when the Wiimote does not accelerate.

$$roll = asin(x_g)$$
$$pitch = asin(y_g)$$

If the Wiimote is accelerating, which is likely to be the case, the normalized acceleration values could be out of the range of $[-1g...1g]$ and, thus, the $asin$ function is no longer defined. To overcome this limitation the acceleration values have to be double integrated in order to obtain the change in tilt relative to the last known normalized value. If the controller is rotated around its z-axis no yaw value can be obtained from the accelerometer values (a gyroscope would be needed to obtain this value). The yaw values still can be obtained if the Wiimote "sees" IR LEDs in the sensor bar, in substitution of the x-position. For communicating with the Wiimote the scripting environment GlovePie [Kenner, 2007] was used. It provides convenient functions to easily obtain above mentioned acceleration values as well as estimated values for the pitch and roll, even if the controller is accelerating.

In order to recognize gestures the accelerometer and orientation values have to be analyzed for patterns. Due to time constraints only simple patterns could be analyzed, namely the ones that result in uniquely identifiable peak values. Gestures that were implemented included, e.g., shaking the controller in x-, y-, and z-direction (fast repeated movements along the axes), punching the Wiimote in any of those directions (short, non-repeated movements in one direction), "throwing" (fast movement around one axis - mainly the x-axis) and rotating back and forth around one axis (mainly the y-axis).

To detect arbitrary motion patterns, more complex motion analysis methods would have to be used. One possible approach, which is described in [Shirai et al., 2007], is to define a 3D vector space that is spanned by the magnitude acceleration values of the Wiimote. A gesture can be defined as one position in this 3D space by integrating all acceleration values that make up this gesture. However, due to the high sensitivity of the accelerometers and due to signal noise, it is hard to consistently match the position defined by previously recorded motions or even to define a approximating subspace. Fuzzy pattern matching approaches, promise to guarantee a more stable recognition as provided by AiLive [Inc., 2007], a motion analysis company. Recently an approach for using the Wiimote as a real 6 DOF controller has been proposed [Kreylos, 2007]. For that purpose, four IR LEDs are arranged in a non-planar configuration, for example in a tetrahedral shape. Because the Wiimote can detect up to four IR LEDs with their approximate radius this LED configuration can be used to solve a set of non-linear equations for the position and orientation of the Wiimote relative to it.

A.4 COMBINING SPEECH WITH DIRECT AND INDIRECT INTERACTION

Research showed, that speech lends itself well for issuing commands [Bolt, 1980]. Many approaches have shown that speech commands can complement interaction techniques for indicating location well [Tse, 2007].

This motivated the investigation of speech commands for selecting tools and switching modes. Similar to the bimanual interaction techniques, described in Section 4.3, the screen area where the action should be carried out remains to be indicated.

There were several challenges during the development of voice commands, that are meaningful, easy to remember and easy to recognize. As with other techniques that do not provide feedback about current selected modes, users have to actively remember this modes. Due to inaccuracies in the voice recognition process lengthy command names had to be used to ensure a proper recognition. For example, the speech commands "create strokes" and "delete strokes" could not be distinguished well and had to be extended, e. g., to "create new strokes". Furthermore, to provide decent voice recognition high-quality cable microphones had to be used. While bluetooth microphones would allow to interact at a distance and up close, their audio transmission quality was not suitable for ensuring consistent speech recognition. While testing speech commands with different users, it became clear that this approach was not suitable if many different users should be able to interact via speech. This is due to a required and lengthy speech profiling process that is necessary to properly recognize speech in the employed Microsoft Speech SDK®. The Dragon Naturally Speaking SDK®would allow speech recognition without the need for prior voice calibration but was unavailable during the project. These constraints led to the termination of a speech-based interaction technique during this project.

BIBLIOGRAPHY

Maneesh Agrawala, Andrew C. Beers, Ian McDowall, Bernd Fröhlich, Mark Bolas, and Pat Hanrahan. The two-user responsive workbench: support for collaboration through individual views of a shared space. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 327–332, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. ISBN 0-89791-896-7. doi: http://doi.acm.org/10.1145/258734.258875. (Cited on page 11.)

