MASTER OF SCIENCE THESIS // BY ÖMER GENÇ, 30.07.2013

NOVEL PEN & TOUCH BASED INTERACTION TECHNIQUES FOR SEISMIC INTERPRETATION

Fachhochschule Düsseldorf
University of Applied Sciences

Fraunhofer
IAIS

SUPERVISION:
PROF. DR. CHRISTIAN GEIGER
EXTERNAL SUPERVISION:
M.SC. DAVID D'ANGELO

UNIVERSITY OF APPLIED SCIENCES DÜSSELDORF
DEPARTMENT OF MEDIA

STATEMENT OF ORIGINALITY

I hereby certify that I am the sole author of this Master Thesis „Novel Pen & Touch based Interaction Techniques for Seismic Interpretation“ and that no part of this thesis has been published or submitted for a degree in any university or any other educational institution.

I declare that the intellectual content of this thesis is the product of my own work and that to the best of my knowledge it contains no materials that have previously been published or are written by another person except where due acknowledgement is made in the thesis itself.

Ömer Genç

ACKNOWLEDGEMENTS

At first, I would like to thank Prof. Dr. Christian Geiger for supervising this thesis.

Moreover, I would like to thank all my colleagues at the Fraunhofer IAIS. In particular, I thank Dr. Manfred Bogen, M.Sc. David d'Angelo, Jannik Fiedler, Phillip Ladwig and Dipl. Inform. Stefan Rilling. Especially, many thanks go to my advisor, M.Sc. David d'Angelo, for daily support, advice and guidance in pursuing this thesis.

I also want to give thanks to the readers of this Master Thesis:
I thank Christian Aldenhoff, Mirco Henze and Tim Kosse for their valuable feedback on the content of this thesis. Especially Mirco found time to read this thesis, while being under time constraints of his own Master Thesis.

Finally, I want to thank my family and friends for their never ending moral support. Foremost, I want to thank my Mother for her love and her understanding that I did not find much time to visit her during the last 12 month. In addition, thanks goes to: Aliye, Christian, Enis, Fatima, Fazli, Manuel, Mehdiye, Nazik, Semih, Sengül, Serkan, Yasar and Yasemin.

ABSTRACT

Traditional desktop based software for seismic interpretation is solely based on Windows, Icons, Menus and Pointer (WIMP) interaction. Humans have multiple fingers and highly developed motor skills to handle physical objects using both hands. The established WIMP paradigm is not powerful enough and limiting and does not leverage these acquired motor skills. In recent years, interactive displays unifying input and output with a broad diversity of form factors have emerged. These devices were designed from the ground to support novel ways of interaction establishing natural and reality-based interaction paradigms. The small Control-Display ratio of these devices provides a close and direct relation between the real world and the digital work by a spatially coincidence of direct input and direct manipulation. This work studies human computer interaction techniques which can significant benefit from applying bimanual direct interaction. The background of this investigation is the analysis and interpretation of seismic data. Novel pen and touch based interaction techniques are integrated into the mouse and keyboard dominated area of applications for seismic interpretation by utilizing users' native bimanual motor skills. The seismic interpretation workflow is perfectly suited for an evaluation of bimanual interaction research, since the fine-granularity of the analyzed data and an interactive exploration process of the huge data sets involved need dexterity, precision, and usability well suited to benefit from pen and touch characteristics respectively. As a result, cognitive benefits can arise by reducing the load of mentally composing and visualization tasks which are imposed by traditional unimanual techniques at an unnatural low level. A versatility of an asymmetric division of labor between two input modalities (pen and touch input) promises a reasonable approach to manifest an integral part of future interfaces. Therefore, the core idea is to separate the roles of the dominant hand (pen) and the non-dominant hand (touch) for an explicit determination of the input modalities. This will be evaluated by an informal user study. A literature review on recent trends of novel human computer interaction including the most relevant bimanual human motor control studies as well as the technical background of this elaboration is also part of this work.

ZUSAMMENFASSUNG

Traditionelle auf den Desktopeinsatz basierte Software für seismische Interpretation basiert ausschließlich auf Windows, Icons, Menüs und Pointer (WIMP) Interaktion. Menschen haben mehrere Finger und hoch entwickelte Fähigkeiten, um physische Objekte mit beiden Händen zu manipulieren. Das etablierte WIMP Interaktionsparadigma bietet nur beschränkte Möglichkeiten, diese natürlich erworbenen motorischen Fähigkeiten zu nutzen. In den letzten Jahren wurden interaktive Displays mit einer breiten Vielfalt an Formfaktoren entwickelt, die von Grund auf den auf Einsatz durch neue Arten der Interaktion konzipiert wurden. Das kleine Control-Display Verhältnis dieser Geräte ermöglicht es, durch eine räumliche Koinzidenz von direkter Eingabe und direkter Manipulation, eine enge und direkte Beziehung zwischen der realen Welt und der digitalen Arbeit abzuleiten. Diese Arbeit untersucht in wie weit ein Beitrag geleistet werden kann, durch den Vorteile durch die Anwendung von bi-manuellen Techniken realisiert werden können. Unter den gegebenen Voraussetzungen der bereits vorhandenen motorischen Fähigkeiten der Menschen, verspricht eine Einführung von bi-manuellen Interaktionstechniken nahtlose und effektive Arbeitsabläufe für seismische Interpretation zu realisieren. Der Hintergrund dieser Untersuchung ist die Analyse und Interpretation von seismischen Daten. Zu diesen Zweck werden neuartige Stift und Touch-basierte Interaktionstechniken in den Maus und Tastatur dominierten Bereich der Anwendungen für seismische Interpretation integriert. Seismische Interpretation ist als Forschungssubjekt sehr gut geeignet um bi-manuelle Interaktion zu evaluieren. Der Grund dafür ist, dass die feine Granularität der analysierten Daten und eine interaktive Erkundung der riesigen Datenmenge Ansprüche an Geschicklichkeit, Präzision und Benutzerfreundlichkeit stellen, die besonders geeignet sind, um jeweils von den Stift und Touch-Eigenschaften profitieren zu können. Durch Entlastung geistiger Arbeitsprozessen und Visualisierungsaufgaben, entstehen kognitive Vorteile, die durch den Einsatz von traditionellen einhändigen Techniken nicht möglich sind. Die Vielseitigkeit einer asymmetrischen Unterscheidung der Aufgaben zwischen den verschiedenen Eingabemodalitäten, verspricht hier einen begründeten Ansatz, um einen integralen Bestandteil zukünftiger Benutzerschnittstellen zu manifestieren. Der daraus resultierende Kerngedanke, sieht eine klar definierte Aufgabenteilung zwischen der dominierenden Hand (Stift) und der nicht- dominierenden Hand (Touch) vor, um eine explizite Bestimmung der Eingabemodalitäten zu gewährleisten. Dies wird durch eine informelle Benutzerstudie evaluiert. Des Weiteren, ist eine Rezension der Literatur über aktuelle Beiträge neuartiger Mensch-Computer-Interaktion ebenfalls Bestandteil dieser Arbeit. Dies beinhaltet, Forschungsstudien zur Untersuchung von menschlicher bi-manueller Interaktion, sowie den dieser Arbeit zugrundeliegenden technischen Hintergrund.

Contents

List of Figures

01 // Introduction

1 INTRODUCTION

This chapter contains an overview of the contents of this thesis. First, the motivation for the investigations of this work is provided followed by the goals of the thesis and a brief description of each chapter.

1.1 Motivation

In the last couple of years, strong effort has been made to replace oil and gas as the primary sources of energy. However, considering global industrialization and the improvement of living standards, long-term trends suggest that the worldwide demand for energy will continue to rise [@USE]. Today, there is already a functional and cost effective global infrastructure for the production and distribution of oil and gas. Therefore these two resources are likely to remain in their dominant position. Unfortunately, most of the large and accessible oil fields in the world have already been discovered as well as exploited. Therefore current hydrocarbon exploration efforts focus on solving the difficulties that are associated with discovering new fields or reach missed oil and gas deposits in already developed fields. The search for hydrocarbons is often associated with high costs, economic and environmental risk. This is the reason why companies spend a large amount of capital on equipment and person even before the actual exploration begins. Exact localization techniques have thus become a cornerstone in corporate business strategies for exploration & production (E&P) companies operating in the market for natural resources such as oil and gas. Seismic surveys are used to identify areas with a high probability for finding oil and gas deposits. Seismic data is acquired to generate a three dimensional model of the subsurface. Seismic interpreters interpret the model and make informed decisions about whether or not oil and gas deposits are located in the geologic structures. Traditional interpretation approaches of large seismic data sets are labor-intensive and time-consuming processes of manual tasks based on using stacks of paper sections, only including a small set of the data within the seismic volume. However, analyzing the seismic data is a complex task, due to the data's unique layered structure. Seismic interpretation is »part science« and »part artistry« [Haroo]. There is an indispensable necessity of expertise of geoscientists for identifying opportunities and plays in the data.

Today, powerful computer workstations have mostly replaced the traditionally based workflow. In order to support geoscientists, computer-aided techniques are used for an automatic detection of potential features of interest in the seismic data. Software for seismic interpretation like Petrel [@Pet] or Decision Space Desktop [@Lan] supports the visualization and analysis of seismic data in a more interactive and structural way compared to paper based approaches. These applications are usually designed for traditional Graphical User Interfaces (GUI). They are driven by mouse and keyboard interaction and are based on the Windows, Icons, Menus and Pointer (WIMP) paradigm. Although the WIMP paradigm had great success over a decade of years, the emergence and the subsequent commercial success of mobile devices, which are using novel ways of interaction, illustrates that there is additional effort needed to enhance traditional desktop applications. The potential for a significant commercial impact included in this shift is also discerned by leading industry companies. This is verified by the adaptation of Microsoft Windows 8 [@Micb], the latest release of the world's most widely used desktop operating system [@Net], to be completely designed to support a multitouch based interaction paradigm. In recent years, interactive displays with a broad diversity of form factors have been developed. In the introductory keynote of the iPad2 [@Appa], Steve Jobs insinuates these devices to be

> *»POST-PC-DEVICES THAT NEED TO BE*
> *EVEN EASIER TO USE THAN A PC.«* [@Appc]

The simplicity of these devices comes from their capabilities of providing more natural ways of interaction by recognizing finger and hand gestures or other common input devices like pens and stylus. Research efforts [ML89; Alb82; @Eur; BFW+08] showed that these Natural User Interfaces (NUI) have a high applicability resulting in more effective utilization of human motor skills in real world tasks.

For the seismic interpretation workflow especially the combination of multitouch and pen input is promising, since the fine-granularity of the analyzed data and an interactive exploration of the huge data sets are demanding dexterity, precision, and usability which are well suited to benefit from pen and touch characteristics respectively. Hitherto, combinations of these novel technologies have not been used in software products for professional-level tasks like seismic interpretation. Therefore, this thesis investigates how bimanual pen and touch interaction can be used to increase the efficiency of the seismic interpretation workflow to have an influence on seismic interpretation software packages of the future. Adapting novel pen and touch based interaction techniques to the seismic application domain is challenging, since the underlying human computer interaction (HCI) techniques of these applications are still reliant on conventional mouse and keyboard interaction. With further progress of computing, communication and display technologies, it is widely assumed that the existing HCI techniques are a bottleneck in the effective utilization of the available information [SPH98]. A successful adaptation of novel interaction techniques to this domain encompasses a variety of research domains, such as cognitive psychology, software engineering and HCI, resulting in a cross-disciplinary research subject.

1.2 Goals of this Thesis

The goals of this thesis are to investigate ways how multimodal interaction, realized through the combination of pen and touch input, can be applied to the currently existing workflow of seismic interpretation. Thereby, interactive displays will be introduced in the WIMP dominated area of desktop applications. In particular, the work in this thesis has the following two specific research goals:

1. Development of novel interaction techniques based on the combination of pen and touch input, to enhance the currently existing manipulation and editing tasks of the seismic interpretation workflow.
2. Integration of these novel interaction techniques into the established WIMP dominated area of desktop applications, without interfering with the system and complementing already existent functionality.

The open-source seismic interpretation framework OpendTect [@Opeb] serves as an example for a first integration of the combination of pen and touch interaction in the area of seismic interpretation. The integration into this widely used application framework allows the evaluation of the implemented techniques with regard to their real world applicability. A key consideration of this integration into an established framework is the preservation of already available functionality. The developed interaction techniques should seamlessly be integrated into the system without interfering with each other or existent features.

These interaction techniques should consist of combinations of pen and touch input to validate new user feedback containing unimodal pen (pen only), unimodal touch (touch/multitouch only) as well as multimodal pen and touch input. Defining new interaction metaphors by converting them from the physical domain into digital software interaction will be the basis of this work. As a result the existing seismic interpretation workflow should benefit by using the combination of pen and touch interaction.

Therefore, the system has been qualitatively evaluated and compared to the existing WIMP based techniques in an informal user study. The results enumerate key issues and points for consideration of the development of multimodal interaction based on the combination of pen and touch input. These considerations can serve as starting point for future research and help software engineers to focus on task requirements, to isolate problem areas and to choose appropriate multimodal interaction strategies for their specified tasks. In conclusion, an iterative design process consisting of redesign and evaluation of chosen strategies should be the incitement to gather most of the benefits for the seismic community.

1.3 Thesis Outline

This thesis is divided into 8 parts.

Section 2, following this introduction, presents the geological background and puts the subject of seismic interpretation into the context of hydrocarbon exploration.

Section 3 starts with a working definition of bimanual interaction including salient features. Based upon this, related work that constitutes the background and context of multimodal interaction is presented. Furthermore, interactive displays and the facility of mode switching techniques are described.

Section 4 introduces computer-aided engineering applications, the background of the application framework, in which the new interaction techniques will be integrated.

Section 5 presents the design method for pen and touch based seismic interpretation based upon the findings of section 3. This section concretely describes how the thesis addresses these design considerations for the seismic domain.

Section 6 lists the details of the implemented system. This does not only include core entities and their ensemble to realize a seamless integration into OpendTect, but also challenges and problems during development and the implemented solutions.

Section 7 contains the description of the conducted user study to evaluate the novel interaction techniques.

Finally, Section 8 summarizes the findings gathered from this thesis and recommends further tests and avenues of research to forward the understanding of novel interaction techniques based on pen and touch input.

1.4 VRGeo Consortium

The VRGeo Consortium [@VRG] is a consortium of the international oil and gas industry and constitutes the context in which this work was elaborated. In December 2012 a first evaluation of the combination of pen and touch input for seismic interpretation took place at the VRGeo meeting with members of the VRGeo Consortium. A first prototype was implemented as a plugin, extending OpendTect to simultaneously handle pen and touch input. This prototype presented first ideas of how pen and touch can be used in the seismic interpretation workflow. As a result a lot of valuable feedback could be included in this work and was evaluated in a second evaluation in June 2013.

02 // Hydrocarbon Exploration

2 HYDROCARBON EXPLORATION

Hydrocarbon exploration is a term describing the search for petroleum deposits in the subsurface of the earth. This search is constantly ongoing and addressed by geologists over a decade of years [JCG08]. As mentioned in the introduction of this thesis, the major oil fields have been discovered and future finds are likely to be smaller and more complicated to identify and to produce. Fortunately, the progress of computation and communication technology did not pass by the field of hydrocarbon exploration. In recent years, new exploration techniques have improved the way geologists can help in identifying potential oil and gas deposits and re-evaluate existing ones. Unlike these advantages, exploration still comprises high costs and economic, as well as environmental risks. Computer-aided techniques are used to keep the complexity of the exploration maintainable and to help in increasing the probability of exploration success.

The following sections will give a brief description of the involved steps in the search for hydrocarbons and will place the process of seismic interpretation in the overall context of hydrocarbon exploration. First, it will be explained how potential locations of petroleum deposits are extracted and located from the subsurface data, followed by the description of the necessary data processing steps. Finally, there will be a demonstration how the data can be interpreted by geoscientists.

2.1 Seismic Data Acquisition

Hydrocarbons can not be found easily. The ultimate goal is to find traps for oil and gas called reservoirs because normally oil and gas would disappear through the different layers due to temperature and pressure in the earth‘s subsurface. The first step in the search for hydrocarbon reservoirs is seismic data acquisition, which aims at measuring seismic data of high quality to generate high resolution images of the subsurface and accurate rock properties at minimum cost. Although the whole process of seismic data acquisition is complex, its most basic requirements are an energy (sound) source, a receiver and the layered structures of the earth subsurface [GS04]. The process, commonly known as the *seismic method*, depends upon changes of the properties of sound waves transmitted through the rock. The acoustic properties of rock are causing different changes of these transmitted sound waves. Figure 2.1 shows a generic diagram of the seismic method.

Seismic sources at or just below the surface generate a signal, that is transmitted in all directions down into the earth. Different physical acoustic properties of the geological

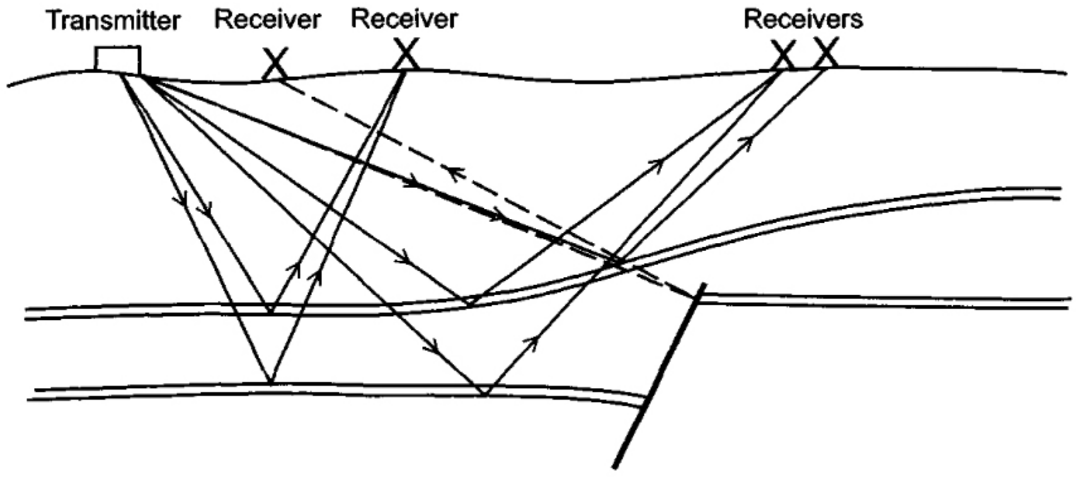


Figure 2.1: Schematic diagram of generating sound waves during seismic data acquisition [GS04]

layers cause the seismic signal to reflect and refract at the boundaries of these layers. Every reflection of the seismic signal transmitted back to the surface is called a *seismic event* [WL06]. Receivers are placed on the surface of the earth, waiting to record these seismic events. While the measurement of one receiver is called a *seismic trace*, the combination of the recordings of all receivers due to one source is called a *shot record*. Usually, a seismic survey consists of a large number of shot records, each recorded at different locations. Subsequently, the measured seismic reflections of the entire seismic survey are processed to create an accurate structural and lithological image of the subsurface. This process is also called *seismic imaging*.

In general, offshore seismic is logistically easier to acquire than land based operations. One reason for this is that there is no need to continuously move land receivers, called *geophones*, around by hand or dig holes for explosive devices. In addition, there are no discrepancies between E&P companies and local population because of dynamite blowing up bits of their land.

2.2 Seismic Data Processing

At the end of the acquisition of seismic data, the data is stored in tapes for later retrieval and processing. At this moment the data is still in its raw form. To get a picture that actually looks like the subsurface beneath the earth, the data has to be processed. It takes a large supercomputing cluster to turn the raw recorded data into the final image of the subsurface geology used by seismic interpreters to make their drilling decisions. In general, algorithms go through all the different traces made by the wavelets and filter out unwanted energy, such as inter-reflections or vibrations made by a tractor nearby during seismic data acquisition.

The primary aims of seismic processing are to enhance the interpretable seismic information relative to the noise in the signal and place the seismic events in their correct x, y, z space [GS04]. The following processing steps illustrate a simplified processing flow for seismic data according to [YD01a]:

1. Preprocessing
2. Deconvolution and filtering
3. Common midpoint (CMP) sorting
4. Velocity analysis
5. Normal moveout (NMO) correction and muting
6. Static corrections
7. Stacking
8. Migration

Note that a real-world processing workflow for any given seismic data set will be different from this simplified example and may include steps not considered here. For a complete and more detailed description of all involved steps see [YD01a] and [YD01b].

Figure 2.2 illustrates the resulting seismic data set after each involved step, beginning with the raw data. The first step of preprocessing (1.) includes amplitude gaining of recorded waves and converting data from the field recording format into a format that is more usable for software processing. Additionally, the geometry setup of the acquired field is stored and unusable traces are removed from the data set. The objective of the deconvolution step (2.) is

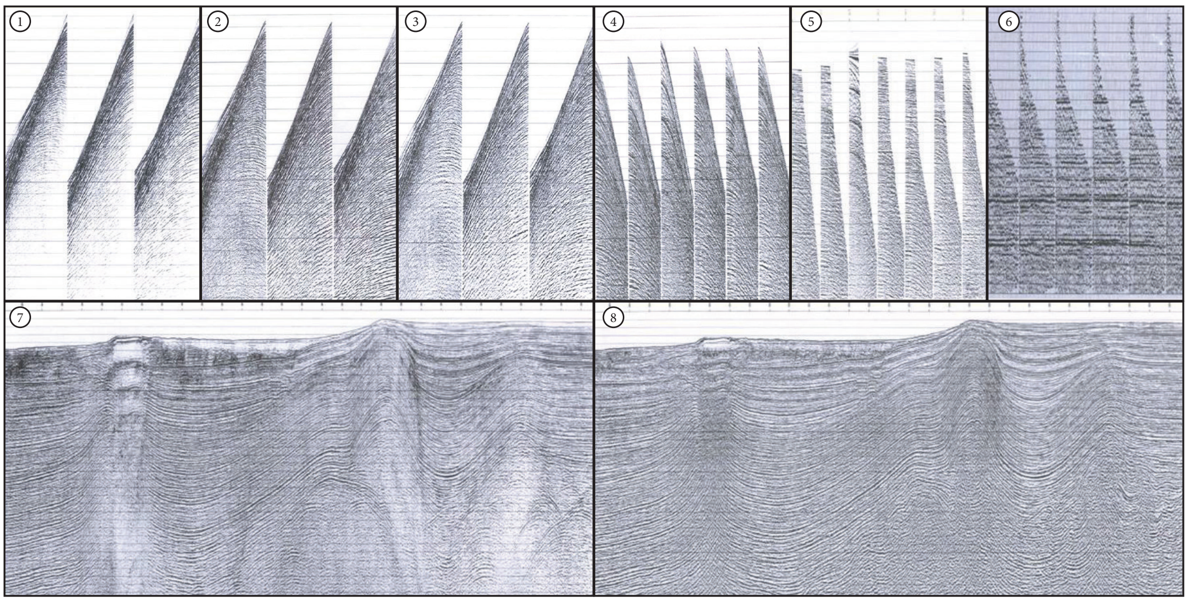


Figure 2.2: Simplified seismic data processing workflow. 1. Raw data, 2. Preprocessing, 3. Deconvolution, 4. CMP sorting, 5. NMO corrections, 6. Static corrections, 7. Stacking, 8. Migration [YD01a]

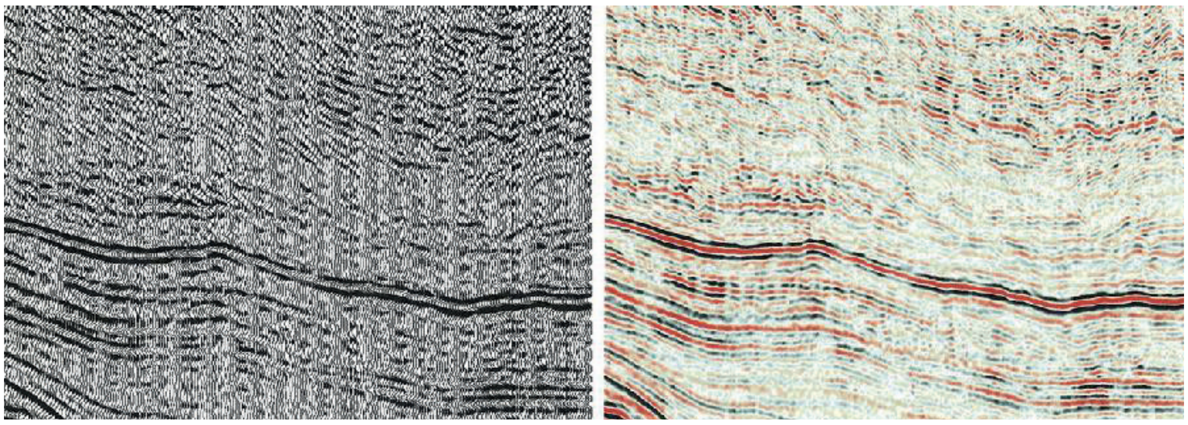


Figure 2.3: Comparison of seismic traces displayed as variable area/ wiggle plot (left) and as variable density color plot (right) [BSR03]

to increase the vertical resolution of the data resulting in a supplement of the ability of the seismic interpreter to identify thin subsurface layers. The CMP sorting (3.) uses the stored data from the preprocessing step to sort traces from shot gathers to CMP gathers, preparing the data for further processing steps. The velocity analysis (4.) determines the layer velocity distribution in vertical and horizontal directions. The following NMO correction (5.) will flatten the hyperbolic seismic curve structures into horizontal reflections. Severely stretched curve parts are muted (zeroed). Afterwards, static corrections (6.) enhance the derived layer velocities and the stacked section quality. The stacking step (7.) sums all traces in a CMP gather into one single trace, resulting in an enhanced signal-to-noise ratio of the data. Finally, the migration step (8.) maps reflectors to their correct positions and removes diffractions.

The resulting image after the migration step illustrates the traditional way of displaying seismic data, the so-called variable area/wiggle trace display. These were presented to interpreters as a stack of paper records. Interpreters marked up regions of interest and tried to trace the path of these regions across different paper sections. Here the seismic traces are displayed as continuous curves. Because of the similarity in appearance it is hard for interpreters to distinguish between positive and negative (peaks and troughs) amplitude values of the data samples [Bro04]. Today, most seismic data are stored digitally and analyzed on powerful computer workstations. The arising capabilities allow the interpreter to assign different colors to different ranges of amplitudes. A common way to color the peaks is using a blue color, where stronger positive values are colored deeper blue. The troughs are usually colored red, with more negative values are being darker red and zero values are colored white. The major benefit of this variable density display is a clear visual distinction between peaks and troughs, enabling the interpreter to gain more information about the seismic data. Figure 2.3 displays a comparison between variable area/wiggle display and variable density display.

2.3 Seismic Interpretation

Once the data has been processed, it is ready for integration into seismic interpretation software. This allows the automation of several tasks, like numerical analyses or different attribute generation. However, seismic interpretation is »part science« and »part artistry« [Haroo]. The petroleum geologist, who works on seismic interpretation, is an integral part of the complete hydrocarbon exploration workflow. It is in his responsibility to make an informed decision about whether or not oil and gas deposits are located in the geologic structures and based upon these considerations to advise where to drill. To address this task, interpreters need to integrate data and concepts from several sources and apply a considerable amount of intuition and experience. This expertise can only be gathered by studying the field of rock and sediments properties and being familiar with the software for the particular seismic workflow step. Section „2.2 Seismic Data Processing" revealed the traditional interpretation approach of using variable area / wiggle displays. Today, powerful computer workstations have mostly replaced the traditional based workflow. Besides automation of numerical analyses, a digital storage of the seismic data offers additional possibilities, such as a rapidly testing of different hypothesises or the combinations with other software packages. Nevertheless, the major benefit of using a computer-based software package is interactivity, which is not possible with paper records. Using interactive workstations dramatically improved productivity by allowing the interpreters to work with the seismic data with adaptation on the currently examined features of interest. This includes reprocessing of seismic data by interpreters themselves, defining interactively their own survey setup or color selection. Moreover, interpreters can now directly evaluate different processing routines, like filter options, trace balancing or deconvolution properties on the already migrated data sets. This allows focusing on and enhancing certain aspects of the data, quickly incorporating new data sets into the interpretation process and removing unwanted noise. Interpreters are able to combine 2D and 3D data sets from different timestamps of the same seismic surveys and view these as fusioned single continuous data sets. All of these new arising opportunities are only consuming a fraction of the time compared to using paper based interpretation workflows.

Data Display

For the integration of seismic data into interactive seismic software packages the data is arranged into a format referred to as *3D data cube* [EMA97]. For this, the 2D survey lines (see section „2.2 Seismic Data Processing") are placed much closer together to obtain a more detailed three dimensional model of the seismic data providing organized access to the volumetric information. This technique is called a *3D survey* [WL06]. Generic and concrete views of a 3D survey are shown in Figure 2.4.

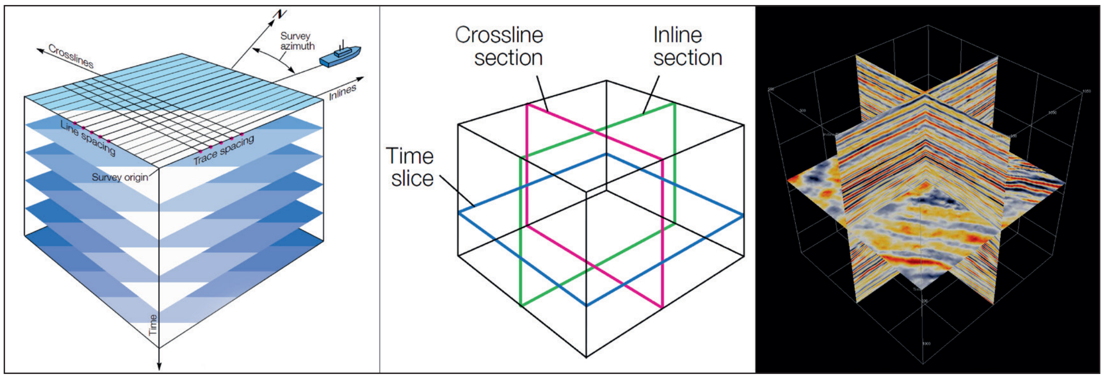


Figure 2.4: Generic view of a data cube according to data acquisition (left) [JTSS94], the definition of the seismic sections in the data cube (middle) [JTSS94] and a concrete view of the data cube sections in the 3D survey of OpendTect [@Opeb] (right)

Visualizing the data in a data cube of a 3D survey offers the ability to view sections through the data in any orientation. The appearance of these sections might seem similar to conventional 2D seismic lines shown in Figure 2.4 (right). Nevertheless, these sections are superior regarding resolution due to their production using 3D processing. The creation of a 3D seismic survey is not an easy task and a geographically correct display of the 3D data heavily depends on orientation information such as the geographic coordinates of the origin of the survey, azimuth, the order and spacing between shot records. Figure 2.5 (left) shows a marine seismic example, in which lines with the same orientation like the source locations during the data acquisition are called *inline* section. Vertical lines perpendicular to these have the same orientation like the receivers during data acquisition and are referred to as *crossline* sections. Horizontal cuts through the data are called *time slices*. Note that applications for seismic interpretation allow the visualization and interpretation of seismic data in a lot more ways than illustrated here.

Horizon Tracing Workflow

Although the 3D visualization tools of seismic software packages offer great potential for manipulation and interpretation of the data, the common workflow for interpretation often relies on using 2D seismic sections taken through the cube [JTSS94]. A generalized workflow for interpretation of either 2D or 3D seismic data can be summarized as follows (according to [Haroo]):

1. Collect all Pertinent Data and Reports
2. Scan Records for Polarity, Static Shifts
3. Scan Through Sections (Line by Line)

4. Tie Well and Seismic Data
5. Pick Horizons and Faults
6. Seismic Stratigraphic Analyses
7. Structural Analyses
8. Contouring/Mapping

Some of the listed points are executed in a loop and the boundaries between several of them are fluid. For the sake of this thesis, point *3. Scan Through Sections (Line by Line)* and *5. Pick Horizons and Faults* are of interest and will therefore be discussed further. The picking of faults is not in focus of this thesis. For a description of fault picking and all other listed steps please have a look at [Har00] and [BSR03].

This section showed that seismic interpretation software allows interpreters to apply different color schemes to the seismic sections with relation to the seismic amplitudes of the respective seismic event. Using the resulting image, interpreters spend most of their time analyzing these amplitude ranges trying to find anomalies, which may lead to oil and gas deposits. Most of the anomalies in the data are occurring when the generated seismic signal is reflected and refracted at the boundaries of the different geological structures in the subsurface of the earth (see section „2.1 Seismic Data Acquisition“). The reflection is known as a *horizon*, which is defined as an interface represented by a seismic reflection of two bodies of rock, each having a different seismic velocity, density, porosity, fluid content or combinations of those [@Sch]. By scanning through the sections and processing one section after the other, interpreters try to create *horizon slices*. A horizon slice is an arbitrary surface through the data cube created by linear interpolation between picked surfaces [BSR03]. The workflow referred to as *horizon tracing* is as follows: Processing one seismic section after the other (step 3. in the interpretation workflow), the seismic event of the horizon under consideration has to be identified on each section. Then, the interpreter marks locations with good continuity and signal-to-noise ratio on each section [YD01b] (step 5. in the interpretation workflow). While these locations are called *seed points*, the marking of those is called *seed picking*. Figure 2.5 displays various different stages of the horizon tracing workflow.

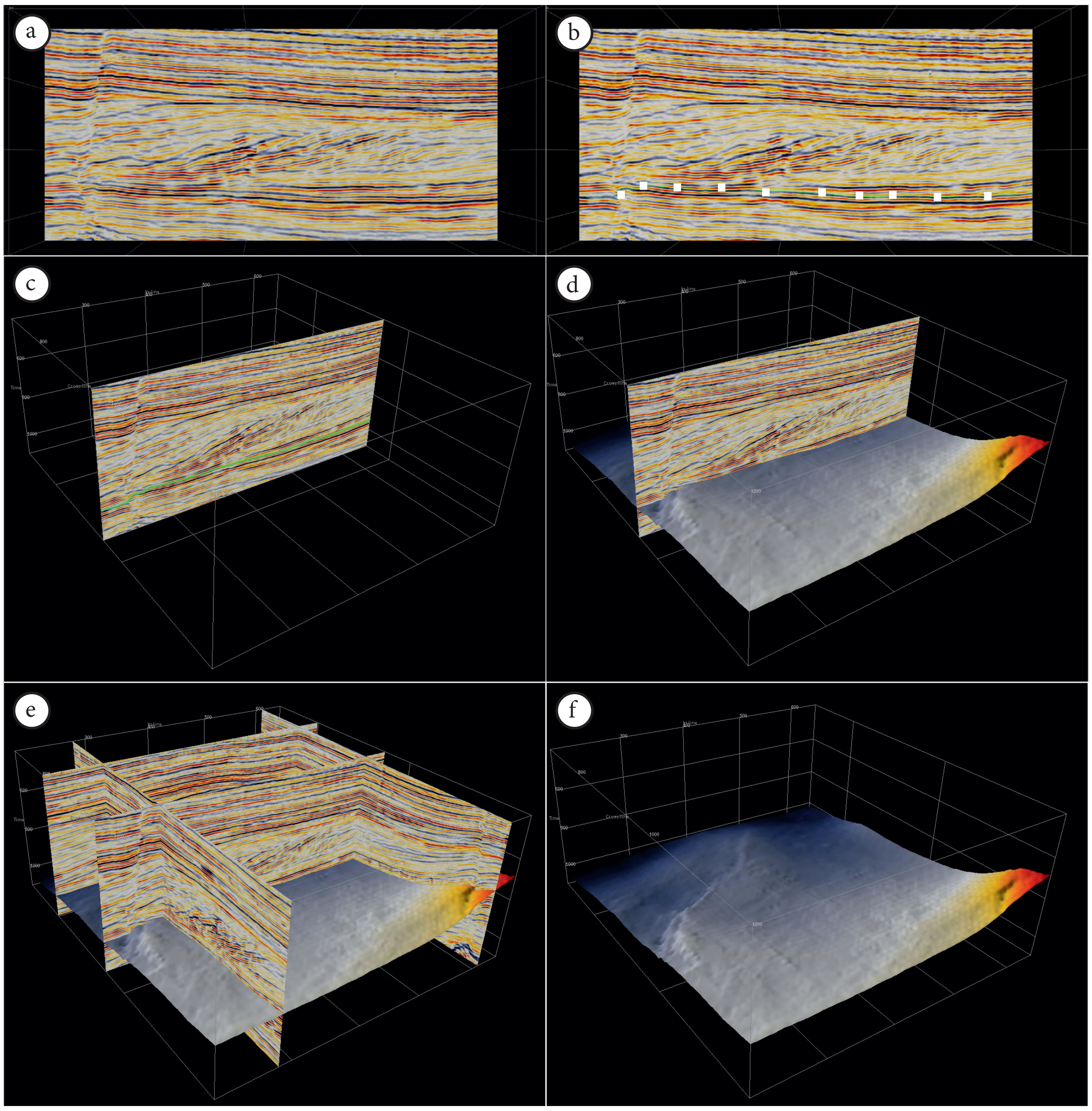


Figure 2.5: Horizon tracing using OpendTect. a) Processed crossline, b) Picked seeds (white rectangles) on event of crossline, c) Perspective view of crossline, d) Perspective view of whole horizon, f) Several inline and crossline sections of the horizon, e) The horizon slice only

03 // Novel Human Computer Interaction

3 NOVEL HUMAN COMPUTER INTERACTION

Recently, novel devices and technologies raised to establish more natural and reality-based interaction paradigms. Jacob et al. [JGH+08] call these devices »post-WIMP« devices. Building upon the pre-existing knowledge of the everyday non-digital work of users, the goal is a reduction of the *gulf of execution*, which depicts the gap between the intended action of the user and allowed action of the system [Nor02]. Humans are used to multimodal communication, including speech, eyes, gestures, tactile feedback or more generally a high degree of freedom (DOF), when interacting with physical objects. The mouse, however, only provides two spatial DOF, which requires breaking up multi-parameter tasks into a sequence of multiple steps. This kind of interaction does not correspond to common interactions in the physical world, as it interrupts the natural workflow. Direct bimanual interaction is a potential solution to overcome these drawbacks, which has led to the emergence of novel devices with interactive multimodal capabilities. There are devices unifying the space of interaction and display, which classifies them as touchscreens [Pos09]. These devices serve as input and output device likewise. Wherever a touchscreen is touched, digital objects can be activated and an event can be triggered. This classifies touchscreens as absolute input devices [ZM98], since the position of direct input and the position of direct manipulation of virtual objects are spatially coinciding. A user is presented with familiar interaction techniques by exploiting close relations between the known real world interaction and the new digital world interaction. The use of touchscreens offers additional possibilities of interaction. Beside direct touch input, there is likewise the opportunity of creating gestures for input. Other devices are even more suitable to embrace a user's natural behavior by distinguishing between additional input devices like pen input. However, these modern interfaces as well as the mentioned traditional interfaces are characterized through their reliance on a single mode of interaction, e.g. mouse movement, key strokes or speech input, which classifies them as *unimodal* [@Weib]. This limitation does not replicate natural interaction, since humans are using multiple modes of communication and interaction when they are interacting in the physical world, incorporating multiple senses. In contrast to unimodal systems, *multimodal* systems are possessing more than one mode [@Weia]. Nigay and Coutaz [NC93] define multimodality as the capacity of a system to communicate with a user along different types of communication channels and to extract and convey meaning automatically. Thus, multimodal systems are assisting users to control systems more easily through combining several interaction modalities. They therefore provide more powerful interaction paradigms than any single modality would be able to provide on its own.

This chapter will introduce the recent developments and research efforts regarding novel human computer interaction. It begins with introducing the foundations of bimanual human interaction and will provide basic concepts regarding the characteristics of human interaction techniques. The explanations will focus the salient features, on which the development of the new interaction techniques of this thesis is based upon. The following subsection will introduce basic findings and issues in the broad and multidisciplinary field of multimodal interaction. This is not an exhaustive survey of all aspects regarding multimodal interaction. For further insight please consider [DLO09] as a start. Subsequently, interactive displays are described, which are becoming more and more present in the daily computer experience of humans, followed by a summary of input techniques, whose usage goes beyond the incorporation of conventional input devices, like mouse and keyboard. Therefore, research efforts are presented building upon the usage of input devices, like pen devices, multitouch interaction or the combination of both classified as multimodal input. Mode switching techniques are discussed realizing a higher degree of parallelism during interaction with an application. Finally, example applications, incorporating novel interaction techniques are presented.

3.1 Foundation of Bimanual Interaction

Bimanual or two-handed interaction is an active topic of human computer interaction, since it is the most common way of interaction humans are comfortable with from their everyday manipulation experiences. Buxton and Mayers [BM86] assumed that the necessary motor skills required to perform bimanual computer tasks, can easily be acquired or are even already existent. Their experiment compared the performance of selection, positioning and navigation tasks between a one-handed setup and two-handed setup. They were confirmed by their results, which lead evidence for a correlation of the task execution time and the incorporated degree of parallelism. Their results showed that a higher degree of parallelism allows faster and more accurate interaction.

Guiard [Gui87] observed similar results in writing tasks. He showed that hand writing tasks involving both human hands have a much higher performance then those solely relying on one hand. He investigated the distribution of work between the dominant hand and the non-dominant hand. For a left-handed person the dominant hand is the left hand, while for a right-handed person, this is the right hand respectively. This investigations lead him to the definition of the *Kinematic Chain Model*. This model describes the division of labor in human skilled manual activities. It is important to note that there are significant differences between varieties of bimanual interactions. Guiard identifies tasks that are *unimanual*, such as throwing darts. Performing this kind of interaction incorporates hand actions that are more convenient when executed using only one hand. In contrast to this,

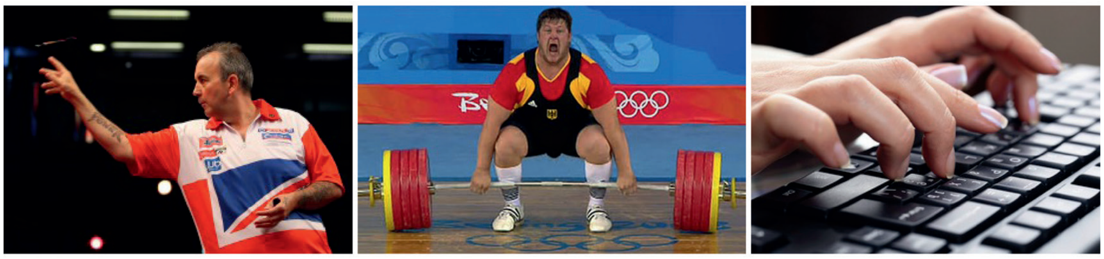


Figure 3.1: Unimanual interaction (left) [@Pro], bimanual symmetric synchronous interaction (middle) [@The], bimanual symmetric asynchronous interaction (right) [@Rus]

bimanual symmetric interactions incorporate both hands. Here the hands are performing identical interactions, either *synchronously* (in parallel) or *asynchronously* (one hand after the other). Examples of the mentioned interaction techniques are shown in Figure 3.1. However, a third class of interaction is more in focus of Guiards investigations, by name *bimanual asymmetric interactions*. Here the hands are performing a complementary task, but each of them by executing different actions. The previous example of hand writing, where the actions of the dominant hand and the non-dominant hand are closely correlated with each other, serves as good demonstration: While the dominant hand (DH) uses a pen to write on a paper, the non-dominant hand (NDH) manipulates the orientation of the page to complement the DH action. For a definition of the underlying mechanisms of bimanual asymmetric interaction, resorting on the current notion of hand preference is inadequate. This notion declares a superiority of the DH over the NDH, since it is more often preferred by humans during interaction. In contrast to this, in bimanual asymmetric activities no hand can neither be excluded, nor plays a more important role than the other hand. This results in an asymmetric division of labor between the roles of the NDH and DH hand implying that the actions performed by each hand are not randomly chosen. Moreover, these actions are chosen with respect to the capabilities of the particular hand manifested through three principles defined by Guiard. These three principles determine the asymmetry of human bimanual interaction and are referenced by various research efforts [HYP+10; BFW+08]. In the following each of these principles is presented:

Right-to-Left Spatial Reference in Manual Motion

The first principle denotes a spatial relationship between the DH and the NDH, with the DH finding its spatial reference in the product of the movement of the NDH. That is, the NDH dynamically adjusts the frame in which the DH inserts content.

Left-Right Contrast in the Spatial-Temporal Scale of Motion

This principle focuses on the different characteristics of the DH and the NDH. Capabilities of the DH are fine control movements, both regarding spatial and temporal accuracy. Again, the handwriting on a sheet of paper is a good example. While the DH moves the pen, the NDH may rearrange the sheet of paper simultaneously. The pen movement is much faster in time and spatial frequency compared to the reorientation of the sheet of paper on the table. However, this principle does not assume superiority of the DH over the NDH. Guiard substantiates this with the complementary function of the two hands resulting in unequal capabilities to perform high resolution tasks.

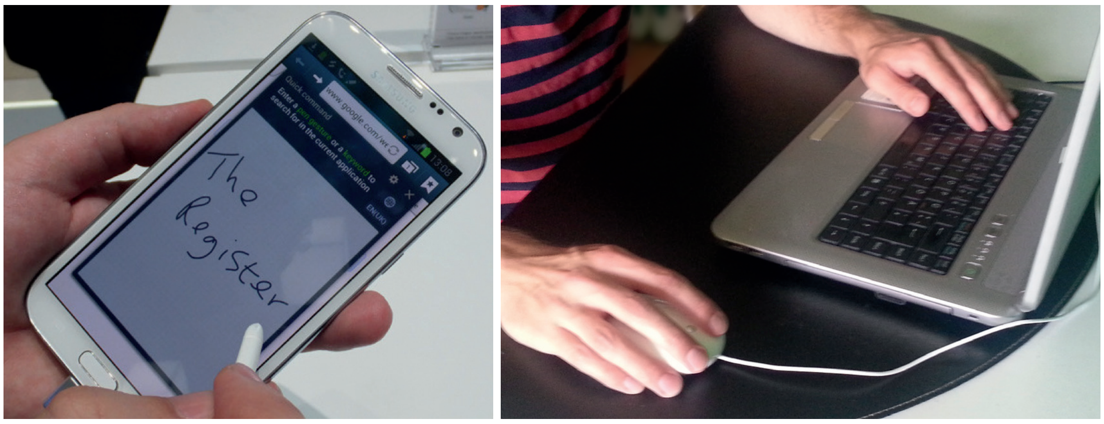
Left-Hand Precedence in Action

This principle simply denotes that the NDH initializes the bimanual asymmetric activity, thus contributes earlier to the complementary action.

Grounding on several user studies and research efforts, Guiard suggests that bimanual asymmetric interaction techniques are most suitable for execution of actions with a high degree of complexity. Besides this evidence of a superiority of bimanual asymmetric interaction techniques the emerge of novel devices makes clear that this kind of interaction becomes more and more present in daily interaction experiences of humans. Figure 3.2 (left) demonstrates the division of labor between the NDH and the DH in mobile device interaction. Similar to the hand writing example, here the NDH rearranges the orientation of the device, while the DH interacts and inserts content. In addition, the asymmetric division of labor between the dominant hand (mouse) and the non-dominant hand (keyboard) is shown in Figure 3.2 (right).

Despite the clear separation of the DH role and NDH role, Bowman et al. [BKLP04] note, that natural tasks embody a certain complexity, making often and rapid switches between symmetric and asymmetric modes necessary.

Figure 3.2: Bimanual asymmetric interaction with a mobile device (left) [@Dor] and bimanual asymmetric division of labor when using traditional input devices (right)



3.2 Multimodal User Interfaces

In this section basic components and characteristics of multimodal user interfaces (MUI) are identified. At first, the background of the human computer interaction in a multimodal system is described, followed by the illustration of the aims and advantages of such systems, including the reason why to introduce such systems. Afterwards, implementation guidelines of MUI systems are provided and finally, the underlying principles of the combination of modalities are displayed.

3.2.1 Theoretical Principles

Grounding upon well accepted findings and taxonomies, Dumas et al. [DLO09] define a model of multimodal man machine communication, listing the major concepts that should be considered when building a multimodal system. The main components of such a system are the *fusion of input modalities* (combination of input modalities) and the multimodal *fission of input modalities* (system response) to generate an adequate message to a user, with regard to the context of use, preferences and profile. Dumas et al. [DLO09] representation of multimodal interaction is shown in Figure 3.3. The figure illustrates the different stages in

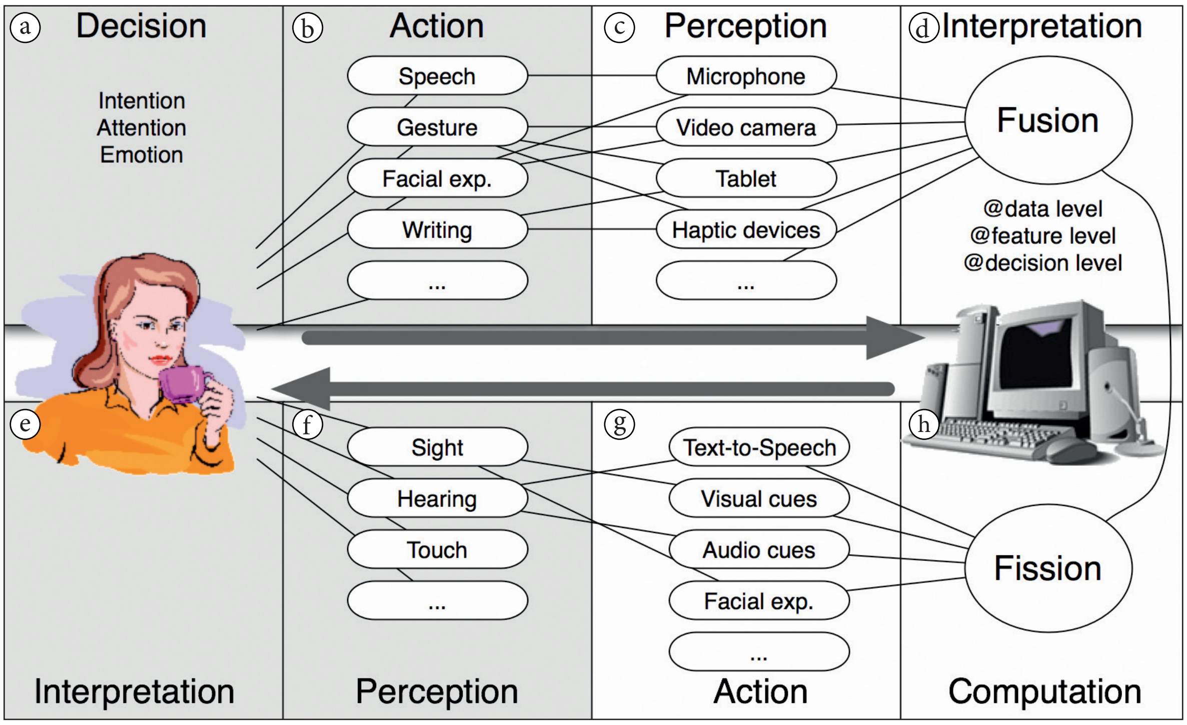


Figure 3.3: Representation of multimodal man machine interaction loop [DLO09]

the multimodal man machine interaction loop from a user-centered view (grey background) and a system-centered view (white background) respectively. In a typical multimodal man machine interaction, both, the user and the system, go through four stages. The interaction starts with the user making a decision (a), where she or he prepares the communication message content. In the second action state of the user (b) direct intervention of the user is executed by choosing the way (input modality/device) for transmitting the prepared content to the system. Now, in the perception state of the system (c), the system receives this information from one or multiple attached sensors (touch-sensitive surfaces, digitizer tablets, webcams etc.). At this stage, the information is often already weighted through input recognizers for avoidance of information overflow. During the interpretation state (d), the system is able to use the weighted information and give custom meaning to the various input streams collected. This is typically the state in the man machine loop, where different input modalities are joined together (fusion). This combination is essential for the computation state (h), where the system executes the action following the business logic and rules defined by the developer in accordance to the fusion of modalities. Depending on the executed action in the computation state, an answer is generated and transmitted during the action state of the system (g, system response, fission). The user perceives this response (f) through either one or multiple senses and can interpret (e) this information under consideration of the immediate circumstances, e.g. the context of use.

3.2.2 Aims and Advantages of Multimodal Systems

In their survey of multimodal interfaces, Dumas et al. [DLO09] identified two objectives of such systems. The first one is to support and accommodate user's perceptual and communicative capabilities, while the second one deals with the integration of computational skills of computers in the real world, by offering more natural ways of interaction with humans. Interaction through the use of speech or gestures, or interaction that is generally based on all five senses, emphasizes a richer and more natural way of communication. Sharma et al. [SPH98] also claim to have found evidence for the integration of multimodal capabilities. From a practical point of view, they describe today's HCI systems as »cumbersome« and »unnatural« because of their dependence on traditional input devices like mice and keyboards. Research efforts have shown that 95% of the subjects prefer multimodal interaction over unimodal. This is unsurprisingly, because the natural way of interaction between humans is multimodal as well. Humans tend to speak, look and gesticulate at the same time. The use of all senses provides further clues about the immediate environment, like for instance listening to the tone of a person's voice, looking at the person's face, smelling or touching objects nearby. Sharma et al. [SPH98] conclude that, although unimodal interaction allows conveying the intent of the interaction to the computer, the »ease« in doing so is unsatisfactory.

The aptitude of multimodal interfaces based upon natural qualification is not the only reason to incorporate multimodal capabilities into human computer interaction. The advance of communication and computer systems results in a higher complexity of applications. With this, a single modality does not permit a user to interact effectively across all tasks and environments [OCW+00]. Multimodal interfaces are giving access to different ways of communicating input to an application either by the combination of modalities or by switching to a better suited modality with regard to the specific task or the environment under consideration. The choice of the modality through the user himself is an important design enhancement, since individual input modalities are well suited in some situations and less ideal or even inappropriate in others [@BUX]. Later sections (see section „5 Pen & Touch based Seismic Interpretation“) discuss the characteristics of pen input and touch input as individual modalities as well as issues specific to their benefits when used in a combined multimodal interface.

3.2.3 Guidelines for Multimodal User Interfaces

Reeves et al. [RLL+04] define categories of guidelines representing a preliminary effort to establish principles for multimodal interaction design. Similar to Sharma et al. [SPH98], they are convinced that today’s HCI techniques are inappropriate, making new interaction paradigms and guidelines necessary to facilitate the design of multimodal systems. The guidelines of Reeves et al. [RLL+04] are presented in the following.

Multimodal systems should be designed for the broadest range of users and contexts of use

Since the acceptance and validation of an application is highly dependent on the background of the target audience, the design of a multimodal system should take these circumstances into account. This includes users’ psychological characteristics, cognitive abilities, levels of experience, domain and task characteristics, cultural backgrounds, as well as their physical attributes.

Multimodal systems should address privacy and security issues

In situations where users wish to maintain privacy because they might be uncomfortable if certain private information is recognized by others, a multimodal system should adapt to the users‘ explicit preferences. For example, when users enter personal identification numbers or passwords in public contexts, speech input would be inappropriate.

Multimodal systems should maximize human cognitive and physical abilities

Based upon the users' abilities to process information, the design of a multimodal system should support intuitive and streamlined interaction. Since multiple modalities might be supported by the system, it should be avoided to increase the cognitive load of a user. For example, in cases where a user must simultaneously attend to multiple input modalities, redundancy in information presentation is inappropriate.

Multimodal systems should integrate modalities in a manner compatible with user preferences, context and system functionality

The support of additional modalities should only be considered if there is a reasonable amount of usefulness expected, such as improved satisfaction, efficiency or other aspects of performance for a given user and context. Bill Buxton denotes that

> *»EVERYTHING IS BEST FOR SOMETHING AND WORST FOR SOMETHING ELSE.«* [@BUX]

Thus, the way input modalities are incorporated should carefully be chosen. Moreover, dynamic adaptation of the interface with respect to the needs and abilities of different users, as well as different contexts of use, are providing access to complementary and supplementary leverage of modalities according to changes in task and context.

Multimodal systems should include error prevention/handling

The integration of complementary and supplementary modalities makes error prevention and error handling a major advantage of multimodal interface design. From a user-centered point of view, this allows more control over modality selection and avoids errors. From a system-centered point of view, improved system robustness is a major benefit of multimodal systems.

In order to develop both innovative and multimodal interfaces, the guiding principles presented here are a good starting point. This thesis incorporates pen input and touch input as input modalities. Both pen input and touch input allow the creation of gestural interfaces. In addition to all the standard interactions available to desktop systems like typing, scrolling, pointing and clicking, gestural interfaces can take advantage of the whole body for triggering system behaviors. Generally, the movement of any body parts, mimicking or gesturing can be defined as gesture. However, here the term gesture refers to hand markings either entered with a stylus, mouse or directly with the hand, indicating scope and commands [RC91] in contrast to body part movements. When creating gestures for interactions, there are particular circumstances that have to be considered. Gestures can become complicated and thus hardly to achieve or even to remember, resulting in the possibility that the interaction will not be accepted by users. The first step in creating a gestural interface should be the measurement of how suitable a gesture is in a particular context [Safo8]. Saffer [Safo8] identifies several characteristics of well-designed gestural interfaces. Four of these characteristics build the foundations of the implemented interaction techniques of this thesis. The following paragraphs give an overview of these characteristics and explain their focus in detail. Further insights of all characteristics defined by Saffer are provided in [Safo8].

Discoverable

Since gestural interfaces are novel interfaces not known by users, these new features must be recognizable. One significant aspect of this characteristic is that users must be aware of the fact that she or he is interacting with a gestural interface at all.

Responsive

Human beings are used to get an immediate response from their interaction with physical objects. This response gives a feedback to the user signalizing the system has understood the intervention.

Appropriate

Gestures are interactions obvious for all others in the immediate environment of the interacting human being. Therefore, a gestural interface should be adapted accordingly to the context in which it will be used.

Meaningful

In the end this characteristic decides whether users will accept a gesture or not. If the gesture does not fit to the needs of the targeted audience, it will become an obsolete gesture.

3.2.4 Fusion of Input Modalities

The combination of input modalities is one of the challenges in the implementation of this thesis. The goal of this fusion is to extract meaning from a set of input modalities and include this in human computer interaction in order to realize novel interaction paradigms [DLO09]. The task to fusion different modalities is a complex challenge, due to the existence of multiple ways for modeling of multimodal interaction.

		USE OF MODALITIES	
		Sequential	Parallel
FUSION OF MODALITIES	Combined	ALTERNATE	SYNERGISTIC
	Independant	EXCLUSIVE	CONCURRENT

Figure 3.4: The CASE-model (based on [DLO09])

Modeling of Multimodal Interaction

A formal model for describing the different multimodal combinations in interaction is the *CASE-model*, defined by Nigay and Coutaz [NC93]. Figure 3.4 displays an abbreviated version of the CASE-model defined by Dumas et. al [DLO09], which is defined along two dimensions. These are *Use of Modalities* and *Fusion of Modalities*. The original one, which includes an additional dimension, can be found in [NC93].

Use of Modalities This dimension covers the temporal availability of multiple modalities. In a multimodal system, typically, multiple sources of input are coexisting, realizing parallelism. The presence of parallelism, referred to as »Parallel«, allows a user to interact using multiple modalities simultaneously. In contrast to this, »Sequential« means the absence of parallelism. A user is able to use the modalities one after the other.

Fusion of Modalities As discussed in previous sections the combination of different modalities is referred to as fusion of modalities. If there is a combination of different modalities, it is called »Combined«, while the absence of fusion is called »Independent«.

The results of the CASE-model are the four properties, *alternate*, *synergistic*, *exclusive* and *concurrent* with each describing a different way of combining modalities in one multimodal system.

In section „3.1 Foundation of Bimanual Interaction“ bimanual asymmetric interaction techniques were introduced. With regard to the implementation of such interaction techniques, an alternate and synergistic fusion of input modalities, according to the CASE-model, is necessary. The implementation of this thesis incorporates a fusion of modalities based on alternate, synergistic and exclusive approaches. There is no integrating of fusion of modalities based on the concurrent approach. The developed interaction techniques incorporate techniques, which include only pen input or only touch input (exclusive). Additionally, there are interactions where pen and touch input are used one after the other (alternate) and also techniques based on a combined parallel usage (synergistic). Later sections will give more details on the chosen design consideration of this thesis.