Bill Baxter, Vincent Scheib, Ming C. Lin, and Dinesh Manocha. DAB: Interactive haptic painting with 3D virtual brushes. In *Proc. SIGGRAPH*, pages 461–468, New York, 2001. ACM Press. doi: http://doi.acm.org/10.1145/383259.383313. (Cited on pages 5 and 6.)

Richard 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. ISBN 0-89791-021-4. doi: http://doi.acm.org/10.1145/800250.807503. (Cited on pages 14 and 62.)

Richard A. Bolt. Gaze-orchestrated dynamic windows. In *SIGGRAPH '81: Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, pages 109–119, New York, NY, USA, 1981. ACM. ISBN 0-89791-045-1. doi: http://doi.acm.org/10.1145/800224.806796. (Cited on page 13.)

William Buxton and Brad A. Myers. A study in two-handed input. In *Proc. CHI*, pages 321–326, New York, 1986. ACM Press. doi: http://doi.acm.org/10.1145/22627.22390. (Cited on page 14.)

Jan Ciger, Mario Gutierrez, Frederic Vexo, and Daniel Thalmann. The magic wand. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics*, pages 119–124, New York, NY, USA, 2003. ACM. ISBN 1-58113-861-X. doi: http://doi.acm.org/10.1145/984952.984972. (Cited on page 12.)

P. Cohen. Speech can't do everything: A case for multimodal systems. *Speech Technology Magazine*, 5(4), 2000. (Cited on page 14.)

Lawrence D. Cutler, Bernd Fröhlich, and Pat Hanrahan. Two-handed direct manipulation on the responsive workbench. In *SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 107–114, New York, NY, USA, 1997. ACM. ISBN 0-89791-884-3. doi: http://doi.acm.org/10.1145/253284.253315. (Cited on page 6.)

Mary Czerwinski, George Robertson, Brian Meyers, Greg Smith, Daniel Robbins, and Desney Tan. Large display research overview. In *CHI '06: CHI '06 extended abstracts on Human factors in computing* *systems*, pages 69–74, New York, NY, USA, 2006. ACM. ISBN 1-59593-298-4. doi: http://doi.acm.org/10.1145/1125451.1125471. (Cited on pages 1 and 11.)

Jr. Dan R. Olsen and Travis Nielsen. Laser pointer interaction. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 17–22, New York, NY, USA, 2001. ACM. ISBN 1-58113-327-8. doi: http://doi.acm.org/10.1145/365024.365030. (Cited on pages 12 and 50.)

Oliver Deussen, Stefan Hiller, Cornelius van Overveld, and Thomas Strothotte. Floating points: A method for computing stipple drawings. In *Computer Graphics Forum*, volume 19, pages 40–51, August 2000. (Cited on page 1.)

Paul Dietz and Darren Leigh. Diamondtouch: a multi-user touch technology. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226, New York, NY, USA, 2001. ACM. ISBN 1-58113-438-X. doi: http://doi.acm.org/10.1145/502348.502389. (Cited on pages 7 and 14.)

Scott Elrod, Richard Bruce, Rich Gold, David Goldberg, Frank Halasz, William Janssen, David Lee, Kim McCall, Elin Pedersen, Ken Pier, John Tang, and Brent Welch. Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 599–607, New York, NY, USA, 1992. ACM. ISBN 0-89791-513-5. doi: http://doi.acm.org/10.1145/142750.143052. (Cited on page 11.)

Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In *Proc. CHI*, pages 647–656, New York, 2007. ACM Press. doi: http://doi.acm.org/10.1145/1240624.1240726. (Cited on pages 15 and 27.)

Bruce Gooch and Amy A. Gooch. *Non-Photorealistic Rendering*. A K Peters, Ltd., Natick, 2001. doi: http://doi.acm.org/10.1145/558817. URL http://www.cs.utah.edu/~gooch/book.html. (Cited on page 7.)