3.3 Interactive Displays

In their evaluation of multiple DOF input devices Zhai and Milgram [ZM98] defined a continuum for classifying the directness of computer input devices, in which they allocate special means to the transformation from the *control space* to the *display space*. The control space is represented by the user's controlling action, whereas the display space describes the cursor movement. They conclude in their observation that

> *»THE MORE MATHEMATICALLY COMPLEX THIS TRANSFORMATION IS, THE MORE INDIRECT THE INPUT TECHNIQUE IS.«* [ZM98]

An abbreviated version of their isomorphism (directness) continuum is shown in Figure 3.5. For the original and complete continuum please have a look at [ZM98] . With respect to their continuum, touchscreens and tablets are absolute input devices, realizing a high degree of direct manipulation. Accordingly, the mouse is a relative input device, requiring a special mechanism to guarantee the linkage between a user's control action and the system response (mostly mouse movement). In case of the mouse, this linking mechanism is the constraint to the mousepad. If the mouse is lifted, the linkage between control space and display space is lost. Zhai and Milgram conclude generally, the more isomorphic (more direct) a design is, the more intuitive it is and it requires less learning.

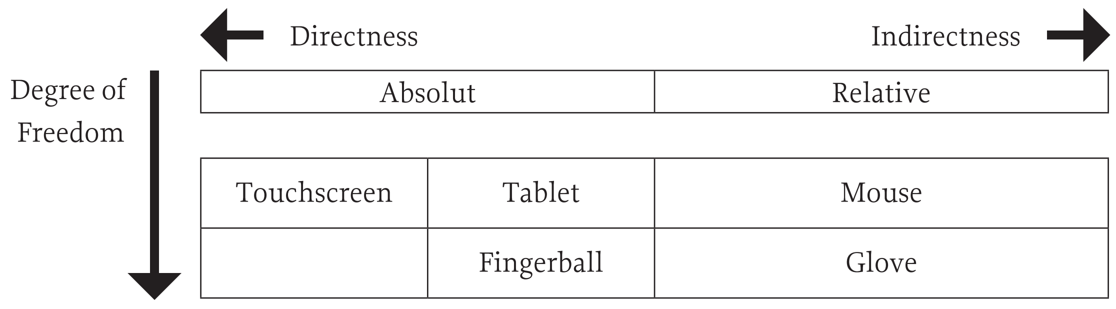


Figure 3.5: Isomorphism continuum of input devices (based on [ZM98]

Albert [Alb82] targets in his seminal comparison of several input devices an acquisition of their performance in cursor positioning tasks. Based on his quantitative measurements a design guideline for the development of new input devices is provided. He compared pen, touchscreens, joysticks, tablets, trackboards and the keyboard with each other. His results showed that finger-operated touchscreens are best in speed but worst in accuracy. He also investigated the influence of direct and indirect *eye to hand coordination*. Eye to hand coordination refers here to the isomorphism of the input device according to Zhai and Milgram [ZM98]. Touchscreens or tablets are direct input devices, since the position of direct input and the position of direct manipulation of virtual objects are spatially coinciding. The drawback of a direct input device is the occlusion of the display area by the moving hand, also realized by other research efforts [Yee04]. Despite this lack of direct input devices, Albert's results showed that a spatially coincident of the display space and the control space yields a significantly faster interaction.

The work of Zhai and Milgram [ZM98] and Albert [Alb82] confirms that an establishment of direct input devices in professional desktop environments is useful. For a seamlessly integration of gestural interfaces in modern desktop applications, the underlying hardware must be capable to support the novel interactions. For the sake of this thesis an interactive display, offering the demanded hardware support, is the most fundamental starting point. Several research studies investigated bimanual multimodal interaction techniques, using a self-built interactive surface by combining different technologies to achieve a simultaneous identification of different input modalities. Besides these prototypic setups there are consumer products available also distinguishing between different input modalities. The following paragraphs will present examples of each of these.

Prototypes

A variety of research efforts investigated bimanual multimodal interaction, using a self-built setup to evaluate new interaction techniques. Brandl et al. [BFW+08] overlaid a DiamondTouch [DL01] interactive surface with the Anoto [@Ano] pattern to distinguish between pen and touch input. Hinckley et al. [HYP+10] used a prototype based on an optical approach. Their system distinguishes respectively between unimodal pen, unimodal touch and multimodal pen and touch. Yee [Yee04] used a combination of commonly used touchscreens and tablet screens to present three applications, which allow interacting using touch input and pen input simultaneously. However, his system is only able to sense a single touch point. Frisch [Fri12] used an optical infrared-based technique, called Frustrated Total Internal Reflection (FTIR) for his investigation of multimodal input for visualization of node link diagrams. In an early study Kurtenbach et al. [KFBB97] used two Wacom digitizer tablets and two Wacom input pucks for bimanual interaction with a tool glass menu.

Although the self-built environments described above are sufficient for specific evaluation purposes, their potential to expand the accessibility of daily desktop computing is limited. However, industry manufacturers of interactive displays perceive this progress and the potential of productivity enhancements of novel interaction techniques. These landmarks helped to convince manufacturers building more general and robust multimodal systems. Examples of these will be presented in the following section, illustrating the potential for a significantly commercial impact.

Consumer Products

The Wacom Cintiq 24 HD Touch [@Wacd] has been introduced in mid-2012. This multitouch- and pen-enabled interactive surface allows an explicit distinction between simultaneous pen and touch input, that is hardly achieved by any other consumer product. This device uses a capacitive sensor for the processing of touch data and an inductive sensor for processing of pen data. Figure 3.6 advertises how the multimodal capabilities of the Wacom Cintiq 24 HD Touch are incorporated into professional artwork workflows.

This device has some features, being a fundamental part of the techniques developed in this thesis. The ability for simultaneous pen and touch input is essential for the implemented bimanual input techniques. Regarding the continuum of Zhai and Milgram [ZM98], the Wacom Cintiq 24 HD Touch offers two distinct ways of input, with each of them including a high degree of directness. This classifies the Wacom Cintiq 24 HD Touch as multimodal input device and it is therefore best-suited for into the evaluation of bimanual interaction techniques.

Additionally, the support of the *confidence bits* of the Wacom's Feel™ Multi-Touch Application Programming Interface allows applications to perform palm rejection during interaction. Palm rejection allows the separation of touch data at such times where it would cause interference with the application a user is currently interacting with [@Waca]. The confidence bits provide information on whether the tablet driver software has algorithmically determined the finger to be sufficient for being interpreted as intended touch. The implementation of combinations of pen and touch input benefits from the interpretation of the confidence bit data. The tablet driver flags touch points as non-confident which are in the vicinity of the pen interaction location with respect to the interacting hand. When using the right hand for example, the driver primarily separates touches to the right of the pen location, as the probability of them being caused by the palm resting on the surface while interacting with the right hand is very high. The Cintiq 24 HD Touch also offers fully customizable *ExpressKeys* and *Touch Rings*. One common way of using these shortcuts is to assign them to frequently used commands or keystrokes, thus improving the currently involved workflow.

Figure 3.6: Wacom Cintiq 24 HD Touch [@COM]

3.4 Input Techniques

Struman [Str91] investigated the potential integration of whole-hand input techniques and their appropriate use for the control of complex task domains. His taxonomy for the design of whole-hand input enumerates key considerations in the development of such techniques. He divided the input methods into two classes, discrete input methods and continuous input methods. Continuous input methods describe techniques which incorporate different combinations of the degrees of freedom of the hand, while discrete input methods only rely on a subset of the degrees of freedom of the hand. Hand postures are a good example for a discrete input method. An example for a continuous input method would be the kinematically correct mapped transformations of the human hand including motion, orientation and velocity to control a robotic hand. Furthermore, Struman divided both categories into three sub classes, describing the possible interpretation of the whole-hand input in the task domain. These classes are *Direct Interpretation*, *Mapped Interpretation* and *Symbolic Interpretation*. The simplest and most basic one is Direct Interpretation. Here the intervention of the hand is transferred directly into the system, like in the robotic hand example. In Mapped Interpretations, only parts of the degrees of freedom are triggering actions in the task domain, like the up and down moving of a finger to control the intensity of a light. In symbolic interpretations postures and gestures signalize commands to the application to invoke system functions. Here a pre-programmed action is triggered when the system recognizes a specific gesture or posture.

Many of the techniques developed by researchers regarding multitouch or more generally bimanual input techniques can be classified by combining the materials of Struman [Str91] and Zhai and Milgram [ZM98] (see section „3.3 Interactive Displays“). The following subsections describe techniques, which mostly fall in one of the above mentioned categories.

3.4.1 Pen Input

Technically, all desktop applications can be controlled using a pen. The pen seamlessly replaces mouse-based interaction. Both input devices supply a similar fine-granularity on both speed and accuracy, verified by Mack and Lang [ML89]. According to Strumans classification both input devices are allowing direct interpretation. With respect to the continuum of Zhai and Milgram (see section „3.3 Interactive Displays“), the pen is however the more direct input device, because of the higher linkage between the control space and the display space. Besides this superiority, the main reason to incorporate a pen has a background regarding ergonomically benefits. Although the mouse based interaction had great success over decades it introduced health complaints that had never been there before. Users reported to suffer from the so-called *Repetitive Strain Injury* (RSI, also called Mouse Arm Syndrome). RSI describes a condition when body parts, like the fingers, hand, arm and shoulders, have been injured from repetitive movements and static load [@Asi]. In the domain of computer usage this is typically caused by repeated movements and actions using conventional input devices like mice or keyboard. The reason for this is an uneven distribution of the muscular tension when using mice as input devices. The use of a pen distributes movements and actions over a variety of muscles in the fingers, hands or arms thus preventing forearm twisting, which typically results in unnatural arm and hand postures and strains muscles and tendons.

A study by the TU Darmstadt investigated how far a pen is an ergonomic alternative input device [@Eur]. Through a simulation of typically demanded computer work like office tasks, they conclude that the pen has proven itself as superior regarding ergonomically comfort compared to the conventional mouse interaction. After a short time users were already able to achieve an enhanced performance in everyday computer work while simultaneously their interaction was less error prone. Figure 3.7 gives two examples of the natural muscular tension caused by pen usage compared to the unnatural muscular tension caused by mouse interaction.

These facts are also acknowledged by leading software companies from various industrial domains. There are plenty examples of software, explicitly designed or extended to allow interaction using a stylus or pen, which demonstrate how important the aspect of an ergonomically designed workspace is. For example, Adobe Photoshop [@Pho] is explicitly designed to take advantage of the pen-pressure sensitivity, pen-tilt sensitivity

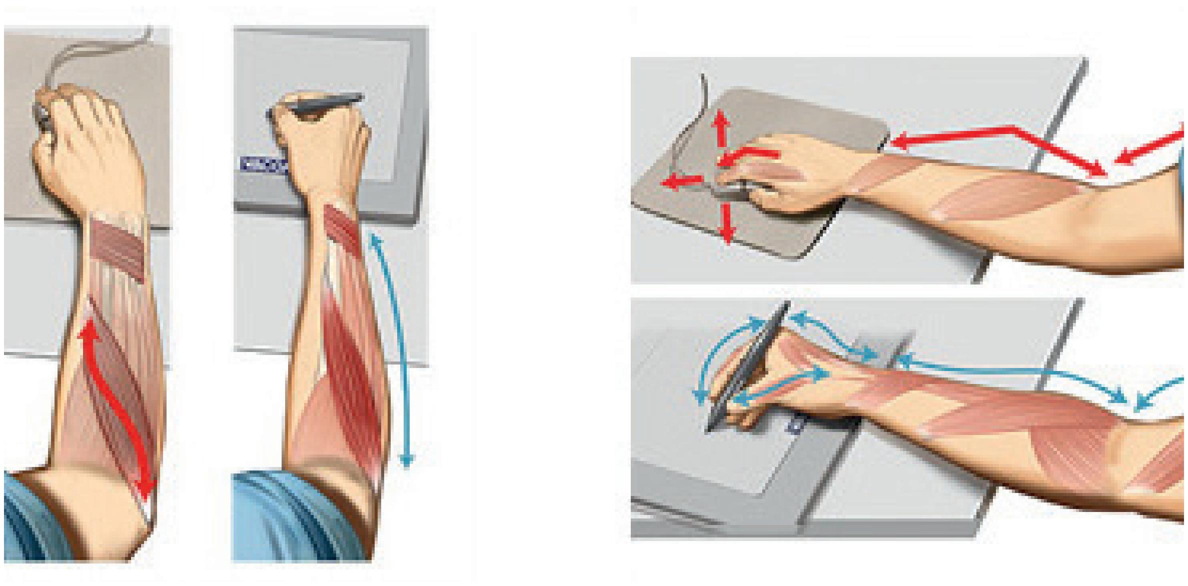


Figure 3.7: Uneven distribution of the muscular tension (red arrows) compared to the natural muscular tension (blue arrows) [@Wacb]

and productivity features of Wacom's pen tablets and interactive pen displays. Adobe Photoshop offers a variety of tools that can be used for drawing and painting interaction, requesting the fine-granularity and also the ergonomically benefits of a pen. Combined with a Wacom tablet, Adobe promises a faster workflow and a more natural interaction process.

Maya 2013 [@May], developed and maintained by Autodesk, another leading software company in the domain of industrial 3D digital art, also incorporates the use of Wacom digitizer tablets. With the latest upgrade, pen-enabled toolsets are introduced to add functionality that helps to facilitate parallel workflows and complexity handling.

These are all examples clearly demonstrating for which tasks the use of a pen and a digitizer tablet is beneficial. Typically, these are high precision tasks that are requesting the fine-granularity of a pen. It is obvious that for this kind of interaction, ergonomically designed equipment is desirable. This enables usage that is natural to human anatomy, especially regarding hand and finger movements, where a decrease of stress, strain, fatigue and discomfort improves work efficiency.

In section „2.3 Seismic Interpretation“, the workflow of horizon tracing using seismic interpretation software was presented. This workflow seamlessly integrates into the area of high precision tasks and is suitable for the use of a pen. With the release of version 4.2.0 OpendTect became the first seismic interpretation system to support Wacom digitizing tablets [@Opea]. Especially the horizon and fault interpretation workflow had been adapted to benefit from the usage of a pen in combination with a digitizer tablet.

All of the presented applications primarily utilize the pen capabilities of Wacom digitizing tablets as solely input modality. However, consumer software products that already incorporate the pen and touch capabilities of Wacom digitizer tablets already exist. Two sample applications, which have a similar setup like the setup used in the development of this thesis, will be presented in section „3.6 Applications“.

3.4.2 Multitouch Input

The previous section showed that the industry already recognized the benefits of pen input in suitable professional task domains. At the moment of writing this thesis, however, there is a lack of consumer software products for professional-level tasks, which incorporate multitouch interaction. Professional digital content applications are still reliant on conventional mouse and keyboard interaction. This interaction often requires digital creators to perform several sequences of steps to fulfill their task. However, various research efforts exist investigating the integration of multitouch interaction to enhance workflows of desktop applications or even a full replacement of the traditional interaction. The following sections present some examples.

Rekimoto [Reko2] presented with SmartSkin two prototype applications to evaluate new interaction techniques. These techniques are based on various different approaches using a vision-based touch sensing hardware. There is single touch, multitouch, palm or whole arm recognition to create new input techniques. For example, users are able to mimic mouse-based interaction using single touch. While this is a common practice in touch-based applications, SmartSkin also allows simulating mouse-over states by sensing finger positions while they are floating above the surface. For this sake the distance between the hand and the surface is measured. Respectively, this allows mouse pressed and released interaction. The system does not only recognize multitouch but also multihand and therefore supports collaboration. Furthermore, with the use of the 2D positions of touches, users are able to manipulate and select objects using single touch and multitouch interactions. The so-called »shape-based manipulation« allows completely new interaction techniques, which have a close relation to interaction with physical objects. For this kind of interaction the position data of the touches is only of secondary importance; much more of interest is the posture of the forearms on the surface. With respect to the classification of Struman [Str91] this is a continuous interaction technique allowing direct interpretation that was not discovered so far. With laying down one or both arms on the surface, various rules of object manipulation can be realized. For example, objects can be moved or selected by sweeping one or two arms over the surface. In the second setup of SmartSkin, a tablet-based prototype allows a more accurate determination of the position and the shape of the hand. This platform allows interacting using commonly known multitouch gestures like panning and pinching with two fingers. In this setup it is also possible to identify physical objects and to use them as tangible user input devices. Moreover, this system utilizes palm recognition for beneficial use. When the system recognizes a palm a context menu is opening and offers new menu items.

In another study Hinckley et al. [HCS98] used bimanual interaction in 2D map navigation tasks for unified zooming and panning. They presented several variations of two-handed navigation techniques with a combined use of a self-built touch sensing capable mouse and a touchpad or a stylus in combination with a puck. Although they incorporated different input devices in their experiments, the mapping of the interaction techniques was pretty similar in each of these. They choose a clear distinction of the input modalities to accommodate to non-dominant hand use or dominant-hand use. They mapped the interaction of the NDH primarily to manipulate the camera view of the map, while the DH was used for annotating tasks. Combining interactions from NDH and DH are used to create further interaction techniques. For example a »stretching and squeezing« of the map view was initialized when clicking with the puck using the NDH and simultaneously pressing down with a stylus using the DH.

The work of Wu et. al [WSR+06] illustrates a set of design principles for constructing multihand gestures on interactive surfaces. In their system, conventional point based input and freehand gesture input co-exist. There are three basic principles that should lead to a generality in bimanual interaction. These are gesture registration, gesture relaxation and gesture reuse. Gesture registration is the invocation point of each interaction. A particular gesture must be recognized by the system to set the further context of the application. Once the gesture is recognized, the context of subsequent interactions is manifested. For example, when two fingers are placed on the tabletop, the application recognizes this and sets the state of further interactions. Now placing a pen stylus will be assigned to a writing tool, allowing annotating objects. Although the invocation gesture must be kept alive over the whole interaction time, gesture relaxation gives scope for little modification of the invocation hand gesture taking away the burden from the user to maintain muscular tension. Once entered the annotating mode, the invocation hand gesture can be changed for example by laying down the whole hand on the surface. The application will still remain in the annotation mode. Gesture reuse allows using the same gesticulative input to perform different interactions depending on the mode the application entered through the gesture registration phase. Thus, establishing a reference to Strumans [Str91] classification, this interaction technique can be classified as continuous interaction with mapped interpretation.

3.4.3 Multimodal Input

The work of Hinckley et al. [HYP+10] grounds on the adoption of a division of labor between direct pen input and direct touch input. Based on the work of Guiard [Gui87] the goal of their research is to define the logic of division of labor between pen input and touch input in the domain of user interface design. Their design consideration is summarized by

> *»THE PEN WRITES, THE TOUCH MANIPULATES AND THE COMBINATION OF PEN + TOUCH YIELDS NEW TOOLS.«* [HYP+10]

They include considerations based on a complementarity of both modalities rather than treating them isolated from each other or assigning one of them to be the more preferred one. However, an explicitly distinction between pen and touch is indispensable because interchangeability would lead to ambiguities verified by other research efforts [FHD09]. The clear distinction between pen and touch is realized by assigning one of two core features of the application to each of the two modalities. Object manipulation, object selection and zooming are realized through touch interaction, while the pen is primarily used for inking. In this approach several combinations of pen and touch techniques are realized. For example, a user can group items into a stack by tapping items. Now, while holding one of the selected items and tapping the same item, this time with the pen, the selected items are piled on a stack. Another pen and touch gesture is realized through the selection of an item by touching it and crossing the interior of the object with the pen, starting and ending from the outside of the object. This results in a cutting of the object along the line from the starting point to the endpoint of the pen interaction. An interesting gesture is the so-called *Carbon Copy*, where selecting an object with the finger and then dragging off with the pen creates a copy of the object. Frisch [Fri12] reused this gesture to create a copy of a diagram node.

The conclusion of this approach is that each modality has its benefits in some situations, where other modalities might have drawbacks. Mapping this conclusion to the real world, one would easily agree that signing a contract is always more comfortable with a pen, whereas igniting a lighter will always be easier using the fingers. This concept is also true for the interaction with virtual objects and tools in the domain of user interface design of desktop based applications.

This argument is also the motivation of an exploration of Brandl et al. [BFW+08]. They provide intuitive input mappings as well as clear means to enable real time learning of more advanced input techniques. In this exploration three different ways of bimanual input are compared with each other. These different ways are pen/pen, pen/touch and touch/touch input. The Kinematic Chain Model defined by Guiard [Gui87], again serves as a starting point for their design considerations. With this model as a foundation, the work of Brandl et al. also argues for a division of labor between the two input modalities. They are convinced that a combination of DH pen input and NDH touch input has the potential to simulate the real world use of both modalities. In this manner, the user requests the fine-granularity of the pen while drawing but simultaneously obtains direct haptic feedback of the bare hand touch input. A sketching application has been developed for evaluation of their bimanual interaction techniques. In a formal experiment users were asked to navigate through a virtual maze game. They mapped the drawing of the completed way to the dominant-hand, whereas zooming and moving the view was mapped to the non-dominant hand. By measuring the time needed to complete traversing the maze and counting the number of errors occurring while doing so, a superiority of pen/touch input compared to touch/touch or pen/pen input was proven. The results of this experiment clearly demonstrated that in the pen/touch setup users were able to complete the task faster and with less errors than in the other two setups. Further insights were provided by the analysis of the lifting number of the dominant hand. In this measurement, the pen/touch setup also yields the most positive values, since the lifting operations were rarely conducted. This results in a smoother workflow of bimanual input modalities when using a division of labor between the DH and the NDH, due to the less interruptions of the DH interaction. As a consequence, the cognitive overhead can be reduced.

In his work, Yee [Yee04] presents a two handed system that places both hands in the same reference frame to measure the benefits of asymmetric interactions. With the help of a self-built setup of commonly used touchscreens and tablet screens, he presented three applications, which allow interacting using simultaneous touch input and pen input. At first, a sketch application mapped drawing and annotating tasks to the DH using the pen, while the NDH allows the navigation of the drawing canvas and execution of system control tasks such as changing the drawing color of the pen. Secondly a zoom viewer application was presented. Here unimodal pen or unimodal touch only is used to accomplish the same interaction of grabbing or scrolling a canvas. When both input modalities are simultaneously detected by the system, each of them is assigned a different role. The two input modalities together incorporate rotating and zooming gestures of the drawing canvas. The touch becomes the anchor of the pen movement for rotating and zooming the view. Lastly, Yee integrated the novel interaction techniques in a file browser application to enhance drag and drop interactions. While the pen is used for selecting and holding the dragged object, the touch is still able to navigate, scroll and rearrange windows for reaching a corresponding drop target. An informal user study revealed that these interaction techniques are »natural« and »fast«. However, Yee also observed some drawbacks of bimanual interaction. Using two hands on the same screen will lead to occlusion of display parts. Regarding the form factor of the display this may cause the main portion of the display to be occluded. This is mostly true for mobile devices, like tablet computer or smartphones. Furthermore, the simultaneous interaction of pen and touch could lead to collision of both hands. A careful layout of the interface could be a solution to this. From a technical point of view, the palm rejection is still a challenging and unresolved task. When using both hands and distinguishing between touch and pen input, the resting palm of the pen hand may be interpreted as intended touch and cause interference with the interaction.

The work of Kurtenbach et al. [KFBB97] introduces a new paradigm for two-handed input in graphical user interface based applications. The basic components of their design considerations are two-handed input, realized through the use of tablet and the introduction of a semi-transparent marking menu. Their GUI paradigm has three main design goals. First of all, the system should maximize the usable amount of screen space. Secondly, the impact to a user's visual attention must be minimized and thirdly, by providing continuous input for the NDH, an increase of the degrees of freedom for the interaction should be guaranteed. They developed *T3*, a graphical editor for drawing simple 2D shapes. Although their system does not provide direct bare hand input, their design also distinguishes between the role of the DH and the NHD and can easily be compared to applications from this domain. The role of the DH is the direct input and manipulation of the currently selected tool. The NDH moves a semitransparent tool glass menu. When the buttons of each puck are pressed, an incorporated rotating gesture of the currently drawn shape is realized.

The presented interaction techniques make clear, that multimodal input, when carefully chosen, can enhance productivity and yield to more efficient workflows. The majority of the presented research studies vote for an asymmetric division of labor between the DH and NDH. With respect to the capabilities of pen input and touch input, this separation is making sense. The interaction techniques implemented in this thesis are based upon these considerations taking the most benefits of those into account.

3.5 Mode Switching Techniques

In traditional graphical user interfaces the mouse is still used as primary input device through standardized one-point interaction techniques. This limitation has a high influence on the capabilities of parallelism in the task domain of applications. To overcome these limitations, mode switches are usually introduced. These mode switches change the way the mouse input will affect the digital content of the application. This is traditionally accomplished through selection of a menu item or by pressing a menu button. Depending on the task this may result in a repetitive long traversal between spatially separated elements, thus decreasing workflow productivity. Although this has been the usual workflow over decades of human computer interaction, this kind of interaction obviously disrupts the natural workflow. However, for a consistent and applicable workflow, mode switches are needed. Therefore a variety of research efforts focus on how to improve mode-switching techniques to allow for mode transitions, which are smooth, consistent and minimally disruptive. Some of the research studies investigated novel approaches based on Graphical User Interfaces, while others used gestural commands. The following paragraphs presents examples of each approach.

Graphical User Interface based

Li et al. [LHGL05] conducted a quantitative analysis of techniques for switching between ink and gesture mode in pen interfaces. They investigated five different techniques, each of these based on the usage of different parameters of digitizer tablets. In addition to the pen position, digitizer hardware increases the space of available input by supplying attributes such as pen-tilt, pen-pressure or different buttons attached to the pen itself. Li et al. designed an experimental task by simulating menu commands through the usage of crossing keystrokes on a pie slice. The implemented techniques included standard interaction in pen-based applications, like pressing a button of the pen to invoke a command, using the pen-pressure or a press and hold gesture of the pen on the tablet. They also investigated bimanual interaction by employing a graphical user interface component; the so-called *mode switching button* at the corner of the tablet PC. A similar technique is described by Kin [Kin12], who investigated bimanual interaction for constructing virtual organic environments. The NDH sets the further context of the DH interaction by invoking a command on the mode switching button. The results of Li et al. showed that pressing a button with the NDH provides the fastest performance, while holding the pen in a stationary position delivers slower and more error prone interaction.

These results are similar to those of Lepinski et al. [LGF10], who implemented and evaluated multitouch marking menus. Mode switches are realized by sliding gestures to select a category of a marking menu. By comparing measurements of movement time, articulation time and error rates, a significantly better performance of multitouch marking menus in contrast to traditional hierarchical menus was proven.

Building upon the work of the T3 system by Kurtenbach et al. [KFBB97], researchers at Microsoft Research demonstrated *in-place commands*. In-place commands are popup menus invoked by touch interaction and are positioned next to the invoking finger [@Mica]. Since the menu is only displayed when invoked through the according command, the main advantage is their aptitude to work on any kind of display and thus not to be constrained by form factors. For example, on large displays, it can be cumbersome to select commands from a menu or toolbar located at the edge of the screen [LGF10]. The demonstration of the Microsoft Research elaboration included bimanual and multimodal interaction techniques for sketching applications. Their design consideration is also influenced by the Kinematic Chain Model defined by Guiard [Gui87] and therefore a distinction of labor between touch interaction and pen interaction was chosen. While the pen primarily is used for sketching interaction, the touch interaction invokes an in-place menu. The appearing menu offers a tool palette, which influences the role of the pen. To allow more differentiated tool palettes, the menu supports gestural activation, which is desirable to enable users to interact

»AS NATURALLY AS THEY WOULD INTERACT WITH OBJECTS IN THE REAL WORLD.« [JGH+08]

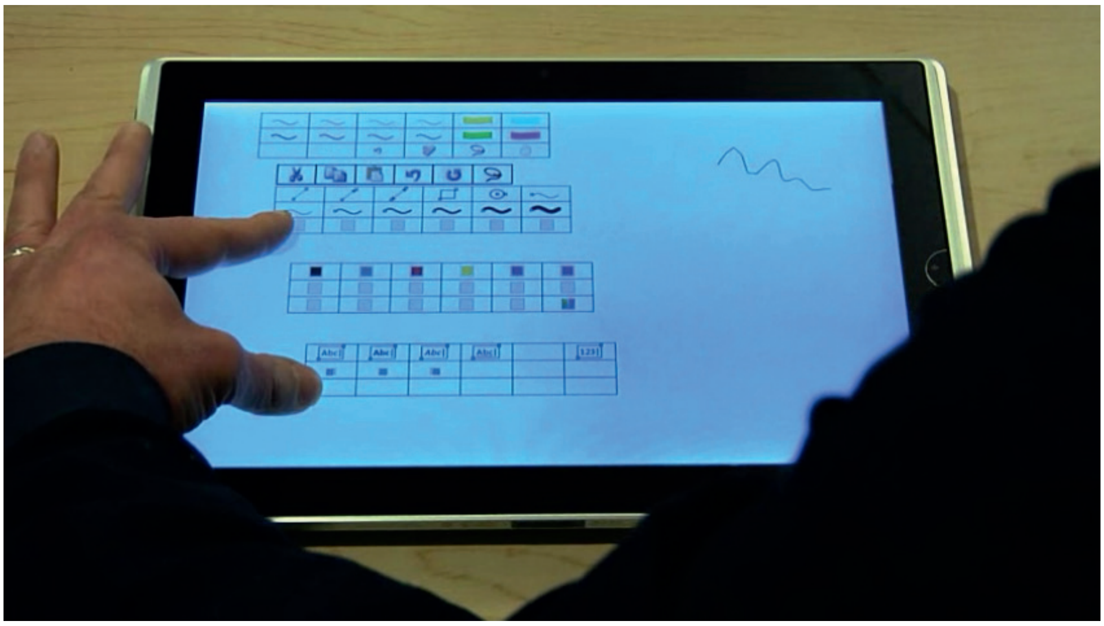
For this purpose the menu is expandable by using a pinch gesture. Figure 3.8 illustrates an expanded menu, offering the possibility to select the current tool palette by selecting it with touch interaction. While selecting one of the palettes the thumb must rest on the surface to indicate the selection mode of the menu.

Gesture based

The approaches of Song et al. [SBG+11] and Sun et al. [SCS+11] are similar to the pen-press technique of Li et al. [LHGL05] (see section „3.5 Mode Switching Techniques“). However, they developed a multitouch pen capable of detecting touch gestures on the stylus itself. Mode switches are realized by detecting these gestures directly on the stylus and the way users are holding the pen.

Wu et. al [WSR+06] use the advantage of gesture reuse to implement more usable systems. Based on their recommendations, Frisch [Fri12] also reused gestures in his investigation of bimanual interaction for visualization of node link diagrams. The gestures used in these elaborations are executed with the NDH to instrument the application of the mode switch. For example, Frisch used a simple holding gesture of the NDH on the background of a

Figure 3.8: In-place menu expanded through a pinch gesture [@Mic]



tabletop application to simulate a mode switch. This approach is similar to the techniques described in the following section. While these are simple gestures, which are technically easy to recognize, Wu et. al [WSR+06] described the gesture registration phase, which allows more complex gestures (see section „3.4.3 Multimodal Input“). In this phase the application explicitly waits for gesture input to set the further context of the interaction.

3.6 Applications

Section „3.3 Interactive Displays“ showed that the capabilities of currently available interactive displays are laying the foundation for new multimodal interaction techniques. Although these possibilities are not entirely used by currently available professional consumer software, there are already applications incorporating new features based on these capabilities. Two of these will be presented here.

With the release of service pack 3, Autodesk Mudbox 2013 [@Mud] supports multitouch devices like the Wacom Cintiq 24 HD Touch. This digital painting and digital sculpting software enables digital creator’s production-ready 3D digital artwork. Service pack 3 focuses on enhancements regarding performance and professional workflow quality to help creators achieving higher productivity regarding their digital 3D artwork. In Mudbox 2013 a distinct mapping of the input modalities was chosen. Single touch or multitouch interaction is only used for navigation task of a 3D canvas. Multifinger gestures are used to trigger shortcuts in a more natural way by simply touching the screen, thus removing the dependence of pressing keys on a traditional keyboard. For example, holding down four fingers on the screen will trigger the same system event as pressing and holding the »ctrl« button on the keyboard. Interaction using the pen is the only possible way to manipulate and create 3D digital artwork.

Another leading software company released a professional consumer application, which also incorporates the novel capabilities of Wacom digitizer tablets. Corel Painter 12.2 offers a more reality-based interaction workflow using multitouch technology [@Cor]. The interaction techniques implemented in Corel Painter are similar to these in Autodesk Mudbox. The application allows navigating the drawing canvas with precision and eases using finger gestures, realizing simultaneous pan, rotate and zoom gestures in a responsive way. The pen is – in this piece of software – the only way to interact with the digital content.

Although these new workflow enhancements are providing a promising solution for integration in a reality based professional application workflow, these new features do not contain any new interaction techniques regarding bimanual interaction with different input modalities. However, the multitouch navigation techniques developed during this thesis are similar to those used in Autodesk Mudbox.

04 // Computer Aided Engineering Applications

4 COMPUTER AIDED ENGINEERING APPLICATIONS

Beside the implementation and evaluation of bimanual interaction techniques for seismic interpretation, another integral part of this thesis is their integration into an existing seismic interpretation tool. This tool is called OpendTect [@Opeb], an open source software system that provides a complete computer-aided seismic interpretation environment. Before the details of the system itself are described, a more general look on the topic of computer-aided techniques is necessary to understand the underlying principles for a successful integration of novel interaction techniques into applications from this domain. Thus, this chapter will introduce all concepts of Computer Aided Engineering (CAE) applications, starting with the definition of the term itself, followed by the specific application components, which played a key role during the development of this thesis. This includes an analysis of the user interface (UI) and interaction techniques, which are typically included in this kind of software. In the end, OpendTect itself will be put in the context of CAE applications. For this, all components of OpendTect, which are classifying OpendTect as CAE application, will be described. Furthermore, the system architecture and all external libraries are described in more detail.

4.1 Definition

Computer Aided Engineering (CAE) is a term used in a wide range of industries describing the use of computer software to simulate performance in order to assist in the resolution of engineering problems [@Sie]. While CAE applications support a broad diversity of engineering disciplines or research phenomena, the majority of these applications typically provide a visualization environment including simulation, validation and optimization of products, processes and manufacturing tools. CAE software is a type of computer programs that replaces manual drafting and prototyping with an automated process, which is based upon pre-processing, problem solving, and post-processing steps. The pre-processing phase comprises visualization steps like modeling the geometry and the physical properties of the investigated scientific object in compliance with the applied loads and natural constraints. In the following solving phase, mathematical calculations of the underlying physics yield the results for the final post-processing phase, in which engineers review and analyze these results.

Benefits of such a process chain include improved product quality and durability accompanied by reduced product development cost and time. Based upon their impact on performance, design decisions can be made with respect to their expense. Computer-aided techniques support engineering teams in calculating risk and reveal a better perception of the performance implications of a design. Moreover, the ability to effectively leverage performance insights and improve designs for contribution to a broader community is realized by integrating CAE data and process management. Money and time are saved by evaluating and refining designs using computer simulations rather than physical prototype testing.

Popular examples of CAE applications are Autodesk Maya [@May] and AutoCAD [@Aut], both from the CAE subdomain Computer-Aided-Design (CAD). While Maya offers a feature set for computer animation, modeling, simulation, rendering and compositing, AutoCAD provides features for computer-aided-design and drafting to visualize concepts through animations and photorealistic rendering and it simulates how a design will perform in the real world. Besides these examples, there are various other applications from domains like modeling, visualization, analysis or interpretation and even computer gaming. However, although these examples are from different domains they all share common prerequisites according to their user interface design and to their provided interaction techniques. The majority of these interactive computer graphics systems visualize 3D data on conventional desktop computers; thus their interaction depends on conventional desktop input devices. Analysis, exploration or manipulation of 3D data, however, makes it necessary to provide 3D interaction techniques. Conventional input devices, like the mouse only capture two DOF simultaneously, making additional interaction techniques for 3D manipulation necessary. One approach that is used in various CAE applications is the separation of degrees of control [BKLP04]. This means the decomposition of a high DOF interaction in a sequence of multiple lower DOF interactions.

4.2 User Interface

A software front end is called Graphical User Interface (GUI) [SS05]. Communication between a user and the computer takes place through interaction with the GUI. The main characteristic of the Graphical User Interface is that it realizes the visual linkage between human interaction and visual feedback, which replicates the experience of humans from the physical world. Dividing the GUI into two major parts is a common design of CAE applications. Saxena and Sahay [SS05] call these two parts *Graphics Window* and *Command Window*. Figure 4.1 illustrates a generic view of CAE applications, which is shared by most CAE applications.

The Graphics Window is the visual manifestation of the currently manipulated data or object and provides visual feedback of all executed transformations. The Command Window allows communicating instructions to define how the selected data should be

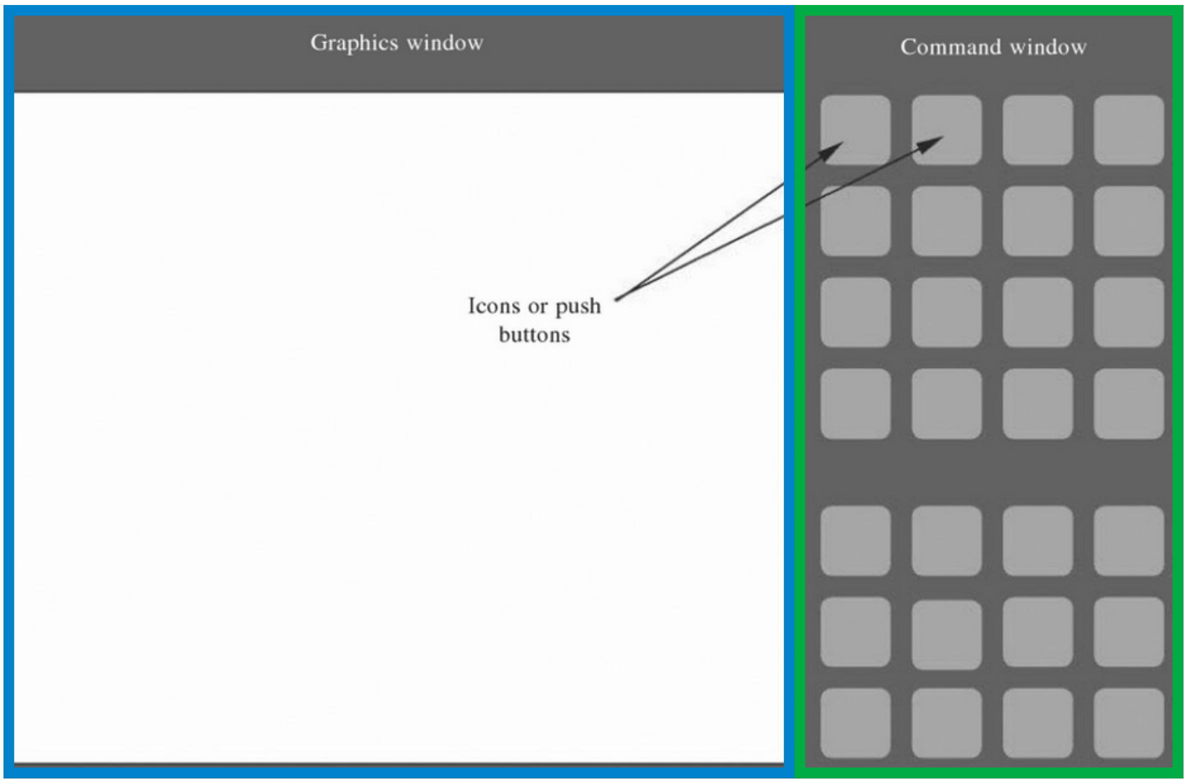


Figure 4.1: Generic appearance of CAE application, illustrating the Command Window (blue rectangle) and the Graphics Window (green rectangle) [SS05]

modified through the use of 2D interface controls (e.g. push buttons, sliders or input fields). A common way to minimize the cognitive overload of numerous 2D interface controls is to group them according to their functionality. Another common technique of separating the degrees of control is to provide several different views of the same 3D data to support the mental visualization task, when examining 3D data. Multiple views externalize this visualization task by providing information for a rapid exploration of the solution space. Thereby, multidimensional input is broken up into various 2 dimensional inputs. A user is able to manipulate two DOF simultaneously in each provided view. Furthermore, there are conventional sliders realizing one DOF controls, like rotating around one of the principal axes. Input fields realize an even more precise way of positioning and orientation tasks, by providing a way of directly typing in values for each transformation. Figure 4.2 shows the Graphical User Interface of Autodesk Maya [@May], a concrete example of a CAE application that inherits the commonly shared GUI layout of CAE applications.

4.3 3D Interaction Techniques

Interaction techniques are methods used to accomplish a given task via the user interface [BKLP04]. The previous section illustrated that the interface of CAE application differs

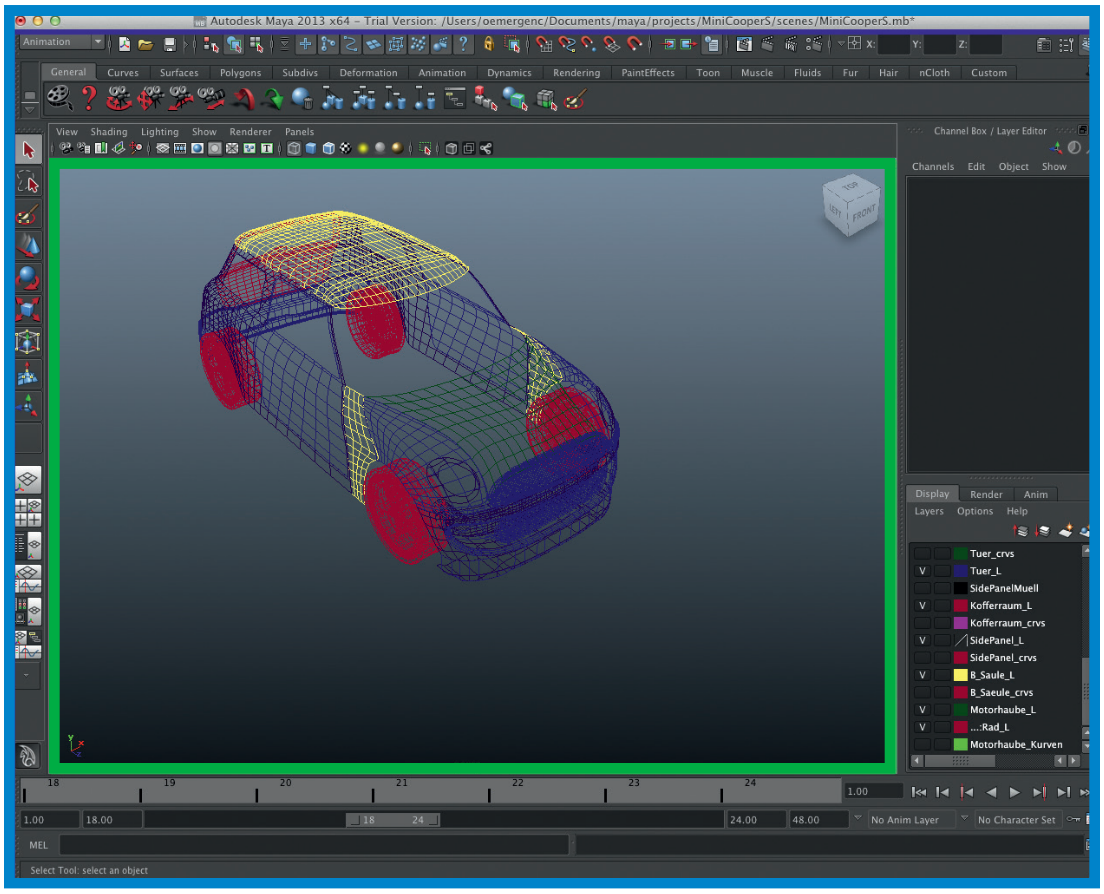


Figure 4.2: User interface of Autodesk Maya [@May], showing the Graphics Window and the Command Window

from that of conventional desktop applications, such as office packages. Interaction with 3D data on conventional desktop computer makes 3D interaction techniques necessary. This section introduces interaction techniques that became a de-facto standard in CAE applications with focus on important techniques for the implementation of this thesis. Note that the list of interaction techniques mentioned here is far from complete. For more insights regarding interaction techniques please see [BKLP04].

4.3.1 Selection and Manipulation

This section gives a brief description of techniques used to manipulate and select 3D objects in CAE applications. The manipulation of 3D objects requires different Control-Display mappings (see section „3.3 Interactive Displays“) of the captured user input as compared to traditional applications. The most common techniques will be described in the following.

3D-Widgets

The term 3D-Widget refers to a visual representation of an encapsulated geometry and behavior to manipulate application objects [CSH+92]. This visual representation supports the perception of object manipulation during interaction. Besides these visual cues, there are so-called hidden active areas, which provide a way to directly manipulate single transformations of object properties. Houde [Hou92] determined the most suited locations for these controls with the introduction of the bounding box, a box surrounding the extent of the included object. Whenever a user attempts to select the interior object, the surrounding bounding box appears and offers direct manipulation though active areas. Because users had difficulties in perceiving these active areas, a visual exposure of these areas seemed to be the appropriate solution. Figure 4.3 illustrates four representations of active areas investigated by Houde [Hou92] to measure their applicability in direct 3D manipulation. The abstraction of the interior shape accomplished through the surrounding cuboid bounding volume shown in Figure 4.3b proved to be a promising solution of providing an intuitive way of manipulating different shapes. However, one drawback of the bounding box concept is the lack of visual cue for the resulting manipulation. Introducing narrative handles, which users identified as interaction possibility, solved this. The cubes in Figure 4.3c are an example of these narrative handles. Although users were immediately able to identify available functionality of the cubes, some were confused about in which way manipulating the cubes would affect the object (translation, scaling or both). This ambiguity was solved by replacing the cubes with »Hand shaped handles« [Hou92] (Figure 4.3d).

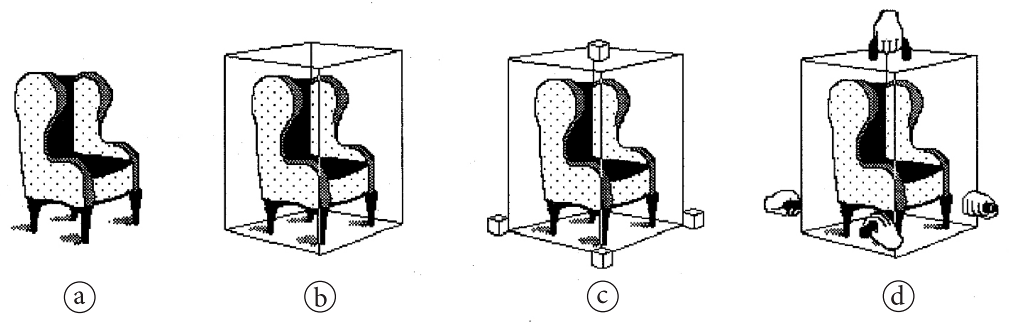


Figure 4.3: Different visual representation of handles, each indicating narrative interaction capabilities [Hou92]

Although today's CAE software does not contain the representations of 3D-widgets presented above and is using further developments and expansions, the core principles and techniques are still the same. These widgets are included in the standard equipment, not only of professional CAE applications, but also of more immersive 3D environments (e.g. virtual environments). Their main advantage is a seamless transition between different manipulation sequences accomplished by the introduction of direct controls in the 3D scene.

Each widget has a sole and clearly defined responsibility during object manipulation, thus reducing the resulting difficulties of a correctly perception of the position and orientation of 3D objects when only using a flat 2D screen view. Furthermore, the visual feedback during manipulation of object transformation is essential. While moving the mouse cursor to accomplish 2D positioning tasks (e.g. drag and drop operations) on a computer display is similar to the experience of moving objects in the physical world, interacting with 3D objects on a computer display is usually foreign to users. The reason for this is the way the 2D mouse movement is mapped into the 3D scene, as this constraint does not replicate the movement of real objects. For this reason, direct visual feedback of manipulation of 3D object properties is needed. Figure 4.4 illustrates the *SoTransformManipulator*, a 3D-widget of the graphics library Open Inventor [@Sil]. After selecting the 3D shape, the manipulator becomes visible. The visual representation of the manipulator itself changes according to the selected transformation, providing a clear visual cue for the currently modified object parameter. This enables a predictable and feasible interaction even for novice users.

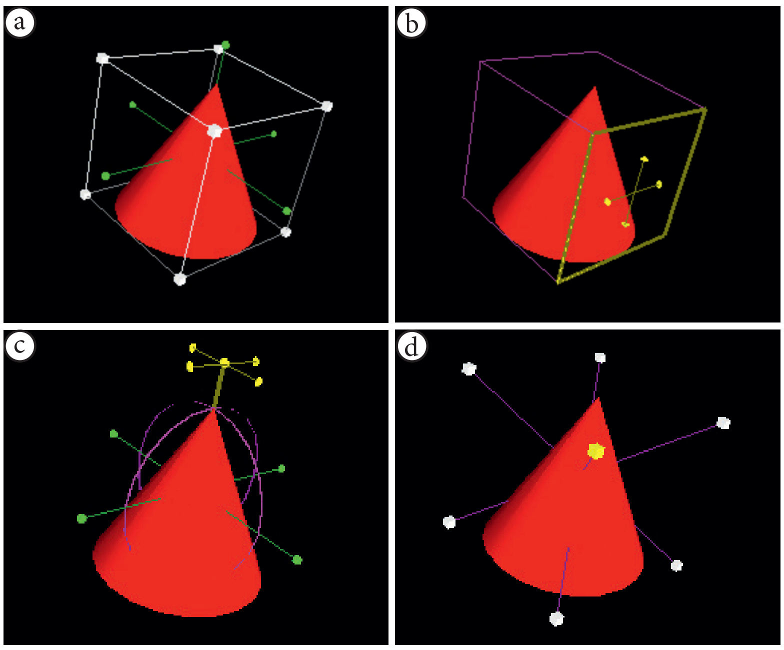


Figure 4.4: SoTransformManipulator of the graphics library Open Inventor. The visual appearance of the manipulator changes according to the selected handle, a) Default appearance, b) Translation, c) Rotation, d) Scaling

Virtual Trackballs

Rotation of three-dimensional objects by a two dimensional mouse is a typical task in CAE applications. The most commonly used rotation technique is a *Virtual Trackball* surrounding the object and operated by the mouse pointer. There are several variations of a Virtual Trackball, but all of them are using a similar approach to provide a natural and intuitive way to rotate objects in a 3D space. Basically, all approaches are placing the object of interest at the center of an imaginary glass ball, which can be freely rotated around any arbitrary axis in 3D-space. When a user clicks and drags the object of interest with the mouse cursor, the software interprets this as pushing and »nudging« [BKLP04] the Virtual Trackball. In turn, the rotation of the Virtual Trackball is mapped back to the object of interest. A user's perception is often complemented through visualizing the Virtual Trackball using a circle or sphere encasing the object of interest on the display screen. In the following, two state-of-the-art rotation techniques will be discussed in more detail. These are the *Virtual Sphere* technique and the *Arcball* technique. For further techniques of Virtual Trackball derivatives please see Hendricksen et al. [HSH04].

Virtual Sphere The Virtual Sphere was introduced by Chen et al. [CMS88] and can be described as a 3D sphere located behind the viewport [BRP05]. Here, a common technique is used to get corresponding 3D points underneath the 2D viewport coordinates of the moving mouse cursor. For this sake, the 2D viewport points of the mouse cursor are projected onto the sphere‘s surface using a ray, pointing in the direction perpendicular to the viewport plane (usually depth or z-axis). This enables a mapping of the 2D motion of the mouse cursor to a 3D rotation from the mouse down location to the current mouse location. Rotation of the object is accomplished by an imaginary pushing of the Virtual Sphere, which in turn replicates the behavior of a physical trackball. While the direction of the rotation is congruent to the direction of the cursor movement, the axis, the Virtual Sphere is rotating around, is defined by the axis perpendicular to this direction (see Figure 4.5). Respectively, rotation about the x-axis (usually viewport width) and y-axis (usually viewport height) are accomplished by nudging the sphere around its center point from the top to the bottom or from the left to the right. However, with this technique a rotation around the z-axis is not possible. Nevertheless, rotation about this axis is realized through mouse movement on the circumference of the enclosing circle or with mouse movement completely outside of this circle.

Arcball The Arcball, presented by Shoemake [Sho92], is basically an improved version of Chen et al.'s [CMS88] Virtual Sphere. This technique also utilizes the projection of 2D mouse coordinates onto a sphere in 3D space to calculate the corresponding 3D coordinates. The main difference between these techniques is in the mapping of the 2D viewport points of the mouse cursor into corresponding 3D coordinates of the Virtual Trackball. The Virtual Sphere interprets the movement of the mouse cursor as tangential vector from the point of the mouse down position and calculates the rotation angle accordingly to the mouse movement. In contrast to this, the Arcball maps the 2D mouse cursor points on a circle segment of the Virtual Trackball, thus every 2D screen point has a mathematically defined rotational value. A user is able to change the orientation of an object on the screen, through well-known mouse dragging operations. The resulting rotation angle is computed by the

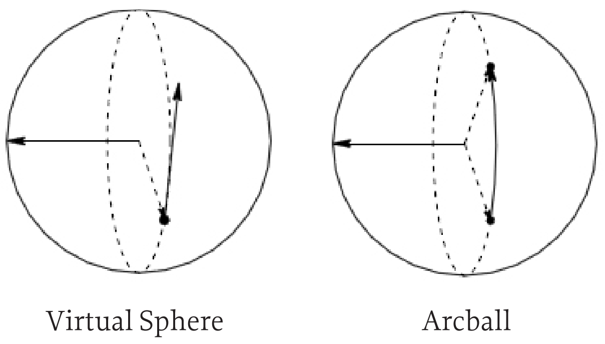


Figure 4.5: While the Virtual Sphere (left) interprets the mouse movement as tangent vector, the Arcball (right) maps the mouse movement on a circle segment [@Mul]

algorithm of the Arcball as an arc on the screen projection of the virtual trackball, defined by the mouse movement between the mouse down and mouse up positions. Figure 4.5 illustrates the underlying concepts of each technique.

Hinckley et al. [HTP+97] investigated the usability of 3D rotation techniques and identify two main differences between the Virtual Sphere and the Arcball technique. The first is the drawback of the Virtual Sphere caused by »hysteresis effects«. Shoemake [Sho92] describes hysteresis as the result of irreversible interaction. A user is not able to »undo« a transformation by reversing the order of the sequence to accomplish the transformation, which causes uncertainty in interaction. The second is a fixed Control/Display[1] (C/D) ratio. For example, moving the Arcball across its inner circle, results in a 360 degree rotation of the virtual object. The same nudging of the Virtual Sphere would only result in a 180 degree rotation of the virtual object. Once a user has understood the underlying principles of the Arcball, she or he is able to change the orientation of an object in any direction with a single action. This makes the Arcball superior compared to the Virtual Sphere, where changing the orientation of an object may require the user to compose multiple rotations. In addition, the resistance against hysteresis effects makes the Arcball technique referred to be the

»BEST KNOWN 2D TECHNIQUE FOR 3D ROTATION.« [HTP+97]

4.3.2 System Control

System control provides a way to allow a user to specify how input will affect the digital content of the application. Bowman et al. [BKLP04] define system control in terms of issuing a command with one of the following purposes:

1. Request the system to perform a particular function

[1] Control/Display ratio: The ratio between the amplitude of the hand motion (Control) compared to the amplitude of the cursor movement (Display) [LBE04]. A high C/D ratio results in low sensitivity of control and vice versa. Low C/D ratio (high sensitivity) is suitable for fast movements like approaching a target, while high C/D ratio (low sensitivity) could be helpful in fine adjustments.

2. Change the mode of interaction
3. Change the system state

Note that a user only requests an action to be executed and leaves the details of the execution to the system, rather than determining these details on her or his own. Saving a document is a suitable example of a well-defined system control task (1.), which is executed by the system itself on request. Mode switches (2.) are conventionally requested through the selection of an according menu item or toolbox item (see section „3.5 Mode Switching Techniques"). Their purpose is to change the way an input stream will affect the application. An example of changing the system state (3.) is switching between different applications, causing subsequent interactions to be interpreted only by the most foreground application.

In conventional 2D interfaces, interaction elements such as pull-down menus, pop-up menus, push buttons, radio buttons, checkboxes etc., are examples of system control techniques, which results in a broad diversity of system control tasks. This makes a more precise classification indispensable. Although the classification of Bowman et al. [BKLP04]

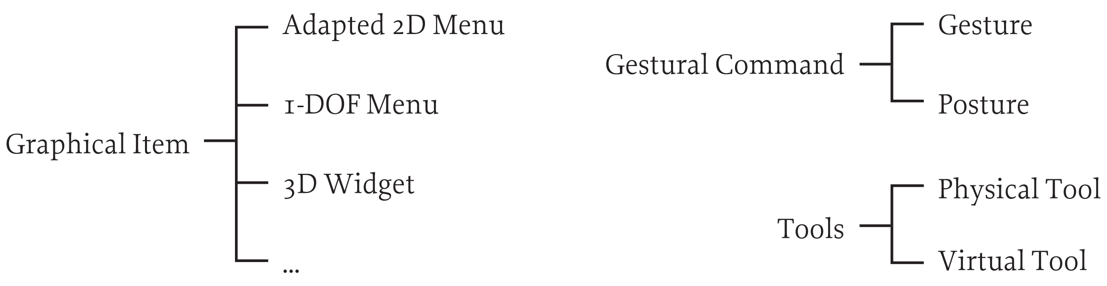


Figure 4.6: Classification of system control techniques (based on [BKLP04]

deals with system control tasks with a focus on more immersive 3D user interfaces; parts of this classification are also suitable for this thesis. The classification in Figure 4.6 is an abbreviated and rather incomplete derivative of their classification. It only contains the system control tasks, which are of interest for the interaction techniques of CAE applications. For more detailed information on the original classification and further system control tasks see [BKLP04].

The classification in Figure 4.6 is organized around three main metaphors – graphical item, gestural commands and tools. Graphical items provide visual feedback to users donating a selection from which the user can choose from. Key aspect of gestural commands is direct input, but also a higher memory load. Although the access to gestural commands is more natural to users, these techniques are depending on heavy memory affordances. A user must remember all commands, which are not presented through choices. Tools are yielding a combination of the other metaphors. They provide visual feedback in their representations (for example, by changing the appearance of the mouse cursor), act as direct input as they are used directly on other objects and require users to remember the way the tool influences the object. In general, it can be noted that these techniques are not isolated from each other and that their boundaries are fluid, resulting in interleaved functionality.

Because of its widely usage in CAE applications, one common interface element should be noted here. Marking Menus are often introduced into the workflow of CAE applications. They provide a fast way to frequently used shortcuts or keystrokes by either popping-up a radial menu or by quickly making a straight mark in the direction of the desired menu without popping-up the menu at all. For more advanced users, these menu systems are serving as the core interaction element for the regulation of the application workflow. Frequently switches between mouse and keyboard are avoided through introduction of these menu systems. Figure 4.7 shows a marking menu of Alias Maya, a formerly version of Autodesk Maya [@May].