Jens Grubert, Sheelagh Carpendale, and Tobias Isenberg. Interactive stroke-based npr using hand postures on large displays. Technical Report 2007-883-35, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, The University of Calgary, 2500 University Drive NW, Calgary, Alberta Canada T2N 1N4, December 2007a. URL http://pharos.cpsc.ucalgary.ca/Dienst/UI/2.0/Describe/ncstrl.ucalgary_cs/2007-883-35. (Cited on page 19.)

Jens Grubert, Mark Hanckock, Sheelagh Carpendale, Edward Tse, and Tobias Isenberg. Interacting with stroke-based rendering on a wall display. Technical Report 2007-882-34, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, The University of Calgary, 2500 University

Drive NW, Calgary, Alberta Canada T2N 1N4, October 2007b. URL http://pharos.cpsc.ucalgary.ca/Dienst/UI/2.0/Describe/ncstrl.ucalgary_cs/2007-882-34. (Cited on page 41.)

Jens Grubert, Sheelagh Carpendale, and Tobias Isenberg. Interactive stroke-based npr using hand postures on large displays. In *Short Papers at Eurographics 2008*, Aire-la-Ville, Switzerland, 2008. In Press. (Cited on page 19.)

Yves Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19(4):486–517, December 1987. URL http://cogprints.org/625/. (Cited on page 15.)

Paul Haeberli. Dynadraw. *Grafica OBSCURA*, July 1989. URL http://www.graficaobscura.com/dyna/. http://www.graficaobscura.com/dyna/. (Cited on page 35.)

Paul Haeberli. Paint by numbers: abstract image representations. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 207–214, New York, NY, USA, 1990. ACM. ISBN 0-201-50933-4. doi: http://doi.acm.org/10.1145/97879.97902. (Cited on pages xi, 8, and 9.)

Aaron Hertzmann. A survey of stroke-based rendering. *IEEE Computer Graphics and Applications*, 23(4):70–81, July/August 2003. doi: http://doi.ieeecomputersociety.org/10.1109/MCG.2003.1210867. URL http://dx.doi.org/10.1109/MCG.2003.1210867. (Cited on page 8.)

Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 453–460, New York, NY, USA, 1998. ACM. ISBN 0-89791-999-8. doi: http://doi.acm.org/10.1145/280814.280951. (Cited on pages 8 and 10.)

Aaron Hertzmann. Paint by relaxation. In *CGI '01: Computer Graphics International 2001*, pages 47–54, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1007-8. (Cited on page 8.)

Ken Hinckley, Randy Pausch, Dennis Proffitt, James Patten, and Neal Kassell. Cooperative bimanual action. In *Proc. CHI*, pages 27–34, New York, 1997. ACM Press. doi: http://doi.acm.org/10.1145/258549.258571. (Cited on page 15.)

AiLive Inc. Livemove. Website, 2007. URL http://www.ailive.net/papers/wp.pdf; visited on January 2nd 2008. (Cited on page 62.)

Tobias Isenberg, André Miede, and Sheelagh Carpendale. A buffer framework for supporting responsive interaction in information visualization interfaces. In *Proc. C^5*, pages 262–269, Los Alamitos, 2006. IEEE Computer Society. doi: http://doi.ieeecomputersociety.org/10.1109/C5.2006.4. URL http://dx.doi.org/10.1109/C5.2006.4. (Cited on page 11.)

Tobias Isenberg, Maarten Everts, Jens Grubert, and Sheelagh Carpendale. Interactive exploratory visualization of vector fields. In *Computer Graphics Forum*, volume 27, 2008. Submitted. (Cited on page 56.)

Brad Johanson, Armando Fox, and Terry Winograd. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, 1(2):67–74, 2002. ISSN 1536-1268. doi: http://dx.doi.org/10.1109/MPRV.2002.1012339. (Cited on page 11.)