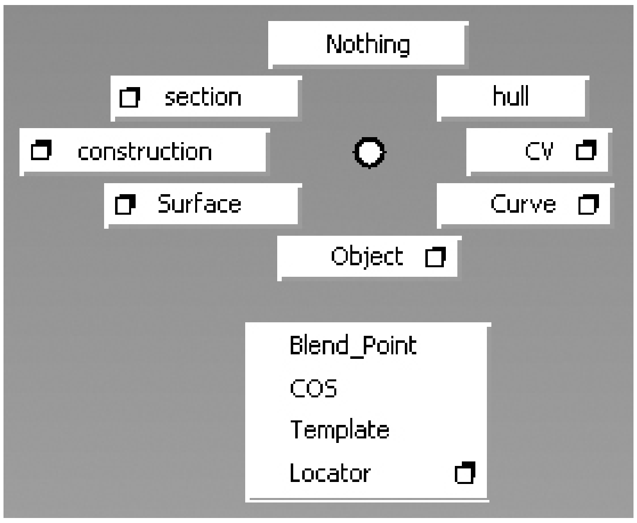


Figure 4.7: Marking menu of Alias Maya [@Ali]

4.4 OpendTect

OpendTect [@Opeb] is an open source software system that provides a complete computer-aided seismic interpretation environment. A look at the interface of OpendTect makes it clear that it is an application from the domain of computer-aided engineering inheriting the commonly shared GUI layout. Figure 4.8 clearly demonstrates the separation into the two major parts of CAE applications. Furthermore, it can be seen that OpendTect also arranges GUI elements in groups according to their functionality. The software is maintained by dGB

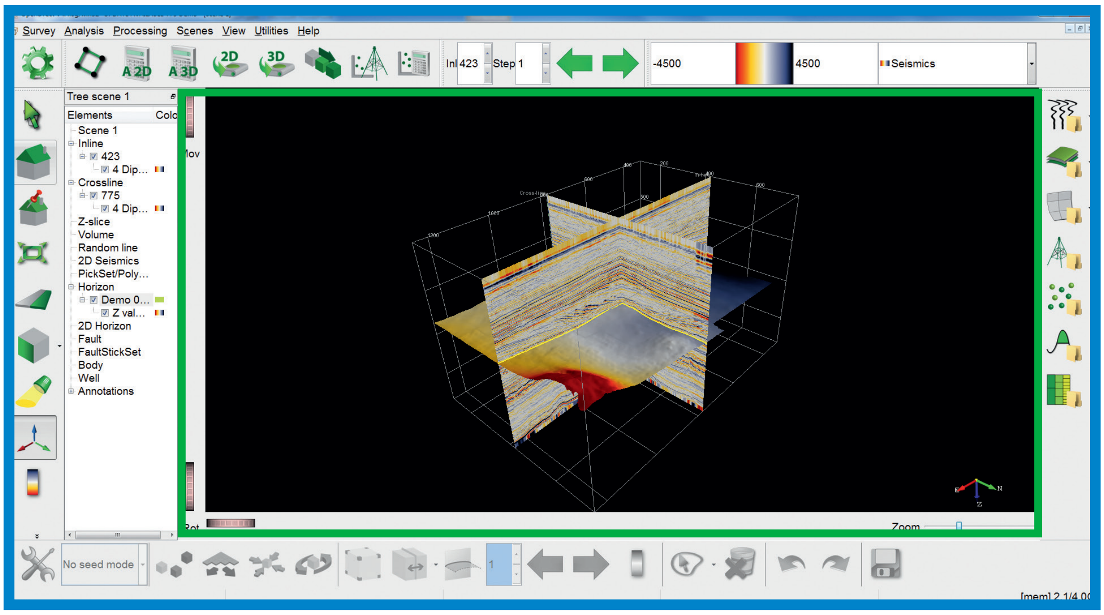


Figure 4.8: Interface of OpendTect, showing the Command Window (blue rectangle) and the Graphics Window (green rectangle)

Earth Sciences, a privately owned company that has been providing seismic interpretation solutions to the oil and gas industry since 1995 [@dGB]. The complete release of the source code of OpendTect for free makes the software systems also a Research & Development platform (R&D), not only for this thesis; as opposed to other commercially available products. The goal of dGB Earth Sciences is to shorten the gap between academic research and operational deployment. OpendTect gives the seismic community an environment for the development of new tools for seismic interpretation.

For a complete list of all features of the application, please look at [@Opea]. The core features of the system, which played a major role in the development of this thesis, are:

- Visualization and analysis of 2D and 3D seismic data in a single survey
- 3D horizon tracking including auto-tracking, plane-by-plane, line and manual tracking
- Plug-in architecture

The following sections of this chapter will discuss each of these features in more detail. In addition to the description of their purposes, there will also be an outline of the incorporated workflows. The next section begins with a general overview of the system architecture and will present all components, which are forming the core system of OpendTect.

4.4.1 System Architecture

OpendTect is a C++ - based environment and uses exclusively open source tools, like the GUI toolkit Qt [@Diga], the 3D graphics rendering library Coin3D [@Kona] and CMake [@CMa], a cross-platform build tool, which takes care of generating system-dependent project files. The part of the Command Window is realized through the GUI toolkit Qt, while the 3D rendering of the geological surfaces and structures is done by the 3D graphics rendering library Coin3D. The connection between these major parts of the application is managed by the GUI – binding library SoQt [@Konc]. The layered architecture of OpendTect is shown in Figure 4.9 (based on [@Konc]). Since all of these libraries are running on all major

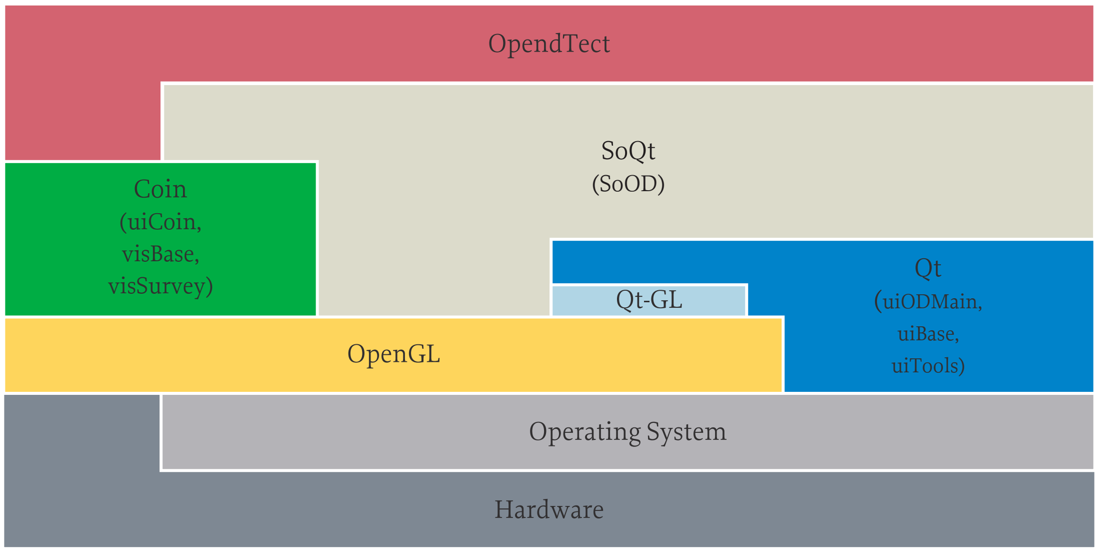


Figure 4.9: Layered architecture of OpendTect. Internal OpendTect modules hiding external services are listed in parentheses (based on [@Konc])

platforms, OpendTect supports all major operating systems, like Microsoft Windows [@Micb], Apple Mac OS X [@Appb] and various Linux distributions. The development of this thesis only uses Microsoft Windows as operating system for the development, since the driver of the incorporated hardware only offers suitable support for this platform.

In the following, each of the core libraries will be presented with focus on the properties of each library, which were of importance for the development of this thesis.

Qt

Qt [@Diga] is a comprehensive C++ application development framework for creating cross-platform applications. Qt is designed to be fully portable over different operating systems.

Development with Qt mainly focuses on GUI functionality, which in terms classifies Qt as widget toolkit. In addition to the easy portability of the API, there are other achievements, which make the API superior compared to other toolkits. One is Qts mechanism for loosly coupled application components, called *Meta-Object Compiler* (MOC). The key service that is realized through the MOC is the *Signal-Slots Mechanism* [BS06]. Signal and slots are handling the communication between application components, e.g. communication between different widgets. One common technique in other toolkits to realize this are callbacks. A callback is typically represented by a pointer to a function. This pointer can be passed as argument to another processing function. When the processing function is executed, it invokes the callback. For example, when a widget changes its state, other widgets can be notified through the invocation of callbacks. The signal and slots mechanism is introduced, because of two major drawbacks of the callback mechanism. When the processing function invokes the callback, it cannot be ensured, that it passes in the desired arguments. Inferentially, this makes the callback mechanism not type-safe increasing the responsibility of the application programmer to detect mismatching types. The other pitfall of this mechanism is the necessary knowledge of the processing function and the callback, resulting in a strong coupling between these components. Signal and slots are enhancing this mechanism by providing an additional layer on top of this communication. Here the notification of a change takes the form of a signal. An object emits a signal when some type of event occurs, e.g. a property change of an object. Other objects then can receive these signals by using slots. A slot is a function that is called when an object receives a signal. Slots are member functions of a class and can either be called directly or be connected to a signal. Emitting the signal is completely encapsulated in the object itself; thus the object does not know which slots will receive the signal. The object emitting the signal and the object calling the slot in response are therefore loosely coupled. Their connection is ensured through the signal and slots mechanism, since the signature of signals and slots must be compatible. This linkage ensures type-safety and it is up to the compiler to detect type mismatches.

More in-depth information about the signal-slot mechanism can be found here [@Digb].

Coin3D

Coin3D [@Kona] is an OpenGL based, retained mode 3D graphics library, maintained by Kongsberg Oil & Gas Technologies and it is fully compatible with Open Inventor 2.1 [@Sil]. Open Inventor provides simplified access to interactive graphics programming problems. It includes a standard set of objects such as cubes, polygons, text, materials, cameras, lights, trackballs, handle boxes, 3D viewers and editors that accelerate programming time and extends 3D programming capabilities [@Konb]. As a consequence, Coin3D benefits from this compatibility in many ways. Firstly, the high quality documentation of Open Inventor also applies to Coin3D offering the application programmer reasonable support. Secondly, code written for one library can easily be ported to other libraries, without a need for major changes. Hence, an extensive set of sample applications with source code exists, exhibiting more complex features of the libraries. The underlying programming model is based on a 3D scene database, called *scenegraph*. A scenegraph represents the structure and the formation of a virtual universe in a hierarchical organization. The scenegraph typically contains nodes, which have different relations to each other. Some of these node objects represent graphical data and others describe the structural relationship between the graphical objects. In general, there are three different kinds of nodes, the root node, group nodes and leaf nodes. The hierarchy of the scene graph can be represented by a tree starting at the root node, which is in turn connected to all other nodes. Connections between nodes are only possible in a directed relation that is from a parent node to a child node. Parent nodes act to group their children together encouraging a spatial grouping of the geometric data. A group node also defines a spatial bound that contains all the geometry defined by its descendants through a diversity of properties. The most commons are used to perform 3D transformations like rotation, translation and scaling. The geometric data managed by the group nodes can be found in the leaf nodes of the scenegraph.

SoQt

SoQt [@Konc] is the glue between Qt and Coin3D, allowing an application programmer to use the 3D rendering capabilities of Coin and simultaneously the simplicity of Qt for the user interface [SIM13]. The fundamental purpose of SoQt is to allow the integration of a Coin3D scenegraph into a Qt application. An example for this integration is the architecture of OpendTect. The Command Window including all system dependencies is completely handled by Qt, the Graphics Window is realized through Coin3D and the fusion of both is managed by SoQt (see Figure 4.9). Rendering a scenegraph into a conventional desktop application involves creating an instance of a SoQt class that is capable of rendering the scenegraph. SoQt offers several classes, each of different levels of abstraction and customization. Some can be used directly while others are abstract allowing full customization.

Besides the advantages listed above, there are also limitations through the combination of these libraries. The incorporation of several libraries makes it hard to change or alter the information flow of the application. The reason for this is that changing or adding new functionality to the application may affect all included libraries at once resulting in a huge impact for the application developer. For simple extensions this might not be the case. However, the goal of this thesis is the design of novel interaction based on the combination of pen and touch to allow the manipulation of the application flow. Thus, all of the described libraries are affected, since these are incorporated in the current handling of the interaction in OpendTect. Section „4.4.1 System Architecture“ showed that OpendTect introduces a new application layer for each of those libraries. This also increases the complexity to integrate new interaction techniques into a desktop application.

More detailed information on this is provided in section „6 Software Architecture“.

4.4.2 Core Features

Visualization of 2D and 3D Seismic Data

The Graphics Window (3D viewer) of OpendTect supports visualization of seismic sections (inlines, crosslines, timeslices), random lines, 2D lines, horizons, faults, wells (tracks, markers, stratigraphy, logs), objects, polygons and picks, volume rendering, and pre-stack gathers. It is also possible to display seismic data in separate 2D viewers. Seismic interpreters must be able to analyze and scan through multiple volumes of data to gather the available geological information and combine this information with their expertise to get the optimal profit of any geological feature of interest. An integration of data processing and visualization supports the cognitive process of combining multiple streams of information. OpendTect allows an seismic interpreter to move visualized elements freely through the data space, it provides interactive ways to analyze data from stored volumes, or it calculates data on-the-fly. The visualized entities include basic elements like slices (e.g. seismic sections, inlines, crosslines, timeslices) and more complex ones like horizons, random lines and cubes.

The workspace of OpendTect can contain one or more Graphics Windows. The basic workflow distinguishes between two manipulation modes, *View Mode* and *Interactive Mode*. Switching between these modes is done by selecting a corresponding menu button of OpendTect. This is illustrated in Figure 4.10 using different button representations. A green arrow represents the Interactive Mode, while a white hand cursor indicates the View Mode of OpendTect.

In View Mode, mouse button operations are used to manipulate the view of a scene, e.g. a user interacts with a virtual camera. A user is able to rotate and zoom each scene independently. For example, left-click and moving the mouse cursor allows virtual trackball navigation (see section „4.3.1 Selection and Manipulation“). Zooming the view can be achieved on several ways. Firstly, a user is able to click and hold the left and middle mouse buttons and simultaneously move the mouse to zoom in and out. Secondly, a user can use the mouse wheel to adjust the zoom factor. Panning the view is accomplished through mouse movement while simultaneous clicking and holding only the middle mouse button.

In Interactive Mode a user is able to manipulate the visualized 3D data. Operations like moving elements in 3D space or resizing elements are done in Interactive Mode. Furthermore, picks for horizons and picksets can be created. For example, moving a slice in OpendTect can be described as follow: A user selects the slice by clicking with the left mouse button in the tree or directly in the graphics window on the object. An appearing 3D-Widget gives visual feedback for a successful selection of the object. Clicking and holding with the left mouse button on the 3D-Widget allows the translation of the slice to a desired position.

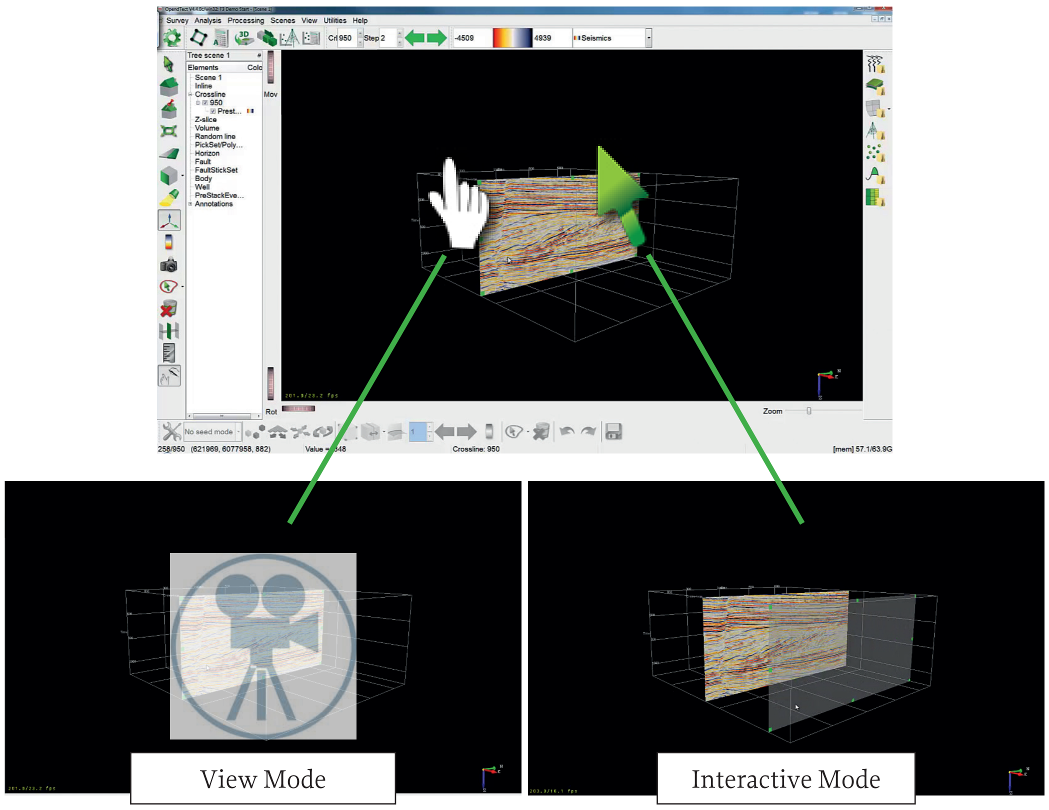


Figure 4.10: View Mode and Interactive Mode in OpendTect. The user interacts with the camera in View Mode, while manipulating the seismic data is done in Interactive Mode

3D Horizon Tracking

OpendTect supports various algorithms for horizon tracking. These include standard amplitude and similarity horizon tracking and step-wise tracking. Horizons can be tracked in the following modes:

- Auto-track mode
- Track on lines (in the 3D scene or 2D viewer)
- Manually pick on lines (in the 3D scene or 2D viewer)

In auto-track mode a user specifies a tracking area in which the surface of the horizon grows automatically. The tracker can be controlled with various parameter configurations. There are specifications based on amplitude only, step-wise amplitude difference or similarity (cross-correlation). While amplitude only and similarity are highly depend on the picked seeds, step-wise tracking first tracks the areas that have a minimal amplitude difference with the picked seeds and allows for a greater difference in subsequent steps.

Tracking on lines and manually picking are providing similar tracking techniques. A user makes manually interpretation on inlines and crosslines, either by drawing lines on the slices or by picking individual seed points. If a user draws lines on the slices, seeds are automatically added by OpendTect and an event is tracked or interpolated between these seeds.

The tracked surface of the horizon can be edited on various ways. Areas that were not adequately tracked can completely be removed and re-tracked after picking new seed points or after adjusting the tracking parameters. There are various interpolation algorithms to auto-fill existing holes in the tracked surface. The basic workflow includes 6 steps, which can be described as follows:

1. Add an inline/crossline
2. Add a new horizon
3. Define mode and tracker settings
4. Pick seeds
5. Autotrack or interpret the horizon on lines
6. Edit the tracked surface of the horizon

Step 4 is of major interest for the development of this thesis, since it requires frequent user intervention. For a complete description of each step please have a look at [dGB12]. The current workflow of OpendTect during seed picking requires frequent mode switches between View Mode and Interactive Mode. The reason for this is that a user is only able to add a new seed in Interactive Mode, while adjusting the view of the scene is only possible in View Mode. A successful and promising tracing of a horizon is highly dependent on the accuracy of the picked seeds. To pick seeds accurately, a user must make several adjustments of the view to get a suitable overview the currently edited event. The major drawback in the current workflow is, that the picking of seeds is done on the Graphics Window, while the necessary mode switches between View Mode and Interactive Mode are commonly done on the Command Window. A user has to leave her or his point of interest (POI) on the Graphics Window to execute a mode switch. This results in multiple mouse manipulation operations, like moving to the corresponding button at the border of the user interface of OpendTect and returning to the POI on the Graphics Window. Thus a seamless workflow is not possible, because of several interruptions through mode switches. Figure 4.11 shows the traditional way of picking seeds in OpendTect. The implemented interaction techniques of this thesis, will present an enhanced and more seamless workflow in later sections.

Plug-In Architecture

There are basically two possibilities to extend OpendTect. Since the software is released as open-source, developers can modify existing classes and modules to add new features. This way of development grants full access and control to all available features in OpendTect. However, there are also drawbacks with this approach. First, it is in the responsibility of the application programmer to keep new OpendTect releases in sync with the modified sources. The most important drawback, however, is the distribution of new implemented features. Since the modified sources are not included in the standard available OpendTect application, it will not be possible to distribute the added functionality within OpendTect. A straight forward way to overcome these limitations is to develop plugins, rather than modifying the underlying sources of OpendTect. The architecture of OpendTect makes it possible to dynamically load plugins into OpendTect at runtime. The developer can implement extensions in OpendTect encapsulated into an independent library and load this library dynamically at runtime. The benefit of this approach is an easy distribution to the community, since only the developed plugin has to be shared. Further advantages are easy integration into OpendTect, because no heavy installation steps are needed. A plugin needs to contain some predefined functions, which will be called automatically by OpendTect when the dynamic library is loaded. With the release of OpendTect 4.2.0 the implementation of this standard function became straight forward, through the introduction of macros defining the plugin functions. One drawback of plugin development is that there is only limited access to control the flow of the application.

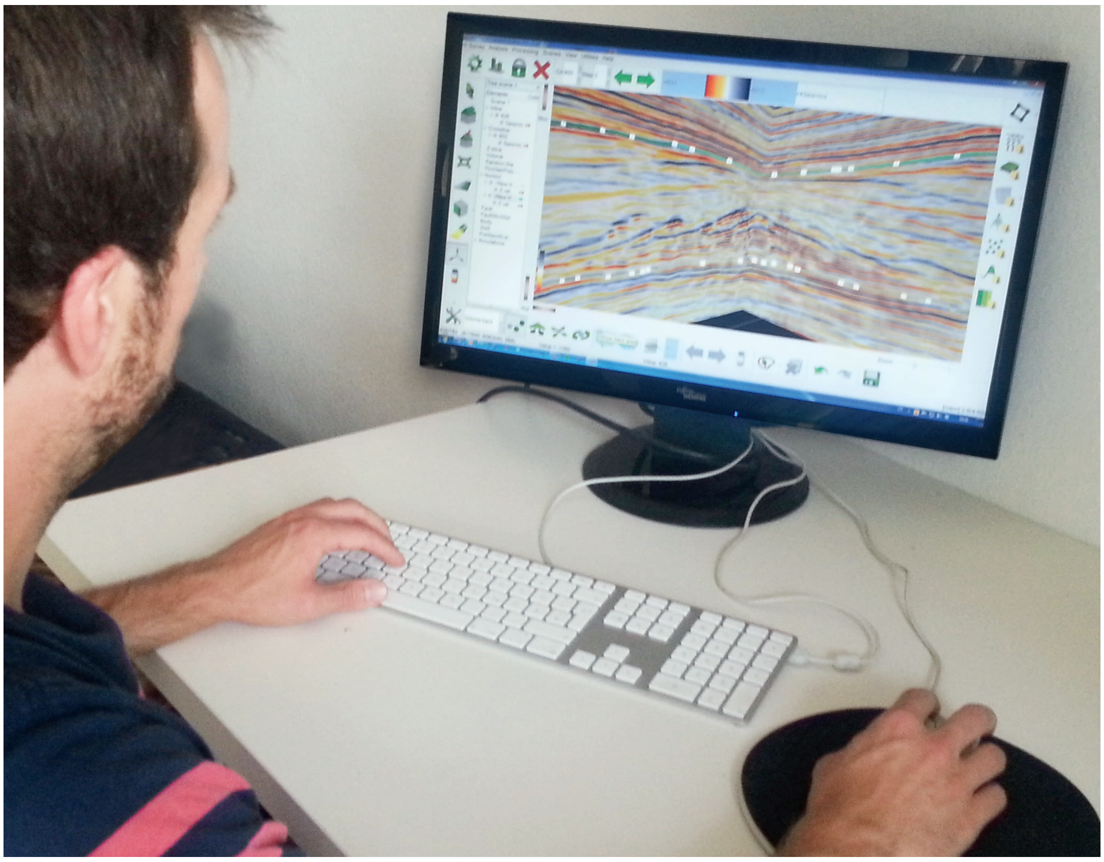


Figure 4.11: Seed picking in OpendTect using traditional input devices. While seeds are picked on the Graphics Window, mode switches are executed on the Command Window

05 // Pen & Touch based Seismic Interpretation

5 PEN & TOUCH BASED SEISMIC INTERPRETATION

Section „3.3 Interactive Displays" showed that there is a shift towards displays that unify input and output. This shift is well-founded on the commercial success of mobile devices. The existing human computer interaction (HCI) however still relies on conventional devices like keyboard and mouse. With progressing of computing, communication and display technologies, it is widely assumed that the existing HCI techniques are manifesting a bottleneck in the effective utilization of the available information flow [SPH98]. The drawback of conventional devices is their restriction on information and command flow between user and the computer system. In more immersive environments, this limitation becomes even more apparent. Thus, in recent years, several research studies examined the introduction of new modalities into HCI to overcome this bottleneck. With respect to the natural communication between humans, these modalities correspond with the five basic human senses (sight, hearing, taste, smell, touch). Voice recognition, gesture recognition, eye tracking or force sensing are only a few active topics of recent research studies to investigate potential HCI modalities. Section „3.2 Multimodal User Interfaces" introduced basic findings about interaction incorporating multiple input modalities. Multimodal interaction encompasses a bright diversity of research domains, including cognitive psychology, software engineering and HCI. All of these interleaved domains result in a complex cross-disciplinary research subject. As mentioned above, recent research studies try to incorporate new input modalities based on the human senses. Therefore the underlying cognitive psychological concepts of human perception during multimodal interaction are playing a major role in the development of such techniques for the use in the computer domain. These underlying principles are not only of interest for interaction designers, but also for software engineers. Software engineers study software architectures and multimodal processing techniques for the development of such multimodal interfaces with respect to these underlying principles. This cross-disciplinarity clarifies, that the development of a multimodal application requires a number of components and careful implementation work [DLO09].

This chapter illustrates and validates the modeling of the implemented features for integration into the seismic interpretation workflow based upon the findings of section „3.2 Multimodal User Interfaces". Firstly, the seismic usage context will be defined, followed by the chosen design consideration of the combination of pen and touch. Finally, concrete characteristics and the modeling of exemplary multimodal seismic interpretation tasks are described.

5.1 Seismic Usage Context

In light of the emergence of mobile devices, it becomes even more important to clearly classify which platforms are supported in the development of novel interaction techniques and which not. On the software side, developers are asked to support a variety of devices and platforms. Regarding supported hardware, it can generally be distinguished between mobile development and stationary development. For mobile development, mobile devices act as development platform, while for stationary development conventional desktop computers are used. Modern software architectures should be designed to be portable from one development platform to the other. However, for the development of bimanual interaction techniques, one has to consider the constraints of the development platform, since there are differences in the achievable effect. This is especially true for workflows of seismic interpretation, since some steps are requiring the full attention of the user, which might also be occupied by the usage context. For example, Figure 5.1 demonstrates a bimanual mobile usage context. Section „3.1 Foundation of Bimanual Interaction" revealed the asymmetric division of labor between the DH and NDH in mobile device interaction.

While the dominant hand interacts with the content, the non-dominant hand may rearrange the mobile device. However, the important bottleneck for bimanual interaction is, whether or not the NDH rearranges the mobile device, that the NDH is occupied by holding the mobile device [dAn12]. A user is either able to use the pen (Figure 5.1 left) or to tuck the pen between the fingers and interact using touch interaction (Figure 5.1 right). The results are unimodal core interactions, because of sequential pen or touch input. According to the CASE-model, this is an exclusive fusion of input modalities. Despite the possibility of using pen and touch input, a user is not able to use the input modalities in an alternate or synergistic way because of the NDH's liability of holding the device. For the described workflow of horizon tracing (see section „4.2.3 Core Features") a mobile usage context is only partially suitable. First, the small form factor of mobile devices only allows to view a small subset of all the data in the seismic volume. Interpreters, however, need to integrate

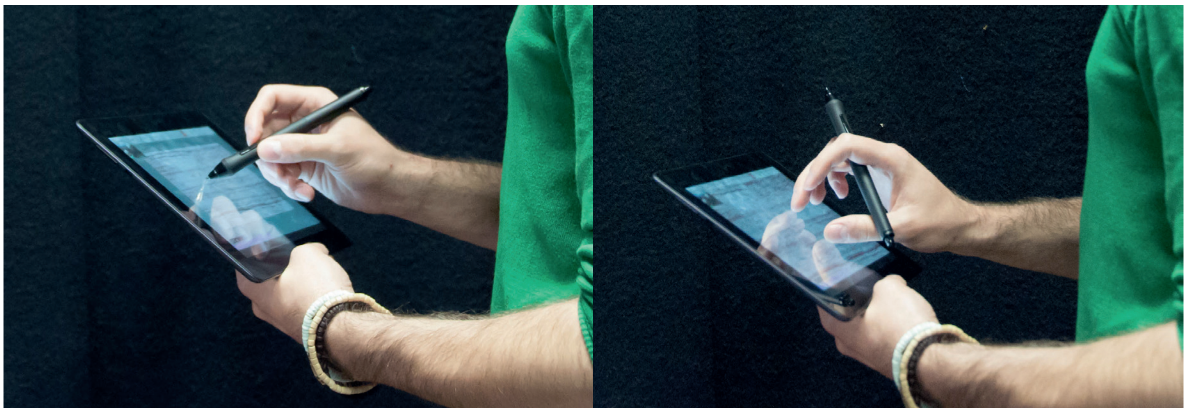


Figure 5.1: Bimanual mobile usage context [dAn12]

data and concepts from several sources. Second, the asymmetric division of labor described above makes it hard to pick seeds accurately. In a mobile usage context, users have to hold the device, which will result in small movements. This might affect the accuracy of the picked seeds, because of incidental movements of the NDH.

In contrast to this, a stationary usage context allows full bimanual interaction. Since the NDH is not occupied for holding the device, a user is able to incorporate pen and touch interaction simultaneously. With this context as foundation, the implementation of bimanual core operations is possible. Moreover, the introduced workflow of seed picking is usually done using a stationary usage context, since it allows a user to fully concentrate on the task. The form factor of stationary devices allows to view a considerable amount of the seismic data. Figure 5.2 illustrates a stationary bimanual usage context, where a user is able to use the NDH and DH, according to the all possible fusion of input modalities supposed by the CASE-model.

5.2 Pen and Touch User Interface

The implemented extension of the seismic interpretation framework developed during this thesis is called *PenTouch Plugin*. In section „4.4.2 Core Features“ the plugin architecture of

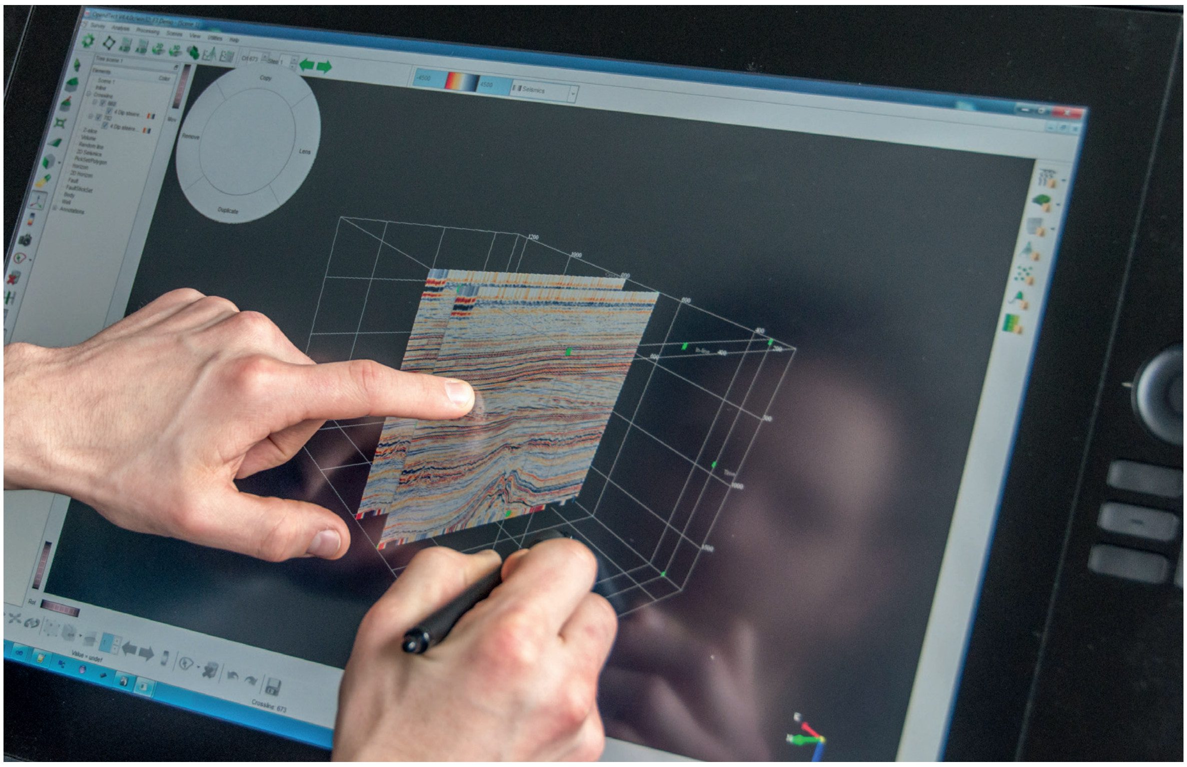
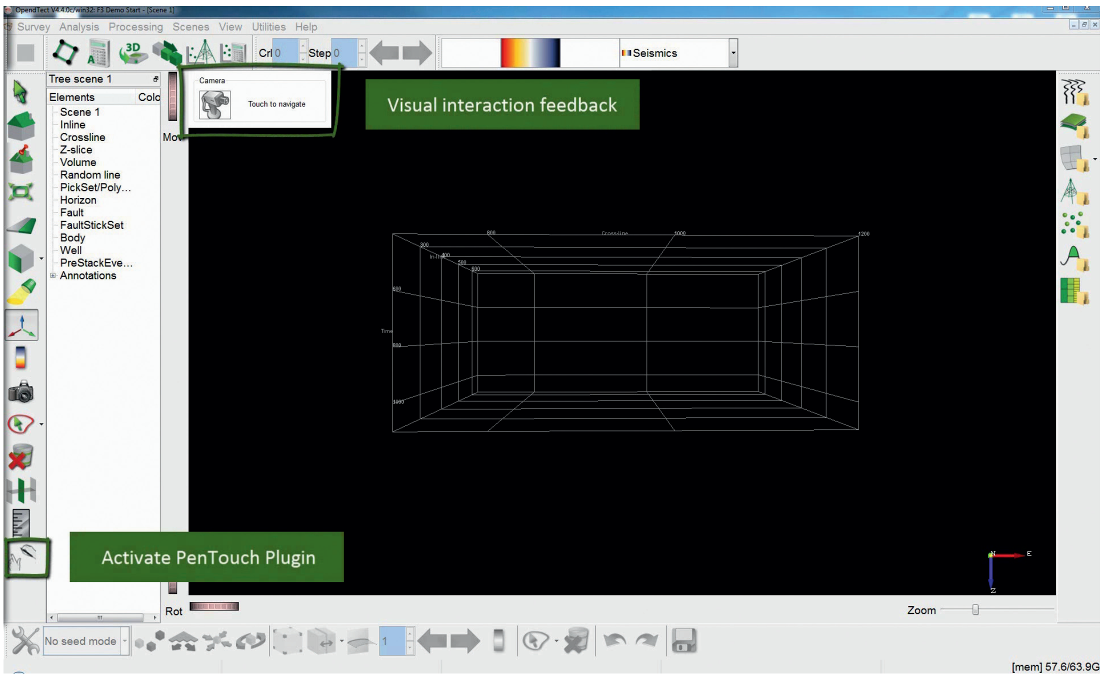


Figure 5.2: Bimanual stationary usage context

OpendTect was described. It was decided to implement all features using this development approach. This guarantees that the implemented functionality can be shared with the seismic community without complicated installation steps. The PenTouch Plugin can be loaded during runtime into OpendTect. When the plugin is loaded users have the possibility to activate the plugin by pressing a newly added menu button. If the plugin is activated, the implemented features of combined pen and touch interaction are available. Users are always able to toggle between the interaction techniques realized by the PenTouch Plugin and the traditional mouse and keyboard based interaction by pressing the according menu button. Section „3.2.3 Guidelines for Multimodal User Interfaces“ listed considerations, which have to be taken into account when designing Multimodal User Interfaces. Saffer [Saf08] identified several considerations for gestural interfaces. One is that users must be aware that they are interacting with a gestural interface at all. Therefore the system should provide appropriate feedback, signalizing available commands. A huge set of available gestures increases the memory load of users, since they have to remember the according commands of each gesture. This was also considered in the implementation of the PenTouch Plugin. When activated, the plugin provides visual feedback for the currently available interaction techniques. These techniques are implemented using the Tool metaphor described in section „4.3.2 System Control". More detailed information on this will be provided in later sections of this chapter. The feedback of the PenTouch Plugin shows a brief instruction how to execute particular interaction techniques depending on the currently selected tool. Figure 5.3 shows the menu button for triggering the activation and the visual feedback of the PenTouch Plugin.

Figure 5.3: PenTouch Plugin loaded into OpendtTect



5.3 Fusion of Pen & Touch Input Modalities

Section „3.2.4 Fusion of Input Modalities“ investigated the fusion of input modalities from a general point of view. In this section, a more concrete view of the fusion of pen input and touch input is provided. These are the incorporated input modalities the novel interaction techniques are built upon. Firstly, the prerequisites and challenges forming the starting point for this fusion are displayed. Finally, the resulting design consideration will be illustrated.

5.3.1 Properties shared by Pen and Touch

Directed by current research efforts and industry trends, Hinckley et al. [HYP+10] defined a design space for the fusion of pen input and touch input. Table 5.1 summarizes the properties shared by pen and touch of their design considerations regarding the combination of these input modalities. The properties listed in Table 5.1 in conjunction with the guidelines proposed in section „3.2.3 Guidelines for Multimodal User Interfaces“ map out, how an effective design space of pen and touch input can be chosen. The main properties listed in the table are the properties *Contacts*, *Occlusion*, *Precision* and *Intermediary*. The pen defines one precise and unique contact point, with only small occlusion of the display. In contrast to this, touch input allows multiple contact points simultaneously. In addition, there is no intermediary in touch input, because of bare-handed input. The properties listed in the table should not be considered as advantages or disadvantages of the input modalities.

PROPERTY	PEN	TOUCH
Contacts	1 point A single well-defined point.	1-10+ contact regions with shape information
Occlusion	Small (pen tip) But hand still occludes screen.	Moderate to Large (pinch, palm, whole hand)
Precision	High Tripod grip, writing, sketching.	Moderate
Hand	Dominant hand	Either hand / Both hands
Intermediary	Mechanical Intermediary Takes time to unsheathe the pen. Pen can be forgotten.	None: Bare-Handed Input Nothing to unsheathe or lose. No lever arm. No buttons.
Acquisition Time	High (first use: grab pen) Moderate on subsequent uses: tuck pen between fingers	Low No mechanical intermediary to acquire.
Activation Force	Non-Zero	Zero (capacitive touch)
False Positive Inputs	Palm Rejection (while writing) Palm triggers accidental inputs, fingers drag on screen, etc.	»Midas Touch Problem« Fingers brush screen, finger rests on device while holding it.

Table 5.1: Properties shared by pen and touch (based on [HYP+10])

Depending on the context of usage, each of these properties has its benefits and drawbacks. However, based upon this enumeration of considerations, Hinckley et al. [HYP+10] articulate their approach as follows:

Differentiation between Unimodal Pen and Unimodal Touch

Unimodal pen interaction is used for ink mode, that is, manipulation of object properties is accomplished through pen interaction. Unimodal touch interaction communicates commands to the application. Mode switches are executed using unimodal touch (touch, multitouch).

Interchangeability of Unimodal Pen and Unimodal Touch

For suitable unimanual usage scenarios, the user can tuck the pen between the fingers (see Figure 5.1 right) and interleave or interchange pen input with touch input. Fission of the application is for both input modalities the same.

Core Tasks are designed for Unimanual Touch Interaction

Unimanual touch interaction is used for frequently used core tasks. In parallel, a bimanual interaction setup is also possible by assigning these tasks to the non-dominant hand which realizes efficient interaction as well as advanced gestures.

Simultaneous Pen & Touch realizes new Interaction Techniques

In a multimodal setup of pen and touch input, touch input is used for object selection. During object selection, pen and touch are incorporated together as a compound gesture. Pen input in reference to an object selected with touch input is interpreted as gestural command.

The implementation of the new interaction for seismic interpretation is based on this design consideration, since it promises a suitable interaction workflow. The concrete design of novel interaction techniques is described in section „5.3.3 Mapping of Input Modalities“.

5.3.2 Fat Finger Problem

Direct–touch finger input has two fundamental drawbacks. First, the interacting finger of the user occludes a valuable amount of the target area during the immediate moment of touching the screen. This problem is referred to as the occlusion problem [WFB+07]. Second, the touch area of the finger is many times larger than a measured pixel of the display, referred to as the fat finger problem [WFB+07]. The reasons for the occlusion problem are easy to trace. Especially on devices with a small form factor it becomes obvious that the finder hides a large part of the display. The reasons for the fat finger problem, however, are more difficult to identify. Holz [@Hol] conducted several user studies and verified his hypothesis assuming that the fat finger problem, in fact, is a perceived input problem. While a user believes to target using the visual cues on top of his finger, current touch-sensitive devices are sensing the contact point using the haptic features at the bottom of the touch finger. The resulting parallax, manifested through the offset between the visual measurement of the user and the haptic measurement of the device, is the main reason for the fat finger problem. The previous section listed the properties of pen input and touch input. With pen input, there is a well-defined unique contact point, which overcomes the fat finger problem. Additionally, there is also a vertical offset between a user's hand, holding the pen and the display minimizing the occlusion problem. Figure 5.4 summarizes the benefits of pen input compared to touch input.

However, several research studies proved touch input to be highly intuitive and gaining from good pointing performance in coarse pointing tasks. Furthermore, the small Control-Display ratio realizes direct bare-handed manipulation and makes touchscreens well accepted because of quick learning and high user satisfaction [PWS88]. The tension between the low precision of finger input and the high precision needed by graphical user interfaces in conjunction with the high acceptance of direct-touch input makes the fusion of pen and touch input a reasonable approach. The following section will demonstrate in which way the implementation of novel interaction techniques meets these claims.

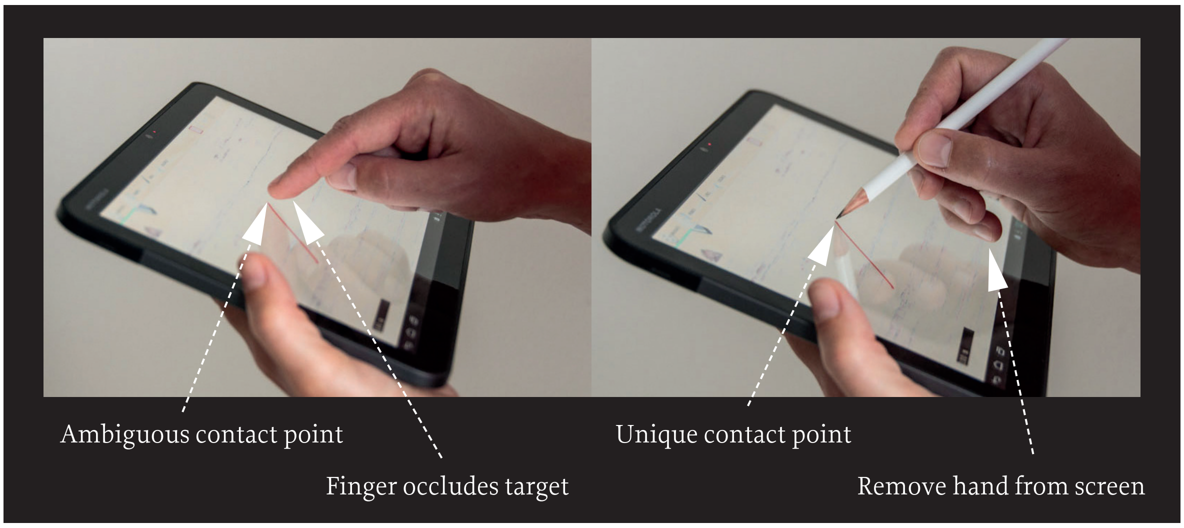


Figure 5.4: Pen input compared to touch input [dG12]

5.3.3 Mapping of Input Modalities

This thesis investigates the design and development of novel pen and touch based interaction techniques for a professional-level seismic interpretation application. So far, underlying principles of bimanual human interaction, computer-aided engineering applications and multimodal user interfaces have been discussed. In addition, most pen and especially multitouch enabled applications are designed for mobile usage context. In contrast, this thesis focuses on using these input modalities for creating and manipulating content on traditional desktop workstations with conventional devices like mice and keyboard. All of these factors have to be considered when investigating novel interaction techniques.

One major design consideration for the interaction techniques in this thesis is that the area of application of each interaction technique should be correlated to each other. Therefore, a traditional desktop workflow has to be chosen, which can benefit from the incorporation of combined pen and touch based interaction techniques. In section „2.3 Seismic Interpretation“ and „4.4.2 Core Features“ the workflow for horizon tracing was introduced. This workflow contains interaction, with a high correspondence to the properties of pen and touch input proposed in section „5.3.1 Properties shared by Pen and Touch“. The workflow of picking seeds requests the fine granularity of the pen, while the coarse manipulation of the Graphics Window is suitable for multitouch interaction.

Moreover, one major tradeoff in the current horizon tracing workflow of OpendTect are frequently mode switches. However, due to rich user interfaces and the high degree of complexity of modern desktop applications, mode switches are necessary and cannot be avoided. The aim must be to allow mode switches in a way that does not increase the cognitive load of the user. This thesis facilitates mode switches by a well-chosen fusion of pen input and touch input to allow smooth, consistent and minimal disruptive mode transitions. The most basic approach to realize this is a *Mapping of the Input Modalities*. This mapping of input modalities is grounded on the Kinematic Chain Model by Guiard [Gui87]. An asymmetric division of labor between touch input and pen input is realized by mapping one of two core interaction techniques of the horizon tracing workflow to each of the input modalities. Using the dominant hand, unimodal pen input is used to insert and modify content, while unimodal touch input executed with the non-dominant hand is used to navigate in the Graphics Window. With this design consideration, the most frequently executed mode switches of changing the system state from view mode to interactive mode in OpendTect are circumvented. Section „3.2.1 Theoretical Principles“ depicted the action state of the user in a multimodal system. Here, the user chooses the input device for transmitting content to the application. The mentioned mapping of input modalities employs implicit mode switches, depending on whether a user interacts with pen or touch. This results in an efficiently interleaving between those modalities. The basic mapping of input modalities is shown in Figure 5.5.

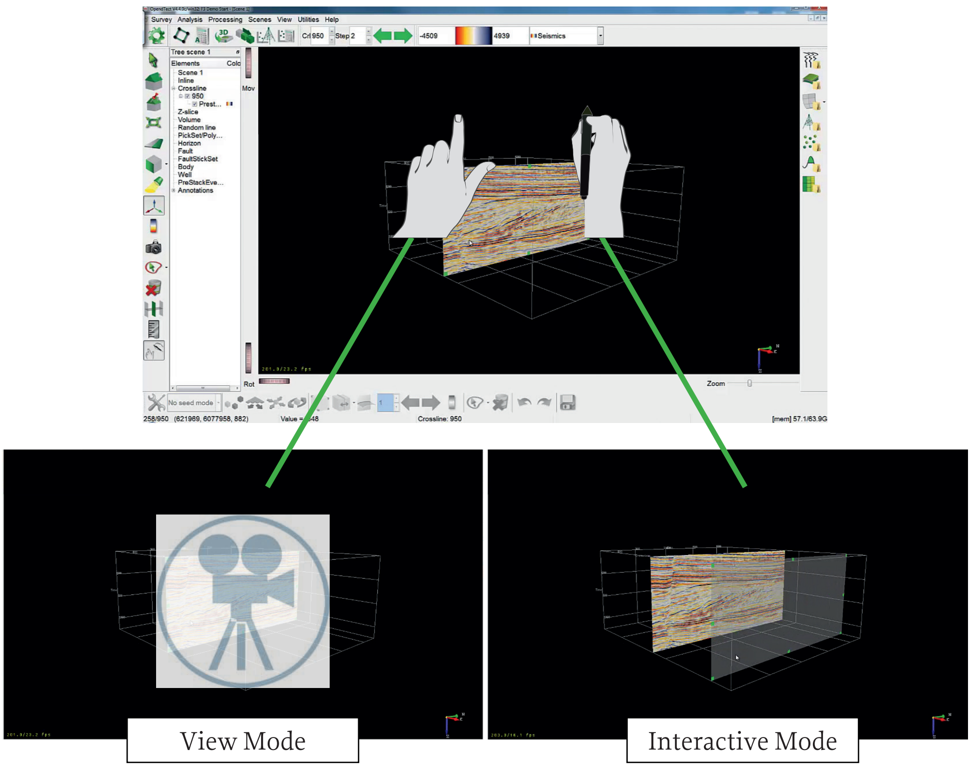


Figure 5.5: Pen and touch based Mapping of Input Modalities. View and Interactive Mode of OpendTect are mapped to touch and pen interaction respectively

Despite the advocated differentiation between the roles of pen and touch, there is one exception to this rule. As experienced by Hinckley et al. [HYP+10], users tend to interchange pen and touch input when interacting with common controls, like menu bars, buttons or checkboxes. In this context, users expect pen and touch to behave in the same way, that is, simply executing the offered interaction of these controls. For example, pressing a button with using the pen is expected to behave the same way, when executed using touch interaction. Section „4.2 User interface" highlighted the commonly shared GUI layout of CAE applications. The introduced Command Window typically includes groups of 2D interface controls according to their functionality. The chosen exception to the advocated differentiation of pen and touch input becomes apparent when a user is interacting with the Command Window. In the implemented design, pen and touch input are fully interchangeable if a user is interacting with controls of the Command Window. Here, pen input and touch input are simply mimicking mouse behavior. The fusion of these input modalities is only available on the Graphics Window. According to the CASE-model (see section „3.2.4 Fusion of Input Modalities"), the interaction on the Command Window of unimodal pen or unimodal touch matches exclusive fusion of input modalities, while the interaction on the Graphics Window corresponds to alternate fusion of input modalities. Moreover, synergistic fusion of pen input and touch input is only supported on the Graphics Window through multimodal interaction techniques.

5.4 Seamless Mode Switching

The previous section introduced the fundamental interaction design implemented in this thesis. The mapping of input modalities already provides an effective functionality for interleaving between the two basic interaction modes of OpendTect (see section „4.4.2 Core Features“). Actually, this elementary design does not use synergistic fusion of input modalities. Bimanual asymmetric interaction, suitable to simplify complex tasks, however, requires this kind of fusion of input modalities. To fully benefit from potential of bimanual asymmetric interaction more interaction techniques were implemented. These techniques are implemented using a tool interface (see section „4.3.2 System Control“). The user selects a tool, which allows using a multimodal interaction technique. These multimodal interaction techniques contain an asymmetric division of labor between the NDH and DH. Each tool defines a constraint to a selectable data type of OpendTect. If this data type is selected with the NDH, the current selected tool becomes active and the multimodal interaction can be executed by additionally placing the pen on the same constrained data type. For a minimal disruptive workflow experience, a so-called *fallback* mechanism is introduced. If a tool is selected, but the currently selected data type defined by the tool itself does not match the data type selected with the non-dominant hand, the fallback mechanism becomes active. The fallback mechanism is implemented through a relationship with another tool, where one tool is the fallback tool of the other. If the defined constraints of the currently selected tool are not satisfied, the fallback tool will verify if it's on constraints are passable. If so, the fallback tool will temporarily become the active tool. Visual cues (see Figure 5.7) provide feedback for this fallback mechanism. A user is informed, whether or not the current multimodal interaction is available. The fallback tool allows implicit mode switching between tools, minimizing the requirement to explicitly interact with custom controls. An in-place menu gives access to one of the available tools. Section „3.5 Mode Switching Techniques“ introduced in-place commands. In-place commands are popup menus placed next to the invoking finger [@Mic]. They seem to be a promising solution for a rewarding user experience, while simultaneously providing seamless and efficient mode switching techniques. Therefore, basic mode switches are executed using the so-called in-place pie menu. The PenTouch Plugin includes visual feedback to highlight the in-place pie menu. This is realized by introducing the PieMenuActivator. Besides the feedback, this implementation component allows to activate the pie menu. For this, a circular feedback for the first touching finger is provided. If a user touches the screen, the PieMenuActivator becomes visible. Now, the pie menu can be activated doing a tap gesture with a second finger. The PieMenuActivator is only available when interacting with a single finger. When more fingers are on the screen, the activation of the pie menu is permitted. This was implemented for the sake of acceleration based on the described characteristics of marking menus of Computer Aided Engineering applications in section „4.4.2 System Control“. Figure 5.6 displays the interaction with the pie menu to select a tool.

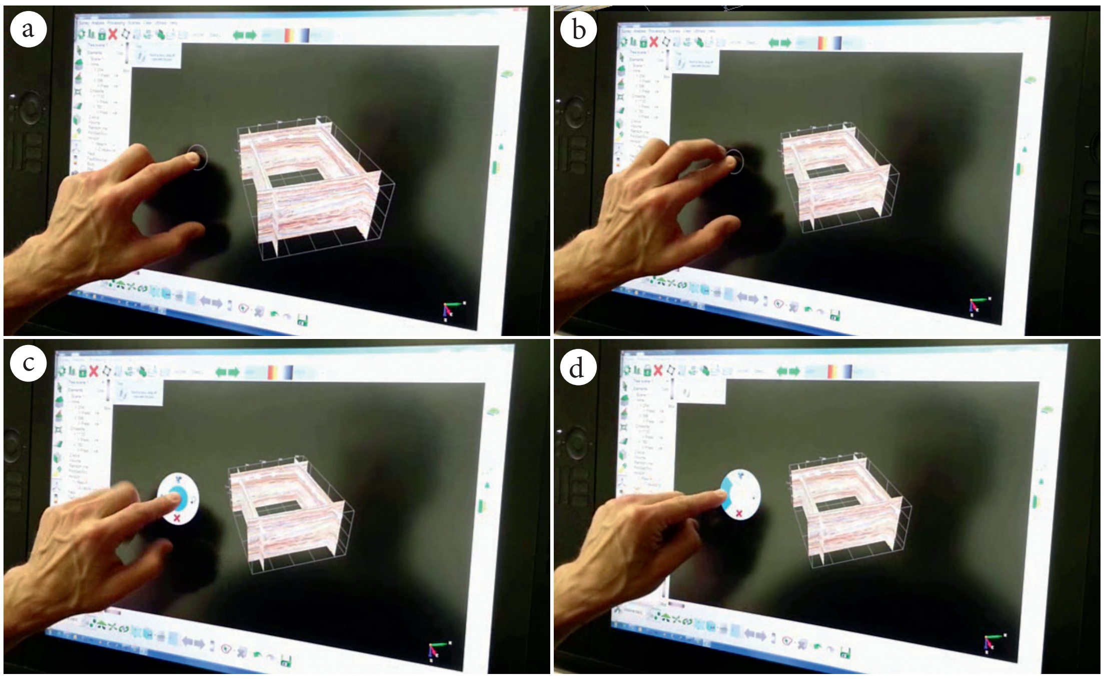


Figure 5.6: In-place pie menu, a) Circle around finger indicates the pie menu activator, b) Invocation of pie menu using tap gesture of second finger, c) Invoked pie menu, d) User selecting tool

The available tools were implemented in order to simplify the incorporated interaction techniques of the conventional horizon tracing workflow. Figure 5.7 illustrates the selection of a tool and the visual feedback provided.

5.5 Exemplary Multimodal Seismic Interaction

The elementary steps of the horizon workflow were introduced in section „2.3 Seismic Interpretation" and include the picking of seeds as well as the manipulation of slices. With respect to slice manipulation the implemented features are simplifying the creation and deletion of slices. Using a bimanual asymmetric interaction technique, where the roles of the DH and the NDH are divided, supports seed picking. Each of the implemented techniques will be presented in the following sections. Basically, the workflow of all implemented multimodal interaction techniques is the same. With respect to the proposed design space of Hinckley et al. [HYP+10] (see section „5.3.1 Properties shared by Pen and Touch"), the NDH using touch interaction is primarily used for object selection. The DH is occupied for holding and interacting using the pen. Invoking a multimodal interaction integrates two steps:

1. Selecting the object with the NDH (touch)
2. Executing the multimodal interaction using the DH (pen)

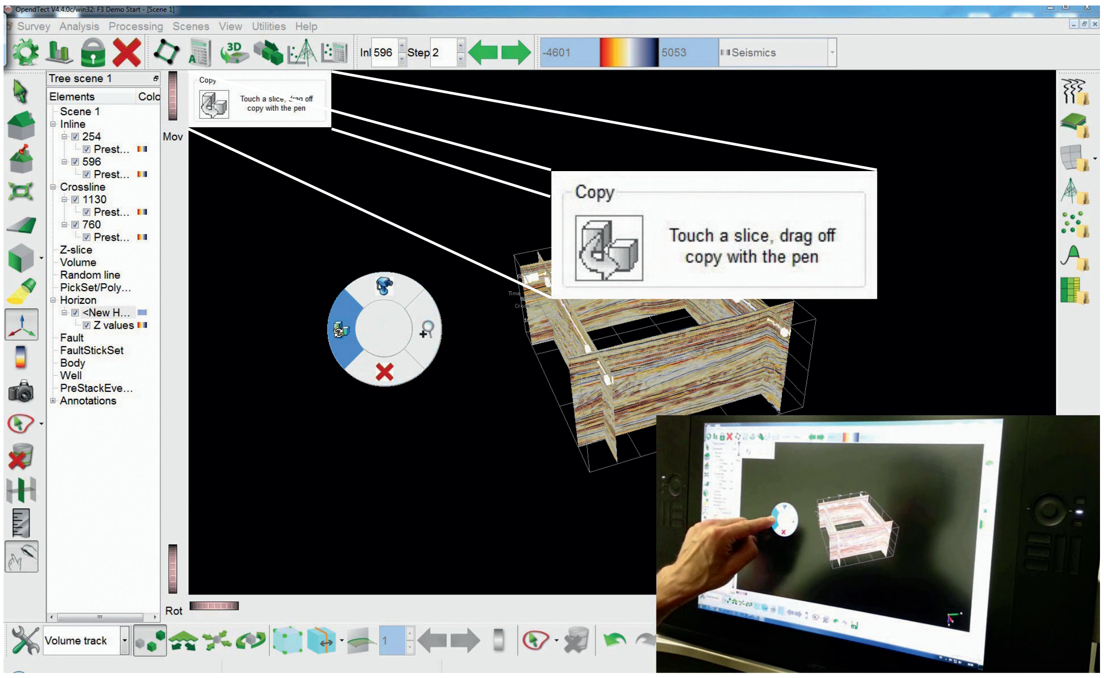


Figure 5.7: Selection of the copy tool, visual feedback is provided in the top left corner

Although the similarity in the course of interaction realizes a consistent workflow, there are also some differences between these interaction techniques. These differences are mainly manifested in the fusion of pen and touch input. The following subsections introduce each of the bimanual multimodal interaction techniques and discuss the implemented fusion of pen and touch input.

5.5.1 Manipulation of Slices

In case of manipulating slices, the fusion of pen and touch input is accomplished using an alternate approach (see section „3.2.4 Fusion of Input Modalities“). Here, touch and pen input are combined but the course of the multimodal interaction technique is highly sequential. This results in an alternate fusion of input modalities. Touch input and pen input are used one after the other. For example, the creation of a new slice incorporates the selection of an already existing slice by touching it with the non-dominant hand, now dragging off a copy with the pen and removing the hands. The course of the workflow is illustrated in Figure 5.8.

Although a user only executes three simple steps, the system executes several mode switches in the background. Simplicity is achieved by hiding these mode switches from the user. A user is not aware of executing a mode switch at all during the interaction of slice

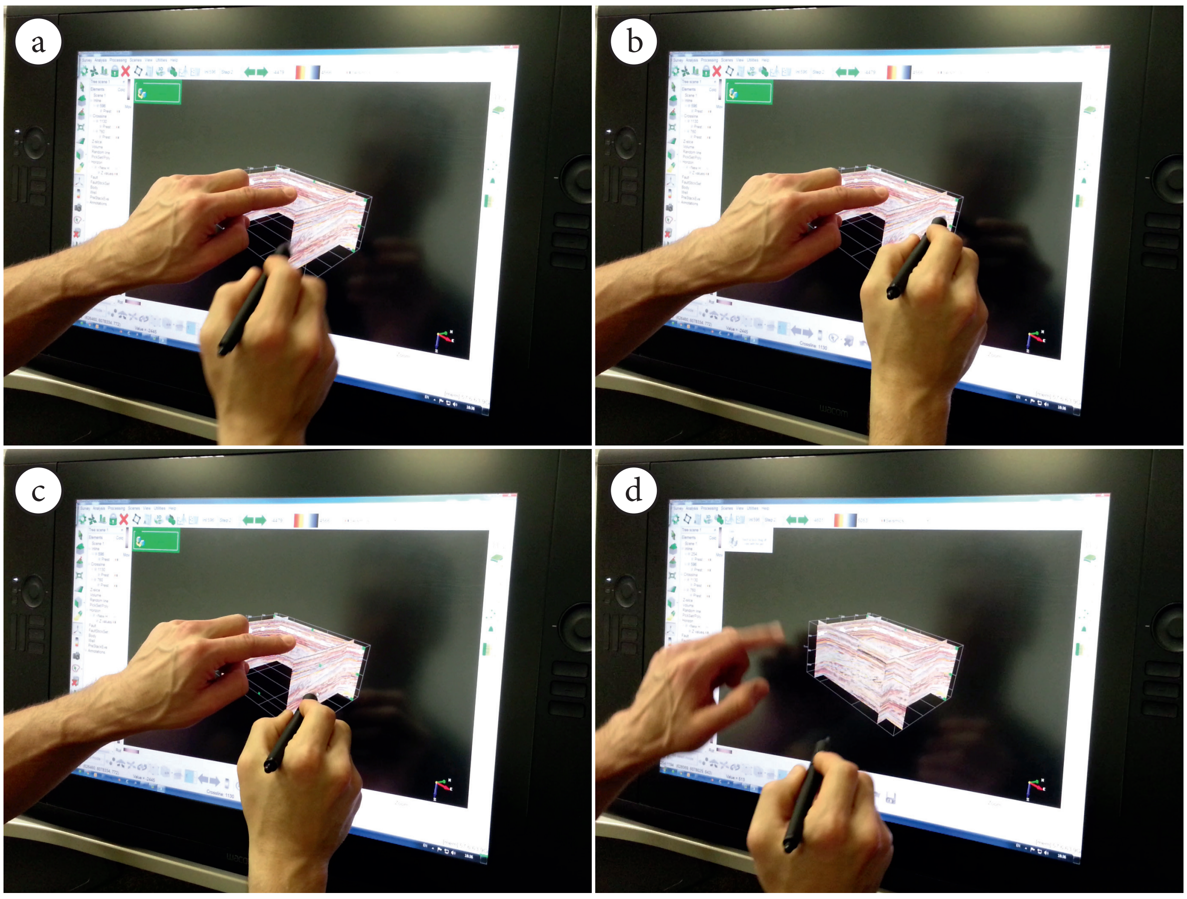


Figure 5.8: Workflow of bimanual asymmetric creation of slices, a) Selection with NDH (touch), b) Dragging off copy with DH (pen), c) Positioning of slice at desired position, d) Invoking creation of slice by removing NDH

manipulation. Implicitly occurring mode switches during the multimodal slice creation can be summarized as follows:

1. Holding a finger on a slice integrates two mode switches
   a. Object selection
   b. Transition to multimodal mode
2. Dragging of a copy with the pen embeds three mode switches
   a. Select same object
   b. Invoke copy command
   c. Release pen at desired location
3. Removing the finger incorporates one mode switch
   a. Releasing the NDH triggers slice creation and returning to default state of the system

The deletion of slices works in a similar way to the listed steps of slice creation. The fission of input modalities is the deletion of this object, therefore pen and touch input are mapped to object selection and confirmation of deletion. In the first step, a user has to select an object with the non-dominant hand. Placing the pen on the same object, while it is still selected with the NDH, triggers the fission of the system, which follows the business logic and implemented rules. In this particular case, this fission is the deletion of the slice.