Paul Kabbash, I. Scott MacKenzie, and William Buxton. Human performance using computer input devices in the preferred and non-preferred hands. In *Proc. CHI*, pages 474–481, New York, 1993. ACM Press. ISBN 0-89791-575-5. doi: http://doi.acm.org/10.1145/169059.169414. (Cited on page 52.)

Robert D. Kalnins. *WYSIWYG NPR: Interactive Stylization For Stroke-Based Rendering of 3D Animation*. PhD thesis, Department of Computer Science. Princeton University, June 2004. (Cited on pages 10 and 12.)

Robert D. Kalnins, Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Joseph C. Lee, Philip L. Davidson, Matthew Webb, John F. Hughes, and Adam Finkelstein. WYSIWYG NPR: Drawing strokes directly on 3D models. *ACM Transactions on Graphics*, 21(3):755–762, July 2002. doi: http://doi.acm.org/10.1145/566654.566648. URL `http://www.cs.princeton.edu/gfx/pubs/Kalnins_2002_WND/`. (Cited on pages 1, 10, and 20.)

Matthew Kaplan, Bruce Gooch, and Elaine Cohen. Interactive artistic rendering. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 67–74, New York, NY, USA, 2000. ACM. ISBN 1-58113-277-8. doi: http://doi.acm.org/10.1145/340916.340925. (Cited on page 57.)

Daniel F. Keefe, Daniel Acevedo Feliz, Tomer Moscovich, David H. Laidlaw, and Joseph J. LaViola Jr. CavePainting: A fully immersive 3D artistic medium and interactive experience. In *Proc. I3D*, pages 85–93, New York, 2001. ACM Press. doi: http://doi.acm.org/10.1145/364338.364370. (Cited on page 6.)

Carl Kenner. Glovepie. Website, 2007. URL `http://carl.kenner.googlepages.com/`; visited on January 2nd 2008. (Cited on page 61.)

Michael A. Kowalski, Lee Markosian, J. D. Northrup, Lubomir Bourdev, Ronen Barzel, Loring S. Holden, and John F. Hughes. Art-based rendering of fur, grass, and trees. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 433–438, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-48560-5. doi: http://doi.acm.org/10.1145/311535.311607. (Cited on page 57.)

Oliver Kreylos. Irbeacon. Website, 2007. URL http://idav.ucdavis.edu/~okreylos/ResDev/Wiimote/index.html; visited on January 2nd 2008. (Cited on page 62.)

Russell Kruger, M. Sheelagh T. Carpendale, Stacey D. Scott, and Anthony Tang. Fluid integration of rotation and translation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2005, Portland, Oregon)*, pages 601–610. ACM Press, 2005. doi: http://doi.acm.org/10.1145/1054972.1055055. (Cited on page 14.)

Dustin Lang, Leih Findlater, and Michael Shaver. CoolPaint: Direct interaction painting. In *Poster Proc. UIST*, 2003. URL http://www.cs.toronto.edu/~dstn/papers/coolpaint-abstract.pdf. (Cited on page 5.)

Sangwon Lee, Sven C. Olsen, and Bruce Gooch. Interactive 3D fluid jet painting. In *Proc. NPAR*, pages 97–104, New York, 2006. ACM Press. doi: http://doi.acm.org/10.1145/1124728.1124745. URL http://www.cs.northwestern.edu/~sle841/papers/npr2006/. (Cited on pages xi and 7.)

Rensis Lickert. A technique for the measurement of attitudes. *Archives of Psychology*, 140:1–55, 1932. (Cited on page 49.)

I. Scott MacKenzie and Shaidah 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. ISBN 3-540-43044-X. (Cited on page 13.)

Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Loring S. Holden, J. D. Northrup, and John F. Hughes. Art-based rendering with continuous levels of detail. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 59–66, New York, NY, USA, 2000. ACM. ISBN 1-58113-277-8. doi: http://doi.acm.org/10.1145/340916.340924. (Cited on page 57.)