5.5.2 Bimanual Asymmetric Seed Picking

Seed picking is the second elementary step in the workflow of tracing a horizon. Seismic interpreters spend most of their time in processing different slices and trying to pick seeds on the horizon under their consideration. The accuracy of the picked seeds is crucial for the 3D generation of the horizon slice. An approximation of the geological structures of the subsurface can only be achieved, if the picked seeds were accurate enough. Since picking seeds is one of the most time consuming steps including frequent mode switches, a bimanual asymmetric interaction technique seems to be a promising solution to overcome the limitations of the traditional based workflow presented in previous sections (see „4.4.2 Core Features“).

The demonstration of the traditional based workflow exposed where the tradeoffs for the seismic interpreters are. For a successful tracing of a horizon slice, seismic interpreters must be able to pick seeds with an input device, supporting pixel accurate inputs. Section „5.3.1 Properties shared by Pen and Touch“ showed that the pen input device is able to realize the required specifications. In addition, the usage of conventional devices for interaction techniques, which require a fine-granularity, can lead to the repetitive strain injury (see section „3.4.1 Pen Input“). The picking of seeds is a good example for an interaction technique requiring fine-granularity. Thus, the usage of a pen can prevent an uneven distribution of the muscular tension caused through repetitive movements and static loads. Furthermore, research studies [@Eur] verified that the pen has proven itself superior in terms of reaching ergonomical comfort compared to conventional devices. Using the pen for seed picking gives reasonable suspicion for a more seamless and ergonomical designed workflow.

Besides the requested fine-granularity, section „4.4.2 Core Features“ clarified that seismic interpreters continuously need to have a good perception of their currently edited overall seismic event. As illustrated, interpreters simultaneously want to pick seeds with high accuracy but also retain a good perception of their currently edited event. In the interaction with OpendTect, this results in frequent switches between the interactive mode and the view mode, since a user has to adjust the view several times in this workflow. This drastically harms the user experience. Users have to leave their point of interest to switch between modes, which is typically done on the Command Window. On the other hand, picking seeds is done on the Graphics Window. This need for switching between the Command Window and the Graphics Windows is a drawback for the cognitive resources of the user and is the reason for integrating a more seamless workflow into the picking of seeds. The aim must be to retain fine-granularity, while simultaneously providing a suitable perception of the seismic data. The solution to overcome this drawback is the implemented *lens-mode*. A user has the ability to temporarily open a magnifier view of a particular region of interest of the seismic data. The opened magnifier view shows the details of the seismic data under consideration and allows manipulating this data. When finishing the manipulation a user simply leaves the lens mode and returns to the default mode of the system including all virtues.

To realize a rewarding user experience the roles of the DH and the NDH are closely correlated with each other. With respect to the Kinematic Chain Model by Guiard [Gui87], the implemented lens-mode complies with all claims of a bimanual asymmetric interaction technique. Moreover, the parallel and combined use of the modalities results in a synergistic fusion of the input modalities in accordance to the CASE-model. The interaction workflow of the lens-mode can be described in terms of the main principles of bimanual asymmetric interaction techniques proposed by the Kinematic Chain Model (see section „3.1 Foundation of Bimanual Interaction“ for reference).

Right-to-Left Spatial Reference in Manual Motion

By touching a slice with the NDH, the NDH opens a magnifier view, in which the DH is able to insert content. That is, the NDH dynamically adjusts the frame in which the DH inserts content using the pen (see Figure 5.9a).

Left-Right Contrast in the Spatial-Temporal Scale of Motion

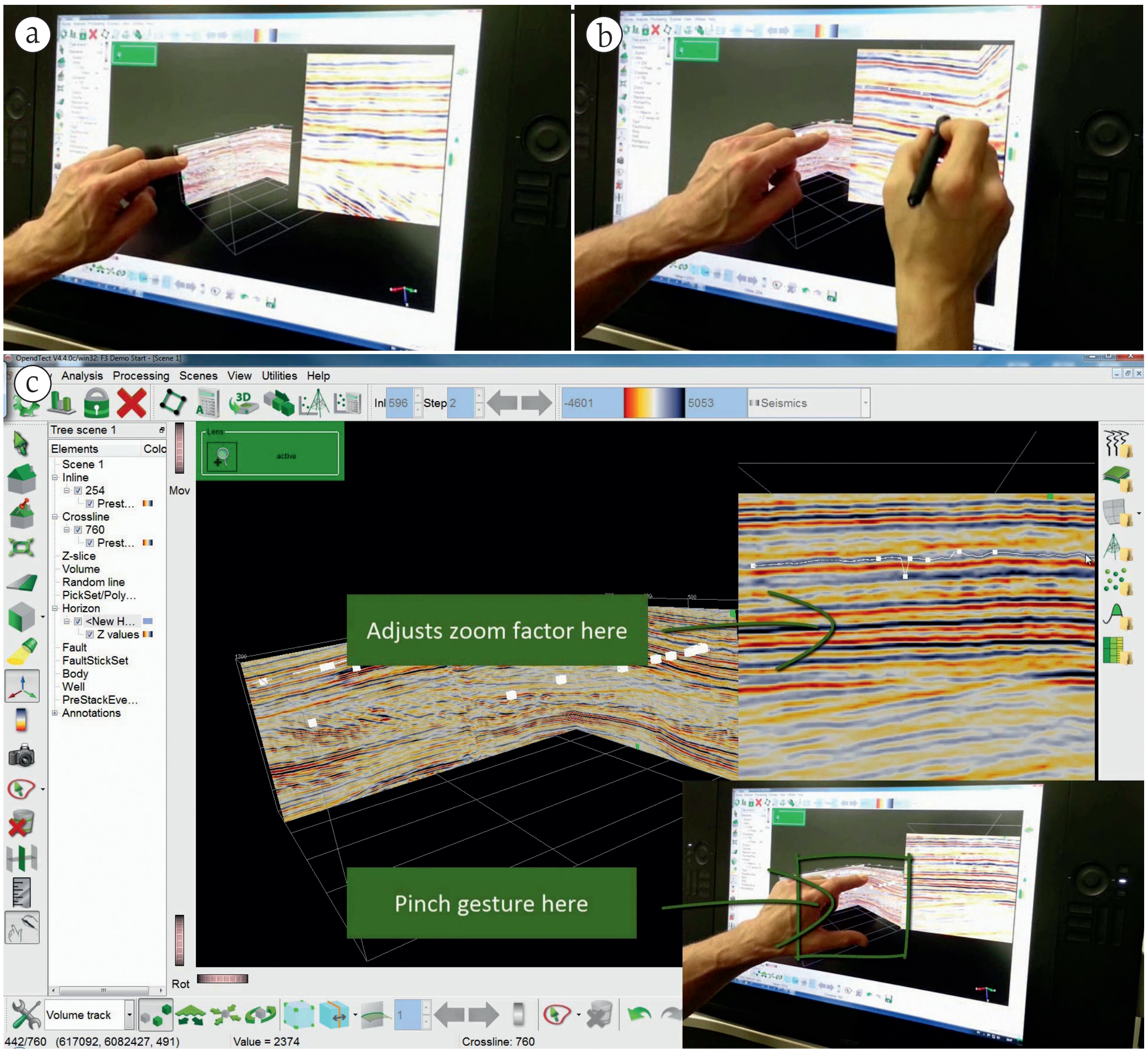
While touching a slice opens the magnifier view, moving the NDH on the slice adjusts the content of the magnifier view. That is, the NDH does the coarse manipulation, while the DH using the pen picks seed points incorporating finer control of movements (see Figure 5.9b).

Left-Hand Precedence in Action

This multimodal bimanual asymmetric interaction technique must be initiated using the NDH hand by touching a slice and thus invoking the lens-mode.

The ability to dynamically adjust the content of the magnifier view retains the perception of the currently edited event. The user gets the impression of following the seismic event across the seismic volume, while simultaneously being able to focus on identifying anomalies in the analyzed data. The arising opportunities are becoming clear when more slices are incorporated, since the magnifier view will always display a detail view of the seismic data specified with the NDH. The same applies, if the NDH moves across slices. For example, sliding from a crossline over to an inline will cause the magnifier view to change the orientation by following the NDH. The user is able to retrace the path of each seismic reflection, which reduces the cognitive load of tracking a horizon through the volume. Supporting this perception, the lens-mode allows the adjustment of the zoom factor of the magnifier view by invoking a pinch gesture with the non-dominant hand (see Figure 5.9c). A user is able to customize the magnifier view according to the tracked seismic events, her or his preferences or the context of use. Therefore, the system permits the requested guidelines of multimodal systems presented in section „3.2.3 Guidelines for Multimodal User Interfaces" by integrating modalities in a manner compatible with user preference, context and system functionality. Figure 5.9 summarizes the described interaction techniques available in the lens-mode.

Figure 5.9: Bimanual asymmetric seed picking. Invoking lens-mode by touching a slice with the NDH (a), picking seeds on magnifier view with using the pen in the DH (b) and adjusting the zoom factor of the magnifier view using a pinch gesture (c)



06 // Software Architecture

6 SOFTWARE ARCHITECTURE

The previous chapter introduced the principles of multimodal user interfaces evincing a high complexity in this challenge because of the cross-disciplinary of this research subject. Chapter „4 Novel Human Computer Interaction“ mentioned that there are abundant approaches of establishing multimodal user interfaces by combining pen and touch input. Notwithstanding this progress, yet there is little work done with attention on building software architectures or frameworks for novel types of interactive systems [Ech09]. Almost all of the described approaches are of prototypic nature and are not designed for integration into high professional-level desktop applications, like OpendTect.

The demand of novel interaction paradigms is twofold. First, the underlying hardware must support the required sensing capabilities and second the software must exploit these capabilities. Section „3.3 Interactive Displays“ exposed that already available technologies are not fully exhausted to leverage from their potential of filling the gap between the existing HCI techniques and the effective facility of novel interaction paradigms. The development challenge is manifested through the lack of software frameworks providing support in developing multimodal interactive systems. When designed well, such a middleware between the sensing hardware and the high level application development will greatly ease standard tasks for developers.

In this chapter, the implemented system architecture which has been developed to integrate a consistent model of novel interaction techniques into OpendTect will be described. In the beginning, the basic approach to overcome the lack of a multimodal software framework will be presented. For this, the ensemble of external libraries of OpendTect and the incorporated hardware, will be described, followed by the discussion of the concrete implementations of elementary development parts will be discussed.

Terms, which are specific to implementation entities, like class names, object names or functions are written in `typewriter font`.

6.1 Architecture Overview

Common traits of existing large interactive software systems are built on a stacked or layered architecture [EK08]. In this design, the complexity of the whole system is hidden by decomposing the system into a series of smaller and self-contained components. These components or layers are stacked together, with the one containing the least specific tasks at the bottom and the one responsible for the most specific tasks at the top. Only directly adjacent layers are able to communicate with each other, yielding a well encapsulated design of the architecture. This enables higher interchangeability of each layer with an alternative implementation without affecting the rest of the system. Widely known operating systems are also inheriting a layered architecture, with hardware drivers and the kernel at the bottom, libraries in the middle and end-user applications at the top of the layered system [Ech09].

In fact, the general layered architecture of OpendTect has been introduced in section „4.4.1 System Architecture". The underlying technical prerequisites for the implementation of this thesis were introduced in section „3.3 Interactive Displays" by introducing the Wacom Cintiq 24 HD Touch. This device natively supports the recognition and distinction between pen input and touch input. Nevertheless, for the fulfillment of the goals of this thesis there is only limited support. The following subsections will discuss these limitations and present the implemented solutions.

6.1.1 Unimodal Interaction

After delivery and installation of the Wacom Cintiq 24 HD Touch, a user is able to interact on a unimodal way either by using pen input or touch input. Figure 6.1 shows a generic view of a layered architecture concerning the processing of user input (left). In addition, Figure 6.1 (right) displays the concrete architecture of all Wacom components for the unimodal interaction processing in this thesis. The whole recognition of the user input, either using pen or touch, is done by the Wacom Cintiq 24 HD Touch and the driver of this device. In the following, this is referred to as Wacom. In later parts of this section, the concrete user input processing is described. For a more general description according to Figure 6.1 (left) please look at [EK08] and [Ech09].

The lowest layer (input hardware) consists of the Wacom Cintiq 24 HD Touch through accepting the user input. The main purpose of this layer is to generate raw tracking data of measurable input streams. These raw input streams are processed to generate position data. At this point, the position data is still in device coordinates and therefore the transformation layer converts this raw position data from device coordinates into screen coordinates. In this particular case, this is done twice. First, the underlying operating system handles the raw position data and adapts the position data to its own coordinate system. Then, the operating system identifies the most foreground application and forwards the processed data accordingly. In the concrete example of Figure 6.1 (right), the data is forwarded to OpendTect. As mentioned in section „4.4.1 System Architecture", the GUI toolkit Qt handles the communication with the operating system. Qt defines its own reference coordinate system and therefore implements its own processing for adapting the

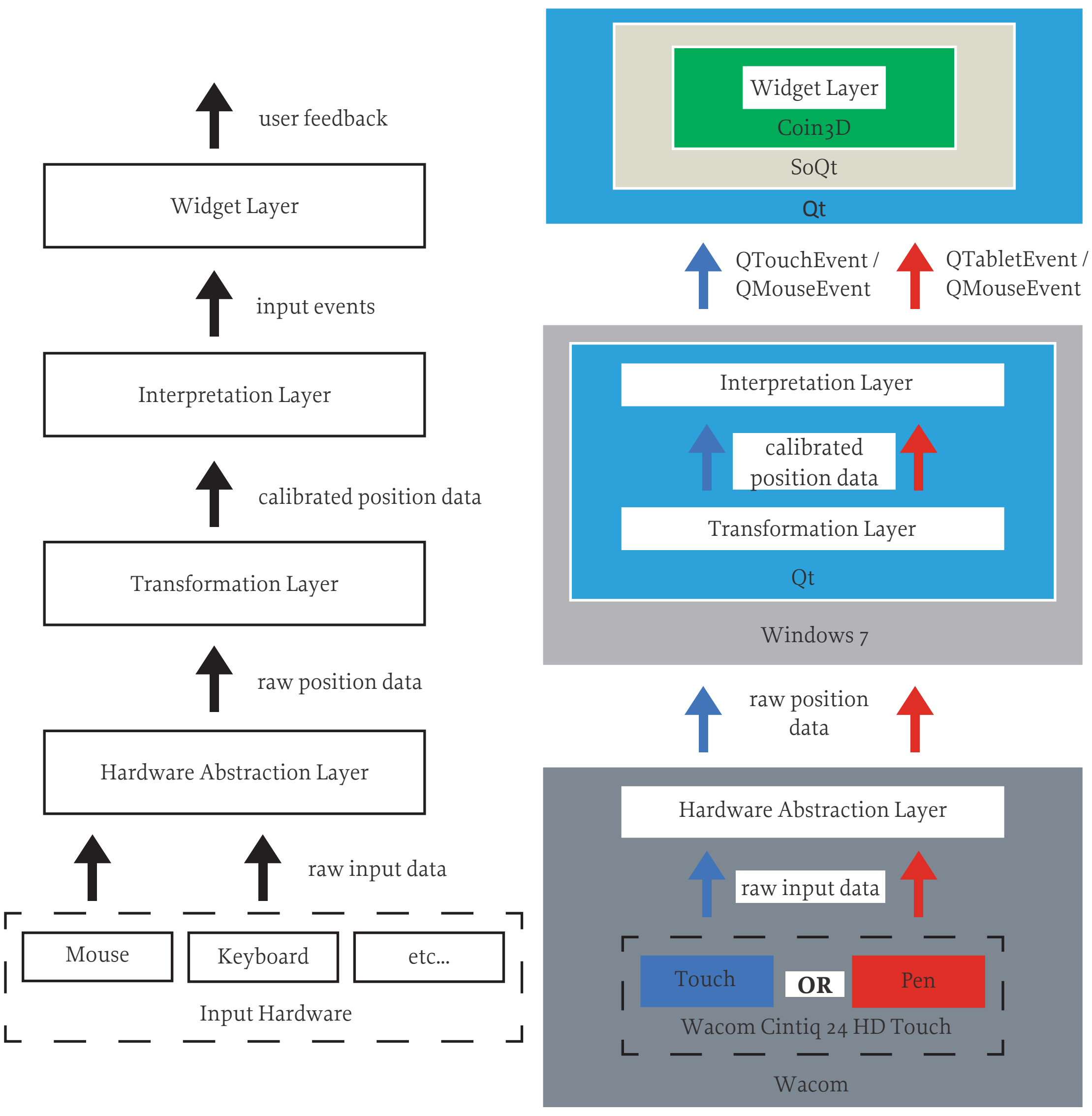


Figure 6.1: Generic view of the underlying layered architecture of user input processing (left) (based on [EK08]) and concrete view of the underlying architecture in this thesis (right)

forwarded position data to this coordinate system. Now, the data is ready for interpretation, which is done in the interpretation layer. With respect to the incorporated input modality, this interpretation can become computational expensive. For conventional devices (Figure 6.1. left) this is not the case due to deterministic input data [DLO09]. In contrast, multimodal interfaces incorporate continuous input devices (e.g. pen and touch, Figure 6.1 right), which require continuous processing of the input data. An example for this is the recognition of multitouch gestures like pinch or pan, which incorporates the interpretation of the input streams by probabilistic recognizers. For avoidance of computational overflow, the screen is divided into regions of interest; a generalization of the window concept used in common GUIs [EK08]. Each of these regions is able to define a set of gestures, for which it is interested in receiving input data. If the correct data occurs in one of these regions, a gesture will be triggered and passed along the process chain into the next layer (here widget layer). In Qt, these regions of interest can be defined in the base class of all user interface objects (`QWidget`). If a `QWidget` is allowed to receive touch and gestures events, Qt will process incoming position data accordingly. If the requirements are met, Qt will convert this data into a corresponding event (e.g. `QTouchEvent`) and send it into the widget chain. However, there are special considerations, when using an interactive display. Figure 6.1 (right) shows, that Wacom provides interaction using pen and touch input. This does not include mouse interaction. The caveats of this fact become obvious, with the approach of integrating multitouch interaction into a conventional desktop application. This kind of application consists of widgets that do not normally handle touch or gesture interaction. Without further processing, users would not be able to interact with the application using the touch or pen capabilities of Wacom. To overcome this limitation, Qt introduces a workaround. If a widget does not handle touch or gesture interaction and there is a valuable amount of incoming position data generated through touch interaction, Qt will use this data to simulate mouse behavior by generating a `QMouseEvent` for the first occurring touch event (see Figure 6.1 right). The same principle applies for incoming pen input data. This makes it possible, to use pen and touch individually in order to interact using mouse behavior.

As mentioned in the beginning of this chapter, there is a lack of software frameworks, supporting the development of multimodal interactive systems. Although, it might seem that the available interaction described above realizes the demanded support, unfortunately this is not the case. Rather than using one unified framework, the way Wacom supports pen and touch input is using two distinct libraries. The ensemble of the Wacom pen and touch architecture is displayed in Figure 6.2. The touch interaction is handled by the Wacom

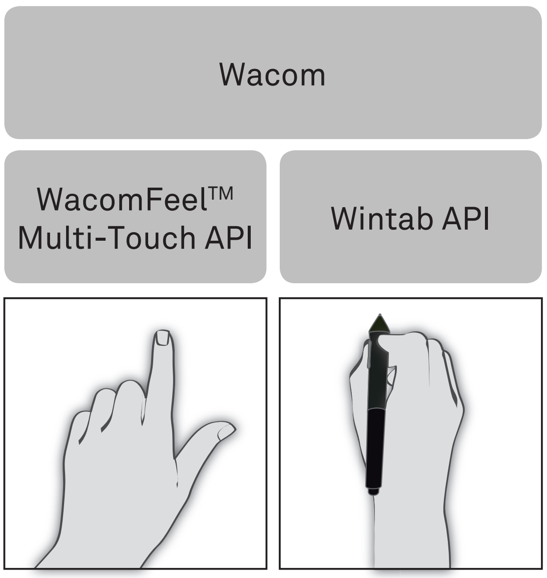


Figure 6.2: Wacom Feel™ Multi-Touch API and Wintab API realizing the handling of multitouch and pen interaction respectively

Feel SDK [@Waca], while the pen support it realized through the Wintab API [@Wacc]. In conclusion, it is natively possible to use the Wacom Cintiq 24 HD Touch to interact on a unimodal way. Users are able to use pen or touch input in their daily computer experience. Moreover, developers have the possibility to fully leverage the potential of each input modality when used individually. With focus on development of an application, which incorporates pen or touch in a unimodal way, developers are able to exhaust all the virtues of pen or touch input. Nevertheless, in fulfillment of the goals of this thesis, Wacom provides only limited support. As mentioned before, the supported interaction only applies to a unimodal setup. Unlike these advantages of unimodal interaction, multimodal interaction suffers from the deficiency of reasonable software support. The interplay of the Wacom ensemble to realize multimodal interaction is left to the application developer. The following subsection describes the work needed to implement a multimodal system architecture.

6.1.2 Multimodal Interaction

The previous subsection highlights the challenges of this thesis. As mentioned there, the incorporated hardware in this thesis, has inadequate support for multimodal interaction. This shortage is manifested, when trying to use pen and touch simultaneously. The problem is that the underlying operating system does not respond anymore to touch interaction, when the pen is already on or close to the display. The pen is higher in sensing priority than touch interaction. For example, touch interaction is immediately interrupted when the pen comes into the vicinity of the display. Thus, the application programmer cannot use the touch data in multimodal application development. Under consideration of implementing bimanual interaction techniques, based on simultaneous pen and touch interaction this is a major bottleneck. Figure 6.3 summarizes the illustrated restraint.

Although, the operating system does not respond to touch interaction, Wacom still delivers the raw touch data. Therefore, a solution has been implemented by the development of the PenTouch Plugin for OpendTect. The PenTouch Plugin can be loaded dynamically into the application at runtime and will handle all the user interaction. This approach makes it possible to use pen and touch input in a combined synergistic way. Thereby, a central logical component receiving inputs from various devices is implemented. This central component of the PenTouch Plugin is called `PenTouchProcessor`. The `PenTouchProcessor` maintains the interaction state and context of the application and coordinates the application flow. Apart from the `PenTouchProcessor`, the PenTouch Plugin encourages application management by delegation to further components decomposing the application management in various smaller subtasks. These delegated tasks are processing the touch data when the pen is on the display, handling of the interaction state and the fusion of pen and touch input. These subtasks include handling of several tasks described earlier for processing of user

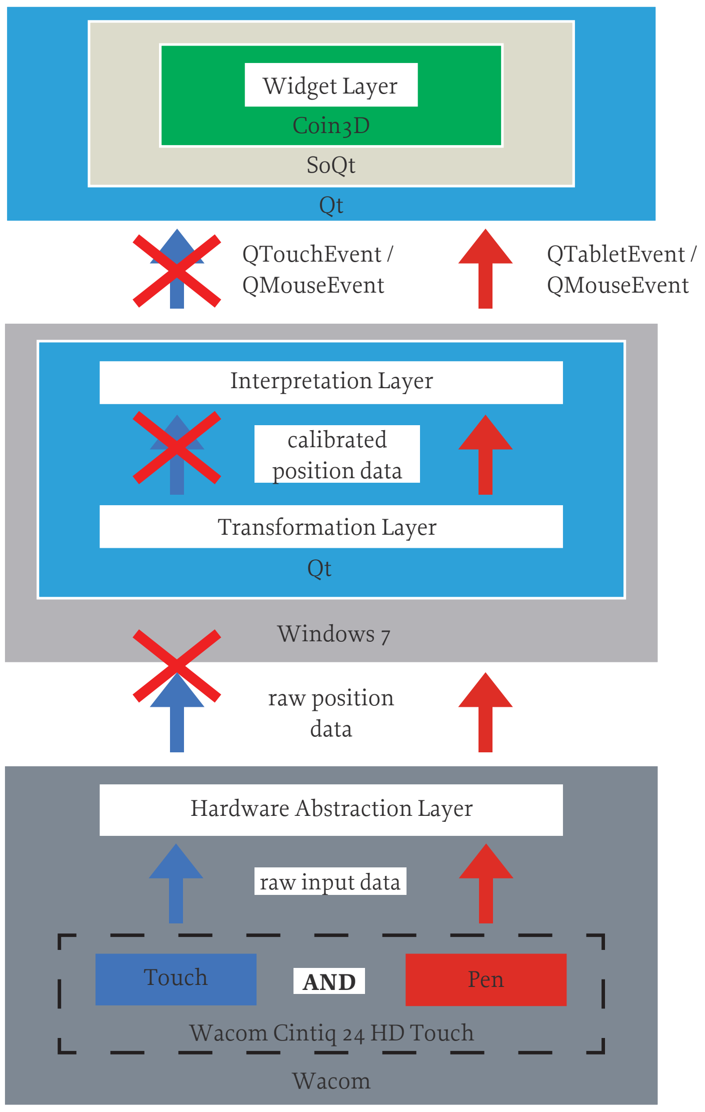


Figure 6.3: When the pen is in the vicinity of the display, the underlying operating system does not respond anymore to touch events

input in a layered architecture. Figure 6.4 demonstrates the way the `PenTouchProcessor` surmounts the lack of multimodal interaction support. The figure shows that the touch data is processed by the `PenTouchProcessor`, transformed into a data format compatible with the GUI toolkit Qt and forwarded directly to the application (OpendTect) into the widget layer. Thereby, the task of the transformation layer and the interpretation layer are included in the processing of the `PenTouchProcessor`, overcoming the pitfall of the touch unresponsiveness of the operating system.

The lack of native multitouch support during multimodal interaction adds more load to the development, since there is no standardized gesture engine that could be used to create sophisticated multitouch gesture interaction. In unimodal interaction, the operating

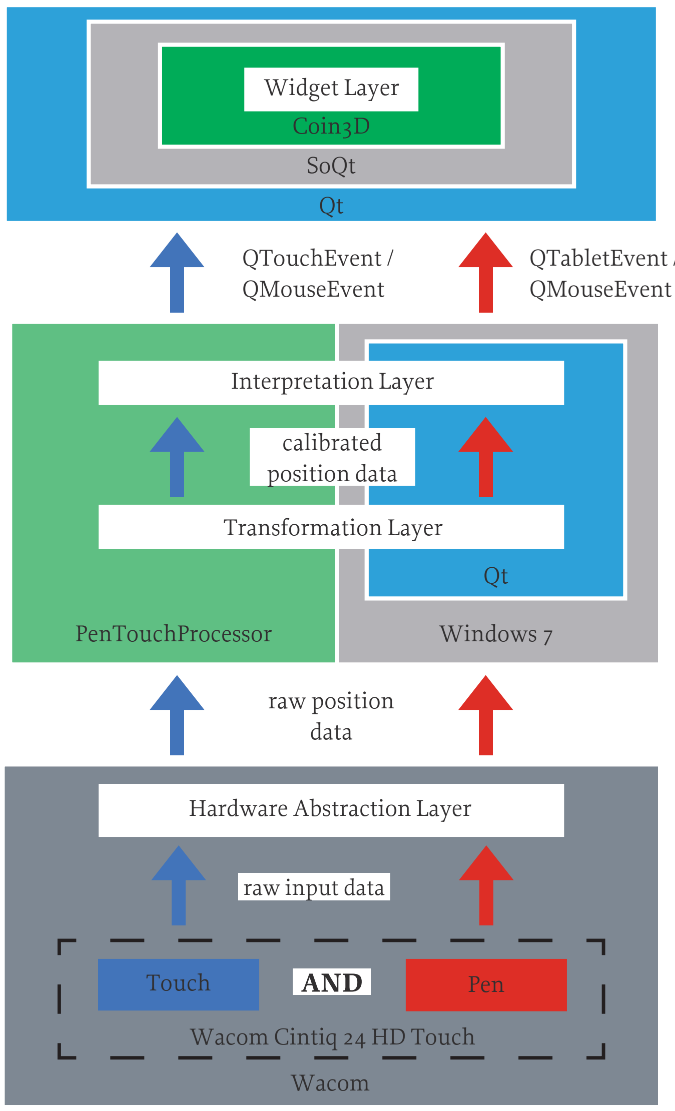


Figure 6.4: Multimodal pen and touch interaction. While the pen is still captured on the usual way, multimodality is realized through the PenTouchProcessor by handling the processing of touch input

system grants support for gesture recognition, therefore application developers are able to use this engine in their development to relay on gesture recognition. However, for integrating multitouch support in OpendTect a multitouch recognition engine is also part of this thesis. The implemented engine is capable of recognizing complex inputs such as pinch gestures, rotate gestures and pan gestures. The architecture makes it easy to focus on the implementation of new gestures in future development, since the adding of new gestures into the application is already supported by the PenTouch Plugin.