K. Mason and M. S. T. Carpendale. Artist-driven expressive graphics. In *Proceedings of Eurographics: Short Presentations*, pages 87–96, 2001a. (Cited on page 9.)

K. Mason and M.S.T. Carpendale. Expanding the expressive palette. In *Proceedings of the Western Computer Graphics Symposium*, 2001b. (Cited on page 9.)

Katherine M. Mason. *A Framework for Element-Based Computer Graphics*. PhD thesis, University of Calgary, Canada, 2006. (Cited on pages 9 and 10.)

Kaye Mason, Jörg Denzinger, and Sheelagh Carpendale. Negotiating gestalt: Artistic expression by coalition formation between agents. In *Proceedings of the 5th International Symposium on Smart Graphics (SG 2005, August 22-24, 2005, Frauenwörth Cloister, Germany)*, pages

103–114. Springer-Verlag, 2005. doi: http://dx.doi.org/10.1007/11536482_9. (Cited on page 9.)

Deborah J. Mayhew. *The usability engineering lifecycle: a practitioner's handbook for user interface design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. ISBN 1-55860-561-4. (Cited on page 17.)

André Miede. Realizing responsive interaction for tabletop interaction metaphors. Master's thesis, Otto-von-Guericke-Universität Magdeburg, 2006. (Cited on pages 11 and 37.)

Brad A. Myers, Rishi Bhatnagar, Jeffrey Nichols, Choon Hong Peck, Dave Kong, Robert Miller, and A. Chris Long. Interacting at a distance: Measuring the performance of laser pointers and other devices. In *Proc. CHI*, pages 33–40, New York, 2002. ACM Press. doi: http://doi.acm.org/10.1145/503376.503383. (Cited on pages 12, 13, 22, and 50.)

Miguel A. Nacenta, Dzmitry Aliakseyeu, Sriram Subramanian, and Carl Gutwin. A comparison of techniques for multi-display reaching. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 371–380, New York, NY, USA, 2005. ACM. ISBN 1-58113-998-5. doi: http://doi.acm.org/10.1145/1054972.1055024. (Cited on page 14.)

Miguel A. Nacenta, David Pinelle, Dane Stuckel, and Carl Gutwin. The effects of interaction technique on coordination in tabletop groupware. In *GI '07: Proceedings of Graphics Interface 2007*, pages 191–198, New York, NY, USA, 2007. ACM. ISBN 978-1-56881-337-0. doi: http://doi.acm.org/10.1145/1268517.1268550. (Cited on page 14.)

Kai Nickel and Rainer 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. ISBN 1-58113-621-8. doi: http://doi.acm.org/10.1145/958432.958460. (Cited on page 13.)

J. Karen Parker, Regan L. Mandryk, and Kori M. Inkpen. TractorBeam: Seamless integration of local and remote pointing for tabletop displays. In *Proc. GI*, pages 33–40, Mississauga, ON, Canada, 2005. CHCCS. ISBN 1-56881-265-5. doi: http://doi.acm.org/10.1145/1089508.1089515. (Cited on pages 12, 15, and 50.)

Choon Hong Peck. Useful parameters for the design of laser pointer interaction techniques. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 461–462, New York, NY, USA, 2001. ACM. ISBN 1-58113-340-5. doi: http://doi.acm.org/10.1145/634067.634333. (Cited on pages 12 and 50.)

Jenny Preece, Yvonne Rogers, Helen Sharp, David Benyon, Simon Holland, and Tom Carey. *Human-Computer Interaction*. Addison-Wesley Longman Ltd., Essex, UK, UK, 1994. ISBN 0201627698. (Cited on page 17.)

Jun Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 31–39, New York, NY, USA, 1997. ACM. ISBN 0-89791-881-9. doi: http://doi.acm.org/10.1145/263407.263505. (Cited on page 14.)

Kimiko Ryokai, Stefan Marti, and Hiroshi Ishii. I/O Brush: Drawing with everyday objects as ink. In *Proc. CHI*, pages 303–310, New York, 2004. ACM Press. doi: http://doi.acm.org/10.1145/985692.985731. (Cited on pages 5 and 6.)

Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-d shapes. *SIGGRAPH Comput. Graph.*, 24(4):197–206, 1990. ISSN 0097-8930. doi: http://doi.acm.org/10.1145/97880.97901. (Cited on page 9.)

Steven Schkolne, Michael Pruett, and Peter Schröder. Surface drawing: Creating organic 3D shapes with the hand and tangible tools. In *Proc. CHI*, pages 261–268, New York, 2001. ACM Press. doi: http://doi.acm.org/10.1145/365024.365114. (Cited on pages 6 and 7.)

Stefan Schlechtweg, Tobias Germer, and Thomas Strothotte. Renderbots - multi-agent systems for direct image generation. *Computer Graphics Forum*, 24(2):137–148, 2005. doi: 10.1111/j.1467-8659.2005.00838.x. (Cited on pages 9, 10, and 20.)

Martin Schwarz. An interactive non-photorealistic canvas. Honors thesis, Department of Simulation and Graphics, Faculty of Computer Science, Otto-von-Guericke-University of Magdeburg, April 2007. (Cited on pages 10 and 11.)

Martin Schwarz, Tobias Isenberg, Katherine Mason, and Sheelagh Carpendale. Modeling with rendering primitives: An interactive non-photorealistic canvas. In *Proc. NPAR*, pages 15–22, New York, 2007. ACM Press. doi: http://doi.acm.org/10.1145/1274871.1274874. URL `http://innovis.cpsc.ucalgary.ca/Research/InteractiveCanvas`. (Cited on pages 1, 8, 10, 11, 12, 17, 20, 28, 37, and 55.)

Abigail J. Sellen, Gordon P. Kurtenbach, and William A. S. Buxton. The prevention of mode errors through sensory feedback. *Human-Computer Interaction*, 7(2):141–164, April 1992. doi: http://dx.doi.org/10.1207/s15327051hci0702_1. (Cited on page 33.)

Akihiko Shirai, Erik Geslin, and Simon Richir. WiiMedia: Motion analysis methods and applications using a consumer video game controller. In *Proc. Sandbox*, pages 133–140, New York, 2007. ACM Press. ISBN 978-1-59593-749-0. doi: http://doi.acm.org/10.1145/1274940.1274966. (Cited on page 62.)

Alexander W. Skaburskis, Roel Vertegaal, and Jeffrey S. Shell. Auramirror: reflections on attention. In *ETRA '04: Proceedings of the*

2004 symposium on Eye tracking research & applications, pages 101–108, New York, NY, USA, 2004. ACM. ISBN 1-58113-825-3. doi: http://doi.acm.org/10.1145/968363.968385. (Cited on page 13.)

Alvy Ray Smith. Digital paint systems: An anecdotal and historical overview. *IEEE Annals of the History of Computing*, 23(2):4–30, 2001. ISSN 1058-6180. doi: http://doi.ieeecomputersociety.org/10.1109/85.929908. (Cited on page 5.)

Thomas Strothotte and Stefan Schlechtweg. *Non-Photorealistic Computer Graphics. Modeling, Animation, and Rendering*. Morgan Kaufmann Publishers, San Francisco, 2002. doi: http://doi.acm.org/10.1145/544522. URL http://isgwww.cs.uni-magdeburg.de/pub/books/npr/. (Cited on pages 1 and 7.)

Edward Tse. *Multimodal Co-located Interaction*. PhD thesis, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, 2007. (Cited on pages 14 and 62.)

Edward Tse, Saul Greenberg, and Chia Shen. Gsi demo: multiuser gesture/speech interaction over digital tables by wrapping single user applications. In *ICMI '06: Proceedings of the 8th international conference on Multimodal interfaces*, pages 76–83, New York, NY, USA, 2006a. ACM. ISBN 1-59593-541-X. doi: http://doi.acm.org/10.1145/1180995.1181012. (Cited on page 14.)

Edward Tse, Chia Shen, Saul Greenberg, and Clifton Forlines. Multimodal multiplayer tabletop gaming. In *Proceedings Third International Workshop on Pervasive Gaming Applications (PerGames'06)*, pages 139–148, 2006b. in conjunction with 4th Intl. Conference on Pervasive Computing. (Cited on page 14.)

Edward Tse, Chia Shen, Saul Greenberg, and Clifton Forlines. Enabling interaction with single user applications through speech and gestures on a multi-user tabletop. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 336–343, New York, NY, USA, 2006c. ACM. ISBN 1-59593-353-0. doi: http://doi.acm.org/10.1145/1133265.1133336. (Cited on page 14.)

Edward Tse, Mark Hancock, and Saul 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, 2007a. ACM. ISBN 978-1-59593-817-6. doi: http://doi.acm.org/10.1145/1322192.1322245. (Cited on page 12.)

Edward Tse, Chia Shen, Saul Greenberg, and Clifton Forlines. How pairs interact over a multimodal digital table. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 215–218, New York, NY, USA, 2007b. ACM. ISBN 978-1-59593-593-9. doi: http://doi.acm.org/10.1145/1240624.1240659. (Cited on page 14.)

Edward Tse, Saul Greenberg, Chia Shen, Clifton Forlines, and R. Kodama. Exploring true multi-user multimodal interaction over a digital table. In *Proceedings of DIS '08*, pages 25–27, 2008. In Press. (Cited on page 14.)

Frederic Vernier, Neal Lesh, and Chia Shen. Visualization techniques for circular tabletop interfaces. In *Proc. AVI*, pages 257–263, 2002. (Cited on page 14.)

Daniel Vogel and Ravin Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *Proc. UIST*, pages 33–42, New York, 2005. ACM Press. doi: http://doi.acm.org/10.1145/1095034.1095041. (Cited on pages 12, 15, and 21.)

Daniel Vogel and Ravin 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, New York, NY, USA, 2004. ACM. ISBN 1-58113-957-8. doi: http://doi.acm.org/10.1145/1029632.1029656. (Cited on page 13.)

Christian von Hardenberg and François Bérard. Bare-Hand Human-Computer Interaction. In *Proc. PUI*, New York, 2001. ACM Press. doi: http://doi.acm.org/10.1145/971478.971513. Posters & demos session. (Cited on page 14.)

Mike Wu and Ravin Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multi-user 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. ISBN 1-58113-636-6. doi: http://doi.acm.org/10.1145/964696.964718. (Cited on page 14.)

Johannes Zander, Tobias Isenberg, Stefan Schlechtweg, and Thomas Strothotte. High quality hatching. *Computer Graphics Forum (Proceedings of EuroGraphics 2004)*, 23(3):421–430, September 2004. doi: 10.1111/j.1467-8659.2004.00773.x. (Cited on page 36.)

Shumin Zhai. User performance in relation to 3d input device design. *SIGGRAPH Comput. Graph.*, 32(4):50–54, 1998. ISSN 0097-8930. doi: http://doi.acm.org/10.1145/307710.307728. (Cited on pages 13, 30, and 31.)

Shumin Zhai and Paul Milgram. Quantifying coordination in multiple DOF movement and its application to evaluating 6 DOF input devices. In *Proc. CHI*, pages 320–327, New York, 1998. ACM Press. ISBN 0-201-30987-4. doi: http://doi.acm.org/10.1145/274644.274689. (Cited on page 47.)

Shumin Zhai, Carlos Morimoto, and Steven Ihde. Manual and gaze input cascaded (magic) pointing. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 246–253, New York, NY, USA, 1999. ACM. ISBN 0-201-48559-1. doi: http://doi.acm.org/10.1145/302979.303053. (Cited on page 13.)

DECLARATION

I declare that this thesis is my own unaided work, and hereby certify that unless stated, all work contained within this paper is my own. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Magdeburg, February 2008

Jens Grubert