6.2 PenTouch Plugin

So far, it has been showed how the PenTouch Plugin enables the development of a multimodal system. Section „5.3.3 Mapping of Input Modalities“ described the design consideration for using pen and touch input chosen in this thesis. Research studies [HYP+10] verified, that users expect pen and touch input to behave the same way, when used on the Command Window. Therefore, the novel interaction techniques developed in this thesis were only applied, when the user is interacting on the Graphics Window. Section „4.4.1 System Architecture“ identified the corresponding components of OpendTect responsible for handling the interaction of the Graphics Window. These are the 3D rendering library Coin3D and the GUI-binding bridge SoQt.

This section is divided into two parts, each explaining one core entity of the implemented architecture. The first part includes the necessary steps to enable multitouch interaction and interaction based on the fusion of pen and touch into OpendTect. The second part presents the technical background of the implemented interaction handling.

6.2.1 Event Translation

SoQt provides automatic event handling, similarly to Qts translation of low-level window system events into higher-level `QEvents` (see previous section). SoQt translates these `QEvents` into Coin3D `SoEvents`, which can be used to manipulate objects in the 3D scenegraph. However, natively the library only supports the translation of mouse events, keyboard events and window resizing events. In conclusion, users are able to use the mouse or the keyboard to interact with 3D objects of Coin3D. There is no support for novel interaction based upon multitouch or the fusion of pen and touch input.

For the development of interaction based on these modalities, the required translation of corresponding events was implemented as part of this thesis. The architecture of the library makes it straightforward to support other input devices beyond mouse and keyboard. It is important to note that SoQt classes are not directly inheriting Qt classes. The integration with Qt is realized through delegation, rather than relying on inheritance to Qt. The result is a cleaner architecture and easier portability to other GUI toolkits. The creation of the rendering context is done internally by SoQt with instantiation of a QGLWidget for rendering. The objective of the developer is to implement a suitable translator. Figure 6.5 demonstrates the implementation required to enable multitouch support in the scenegraph of Coin3D and thereby in OpendTect. With this implementation a user is able to interact using multitouch gestures on the Graphics Window. An analog approach is used to realize the fusion of pen and touch input. The process can be summarized as follows. The starting point is the transformed and interpreted input position data described in the previous

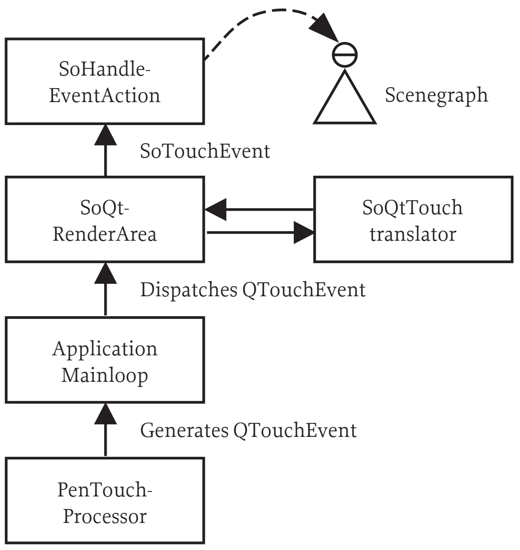


Figure 6.5: Translation of QTouchEvent into SoTouchEvent to support multitouch interaction in the scenegraph of Coin3D (based on [Wer94b])

section. This data is represented through a `QTouchEvent`, which is added to the event queue of the application. All events that are stored in this queue are sent to the corresponding application component. In this particular case, this component is the Graphics Window of OpendTect. The SoQt class representation of this Graphics Window is the `SoQtRenderArea` (see Figure 6.5). For handling of custom events, the `SoQtRenderArea` allows to register a custom translator (`SoQtTouch` translator). When a custom event occurs, the `SoQtRenderArea` translates these events into `SoEvents` by delegation to the according translator and passes them to the `SceneManager`. The `SceneManager` handles the event processing by creating an instance of the `SoHandleEventAction`. `SoHandleEventAction` objects are »smart« nodes implementing Coin3Ds mechanism for automatic event handling by traversing the nodes of the scene graph [Wer94a]. The traversing of the scenegraph stops, when a node has successfully handled the event.

The above described translation workflow only includes a simplified version of the actual implementation. OpendTect introduces a new application layer, whenever services from other libraries are used (see Figure 4.9). For each of the involved libraries (Qt, Coin3D, SoQt), there are one or more layers introduced by OpendTect. This encapsulation of external services makes it easier to replace one of the external libraries, since the dependency of application components is reduced by maintaining them at a central place. In addition, it is often the case that external services have been created for much more general purposes. This results in an enormous set of tools, of which only a fraction is of interest for the application programmer. One drawback of this insulation is that already existing knowledge about specific services of a library might not directly be available during development without deeper change of the underlying architecture. For example, OpendTect hides the Signal-Slot mechanism of Qt, which is Qts mechanism for realizing loosely coupled application components (see section „4.4.2 Core Features“).

6.2.2 PenTouch State Machine

The mapping of input modalities (see section „5.3.3 Mapping of Input Modalities“) brings up the question how to implement the different behaviors with respect to the utilized input modalities. From a technical point of view, a solution to this would be to implement multipart conditional statements depending on the kind of input. This would result in a large number of operations, where each will contain a similar conditional structure. Whenever a new operation should be integrated, more conditional statements would be necessary. This applies for each component interested in supporting this operation. With regard to maintainability and robustness of a system, this would result in high development cost and code duplication. The reason for this is the distribution of the implementation logic in all affected components.

A way to overcome this is to treat the application behavior as object itself and encapsulate it in a separate class. This way, the behavior of the application defines the state in which the application currently is. Gamma et al. [GHJV95] call the technical implementation of this approach State Pattern. The State Pattern encapsulates each branch of conditional statements into a separate class. The presented `Tool` interface in section „3.2.4 Fusion of Input Modalities“ is a concrete example of the State Pattern, since each `Tool` defines its own behavior and therefore sets the state of the application.

However, the introduced `Tool` interface alone does not suffice for a seamless and consistent workflow since it only covers the way pen and touch input are combined. While integrating novel interaction techniques into OpendTect, the conventional way of interaction must be prepared. Thus, there are several further considerations that have to be taken into account. The foundation of the interaction is defined through the mapping of input modalities. Therefore, it must be considered where the interaction takes place in the application. Furthermore, the interaction does not apply for all data types of OpendTect. The incorporated data type in the horizon tracing workflow are only forming a fraction of all available data types in OpendTect. Thus, a determination of the currently manipulated data type is inevitable. Based upon the natural characteristics of touch interaction, the basic design considerations of Hinckley et al. [HYP+10] suggest that unimodal touch interaction should be used for frequently used core tasks. The course of bimanual asymmetric interaction defined in the Kinematic Chain Model [Gui87] necessitates an earlier contribution of the NDH in a compound action of both hands. The caveats of this were identified by Dumas et al. [DLO09], who claim that the order in which input of the modalities in multimodal systems

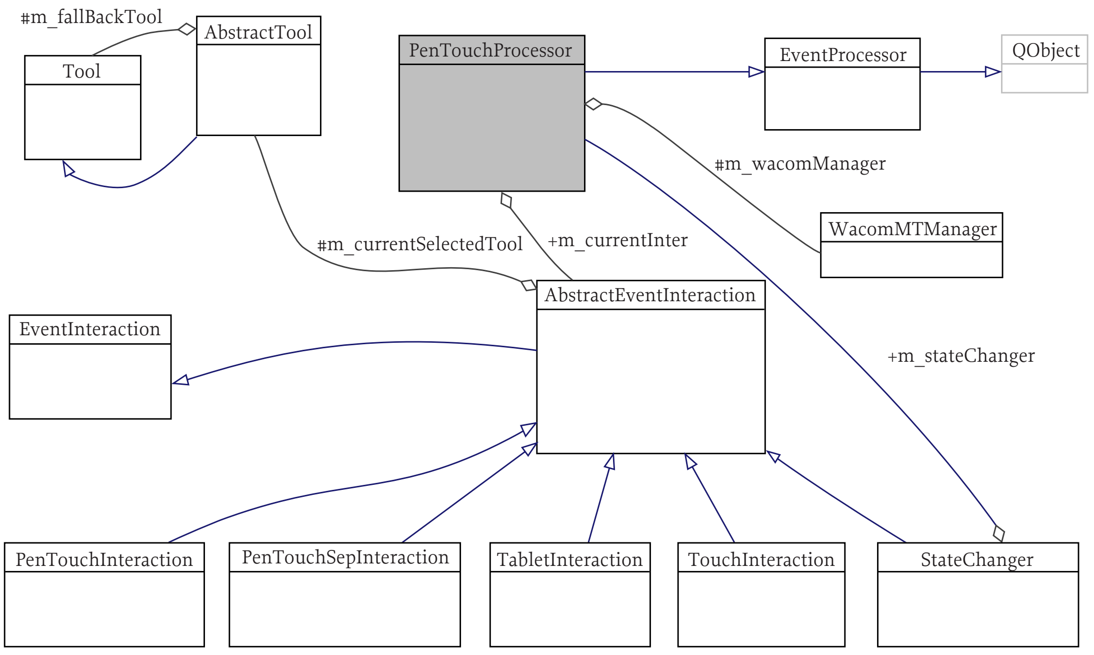


Figure 6.6: Collaboration diagram of the central implementation entities

appear plays a major role. The result is a time-sensitive architecture. In addition, the two different modes of OpendTect, View Mode and Interactive Mode (see section „4.4.2 Core Features“) also contribute to the design considerations. Some functionality (e.g. picking of objects) is only available in Interactive Mode and other only in View Mode respectively.

Section „6.1.2 Multimodal Interaction“ elucidates the software architecture of the application management into multiple smaller subtasks, each handled by an according component. Figure 6.6 shows a simplified diagram of the implemented components and their relations. This architecture realizes a higher control of the application workflow with respect to the listed considerations above. One part of this is the processing and translation of the raw touch data to realize multimodal interaction. This task is delegated to the `WacomMTManager` (see Figure 6.6). Another part of the distribution is the introduced `Tool` interface. The algorithms realizing the combination of pen and touch are encapsulated in the tools package using the State Pattern. However, there is also a mechanism necessary for deciding if all the prerequisites for the fusion of pen and touch input are met. Thus, to realize a persistent flow of the application another interface according to the State Pattern is introduced. This is the `EventInteraction` interface, which applies earlier in the user input processing chain. The `PenTouchProcessor` (see Figure 6.6) delegates the processing of all incoming user events to a concrete instance of the `EventInteraction` interface. This interface interprets the context of the user input and the order in which the input modalities are used and coordinates the transition from one interaction state to the other.

For example, if the user starts the interaction with using the pen in the DH, the concrete `EventInteraction` (here `TabletEventInteraction`) permits further touch interaction. Thus, if started with the pen all followed touch interaction will be discarded by the system. In contrast to this, if the user starts with touching the screen, the interaction is delegated to another instance of the `EventInteraction` interface (here `TouchInteraction`). Now, placing the pen on the display will cause the system to execute a further delegation (here `PenTouchInteraction`). In this state, fusion of pen and touch is possible. This time-sensitive architecture contains a consistent interaction flow, both, from a user-centered view and a system-centered view. If the user wants to combine pen and touch, he must initiate the interaction using touch interaction, exactly as provided by Guiards [Gui87] main principles of bimanual asymmetric interaction. This promises for a more seamless workflow by decreasing the cognitive load of the user. If started with the pen, fusion of input modalities is not provided. Visual cues are supplementing this perception. From a system-centered view, the complexity of the application state interplay keeps maintainable.

To keep the application state maintainable, each instance of the `EventInteraction` interface represents a state in the interaction state machine. This state machine also includes the `Tool` interface. The core process chain is as follows: The currently active `EventInteraction` instance decides whether or not to accept the user input. It delegates the request to the currently selected `Tool`. The `Tool` interface is responsible for deciding whether a transition can be made from one state to another by defining constrains to the selected data type of OpendTect. If these constraints are met, the concrete implementation of the currently active `Tool` is executed. If these constraints are not met, the `Tool` delegates this request to its fallback `Tool` (see „5.4 Seamless Mode Switching“ for reference), which executes the same verification according to its own constraints. This mechanism guarantees a seamless switching between modes. In most cases, the user is able to manipulate the view of the scene, without the need of an explicit mode switch.

07 // System Evaluation

7 SYSTEM EVALUATION

This section presents the user study to evaluate the novel interaction techniques. First, a general look on the purposes of system evaluations is provided. Afterwards, the chosen design and the course of this study will be described, followed by the results of the user test. Finally, an evaluation of the collected user feedback during the study is presented.

7.1 Purpose

Generally, the purpose of user interface evaluations is the analysis, assessment and testing of the entire UI or parts of it, such as specific input devices or interaction techniques [BKLP04]. The aim is to identify misleads in the UI or interaction design. Once identified, these misleads can be taken into account in future implementations and thus evaluated in further user studies. This iterative process guarantees that new designs can start from an informed position, rather than defining everything from the beginning.

Throughout this thesis, guidelines were presented acquired from the analysis of already existent user studies. Some of these focus on gestural interfaces, such as [Saf08] and on guidance for multimodal interfaces, such as [RLL+04] and [SPH98]. Yet others investigated bimanual multimodal interaction techniques in different domains [Fri12]. This evaluation is taking the findings of these previous evaluations into account.

7.2 Objectives

At the time of writing this thesis, there are no evaluations of bimanual multimodal interaction techniques in the domain of seismic interpretation available. Therefore, a more general understanding of the usability of novel interaction techniques in this particular domain is of interest in the evaluation of this thesis. The results of this evaluation should serve as starting point for further user studies and thus pave the way for defining guidelines for the implementation of bimanual multimodal interaction techniques in the workflow of seismic interpretation, eventually also for commercially-available software products.

Since bimanual multimodal interaction techniques have not been evaluated in the seismic domain, the aim of this evaluation is to determine the usability of novel pen and touch based interaction techniques in this domain. Bowman et al. [BKLPO] identified *User Preference* as a suitable metric to measure the usability of a specific interaction technique.

This metric refers to the users' subjective perception of the interface during interaction including ease of use, ease of learning and user satisfaction. Therefore, the first objective in this evaluation is to determine if users perceive the implemented pen and touch based interaction techniques as superior when compared to a reference system, which is formed by the traditional desktop interaction techniques.

The second objective of this thesis is to evaluate the metric of *User Comfort* of pen and touch based seismic interpretation tasks. This is a metric, which is often evaluated in 3D user interfaces. The aim of such evaluations is to measure how comfortable users are when interacting with 3D devices, such as 6 DOF controllers. The separation between pen and touch input and the combinations of these input modalities provides also more degrees of freedom (2 DOF per finger and pen) then traditionally available in seismic desktop applications. However, the problem with these devices is that long usage may strain arm and tendons, when the interaction is wrong designed. Thus, the second objective should prove that the way the PenTouch Plugin incorporates pen and touch based interaction does not feel uncomfortable.

Rating scales provided on a questionnaire are reasonable approaches to quantitative measure the described objectives [BKLPO].

7.3 Design of the Study

In traditional based computer workflows, direct bimanual multimodal interaction techniques are usually non-existent. Thus, it can be assumed that users will not be aware of the proposal of novel interaction techniques. This was verified by Frisch [Fri 12], who investigated bimanual interactions in editing of node link diagrams. Although, his system offered bimanual interaction techniques, without further explanation, all but one user did not recognize these interaction possibilities. Therefore, the chosen elaboration of this thesis included an introduction and guidance to become familiar with the novel techniques. All gestures and modality combinations were explained to the participants and they were asked to repeat them during training tasks. After that, they were asked to interact with the system using the all available interaction techniques and thus, they could freely decide which interaction technique to apply. During the whole evaluation a pen was handed over to the participant. Thereby, they could freely decide which modality (pen or touch) to use.

Participants

Fifteen users participated in this evaluation. Two of them were female and all of the participants were in the age of 23 – 57 years. All of them were right-handed and had somehow experiences in handling of computer-aided engineering applications. Some of them reported to be confronted with such applications daily, while others said to use those applications occasionally. None of them were a novice user of CAE applications. Nevertheless, since this is a pilot study, the participants were not every day seismic interpreters, nor HCI experts. However, some of them had a reasonable background in the seismic domain.

Installation

For the evaluation the same device was used as for the development of the novel interaction techniques (Wacom Cintiq 24 HD Touch). Furthermore, the same OpendTect version as for development (4.4.0) was used and also the included seismic data set was always the same. Laboratories at the Fraunhofer IAIS served as venue. The user study was carried out at two different days. At the first day, 6 volunteers participated while the rest of the participants joined at the second day. Except for the room, the installation was the same. The user sat in front of the device equipped with the pen and were freely able to adjust their positions.

Measurement

A questionnaire to be filled out by the participants was developed concerning previous knowledge about CAE applications, personal data and computer usage. There were no time limits for the fulfillment of the user study. The average time per user took 17 minutes. During the whole period, user feedback was collected and users were observed while interacting with the system.

7.4 Procedure of the Study

At the beginning every participant was introduced to OpendTect, since the application was foreign to the majority of the volunteers. This included a presentation of the unimodal pen and touch capabilities of the Wacom Cintiq 24 HD Touch. Participants could practice both modalities for interacting with the system. Moreover, the conventional workflow of tracing a horizon was shown. This included a demonstration of the conventional creation and deletion of slices and the picking of seeds to trace a horizon slice. After becoming familiar with the system and the haptics of the interactive display, the novel interaction techniques were demonstrated. For this users were introduced to the basic mapping of input modalities. It was shown, that pen and touch on the Command Window was interchangeable, while each of them has a special meaning when used on the Graphics Window. After this the interaction with the in-place pie menu for executing mode switches was presented. Subsequently, the novel pen and touch interaction techniques were executed. After a demonstration how to invoke and execute a combined pen and touch gesture, users were able to experiment the interaction techniques on their own. Thereby, a comparison between the conventional slice manipulation and the pen and touch based slice manipulation was shown by performing each of those interaction techniques one after the other. The same applied for the picking of seeds. Using the lens-mode users were explicitly asked to navigate across slices to observe their comfortableness while doing so. During the complete time, users were able to ask questions and request a repetition of the demonstration of any novel interaction technique.

All participants were asked to complete the workflow of successfully tracing a horizon slice, like described earlier in this thesis. All implemented interaction techniques are included in this workflow. In particular, participants were asked to do a workflow consisting of the following distinct subtasks:

- T1 Create a slice at a desired location
- T2 Delete a slice
- T3 Trace a seismic event on one seismic section

At the end, participants filled out a questionnaire with 9 questions. They rated the available interaction techniques and the usability of the system on 5-point scales. Moreover, they were asked to assess the usability of the novel system when compared to the traditional desktop interaction. The questionnaire can be found in Appendix A. Figure 7.1 illustrates a participant at day one of the user test, while executing task T3.

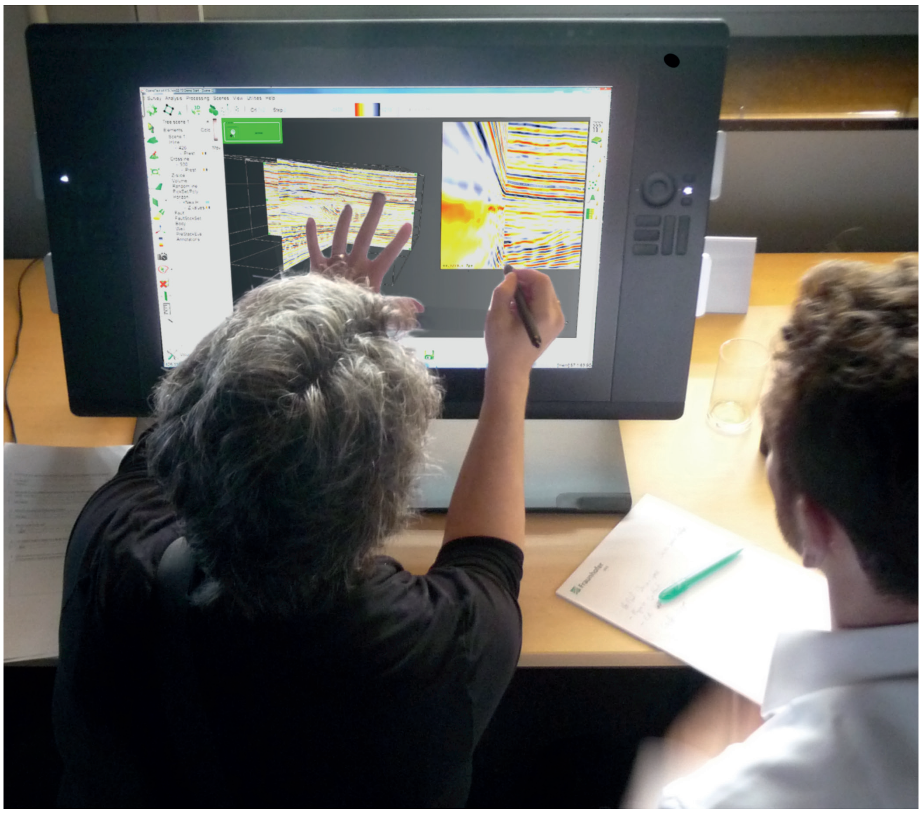


Figure 7.1: Single user during the picking of seeds using the lens-mode [@VRG]

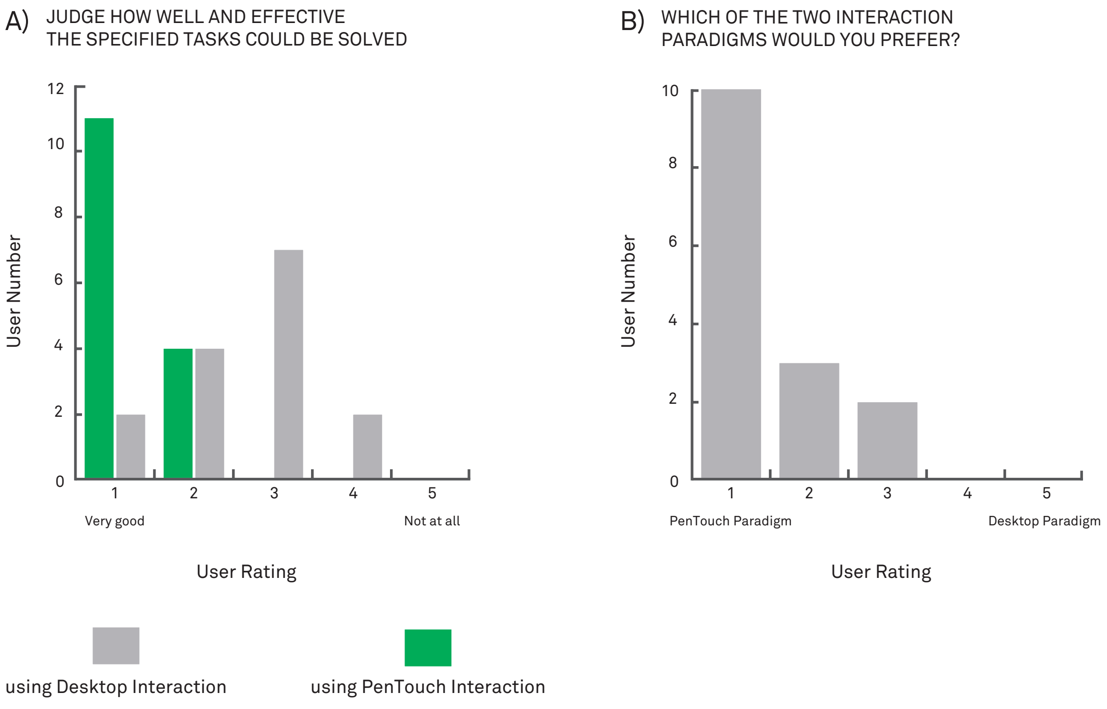
7.5 Results

The first objective of this evaluation was to provide a comparison of the user preference between the pen and touch based interaction techniques and the desktop interaction techniques for seismic interpretation. The second objective was to measure the user comfort during interaction with pen and touch based input devices while executing seismic interpretation tasks.

As the results of the usability questionnaire showed, all participants preferred the pen and touch interaction over the conventional desktop interaction. When asked how well and effective the participants were able to accomplish the specified tasks listed in the previous section, 73% of them stated that the task could be solved »very good« using the pen and touch based interaction techniques. Compared to this, only 13% of the participants specified the same when using the traditional desktop interaction. The diagram in Figure 7.2a illustrates a comparison of the user ratings when asked to specify how well they were able to accomplish the requirement tasks using the different interaction techniques.

Moreover the diagram Figure 7.2b shows, that 66% of the participants would prefer to use pen and touch based interaction techniques during the interpretation of seismic data.

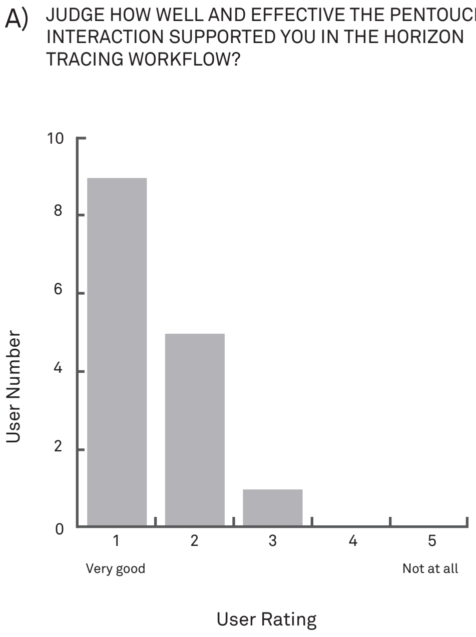
Figure 7.2: Bar chart measuring metrics of user preference of the PenTouch interaction techniques and the reference system (Desktop based interaction)

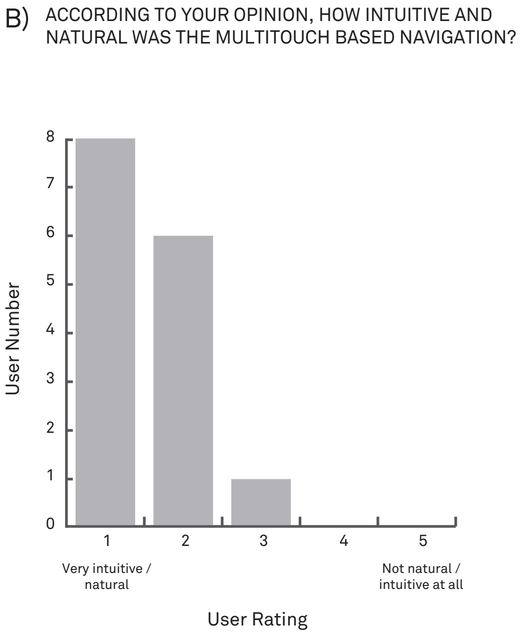


According to the second objective of this evaluation, the questionnaire included questions with focus on the user comfort during the interpretation using pen and touch devices. The majority of the participants explained that they were well supported by the pen and touch interaction during the tracing of a horizon and that this task could easily be solved once they became familiar with the system. Both, more practiced and also novice users of CAE applications, judged the system as satisfactory and effective. There were no differences according to expertise level. Figure 7.3a illustrates the ratings of the users with respect to their opinion regarding the support of the pen and touch based interaction techniques during the horizon tracing. Moreover, the multitouch navigation on the Graphics Window of OpendTect was reported to be very »intuitive« (see Figure 7.3b) and easy to remember. Users had fun adjusting the view using commonly known multitouch gestures. The results of the user comfort measurement showed that pen and touch based interaction supports participants during tasks for seismic interpretation. Although, some of these tasks require high-precision movement of the finger and hands, the way the PenTouch Plugin realizes multitouch interaction is described as »natural« and »intuitive«. No user stated to feel strained through the incorporation of multiple input devices.

Although the distinction between pen and touch input was mainly classified as very natural, participants felt unaccustomed about the difference. The same applies for the novel pen and touch based interaction techniques. Once users had understood the interaction techniques,

Figure 7.3: Measurement of user comfort realized by pen and touch based interaction techniques





they were confirmed by their effectiveness. However, Frisch's study [Fri12] foreshadowed that bimanual multimodal interaction is a novel interaction paradigm for most of the participants. This applies even more to this setup. Frisch used a tabletop application written for multimodal interaction from scratch, in which users are not accompanied by already existent knowledge. Here, an already existent application and workflow based on the widely known desktop metaphor was used. It can be assumed that users are reluctant when interacting with familiar applications including completely new interaction techniques. In all, users perceived the combination of pen and touch as acceptable and beneficial and thus further investigations are eligible.

7.6 User Feedback

Throughout the whole user study notes were taken to collect as much user feedback as possible. In addition, every participant had the possibility to express her or his emotion and suggestions using extra separated space at the end of the questionnaire. This resulted is a large list of valuable feedback, from which the most noteworthy are mentioned here.

The leverage of the pen capabilities and touch capabilities of the Wacom Cintiq 24 HD Touch were acknowledged. Users liked the seamless switching between command mode (touch) and ink mode (pen) without the need of an explicit menu selection.

Most users like the seamless integration of both pen and touch input modalities into OpendTect and see a potential increase of productivity and efficiency in workflows by the combination of both modalities. Users reported the switching between modes as being »very fluent«. It was suggested to bring this kind of interaction to further platforms, not only regarding software but also regarding hardware. Participants would like to see a combination of the pen and touch interaction on mobile devices. In contrast to these advantages, some users stated that it took them a short learning period to get familiar to the new interaction techniques.

The integration of bare handed touch gestures for navigation was well accepted. Users liked the close relation of the implemented multitouch gestures to commonly known gestures from interaction with other touch enabled devices.

Almost all users were convinced by the enhancement of the horizon tracing workflow achieved with the introduction of the lens-mode. One participant described the lens-mode as »awesome«. Regarding the manipulation of slices, there are slightly different opinions. The current implementation requests a user to create an initial slice using the traditional mouse interaction of OpendTect. This was criticized as »bothering« and it was suggested to replace this conventional workflow with pen and touch combinations.

The interaction with the implemented in-place pie menu was not well accepted. Users described the interaction as »difficult« and »unknown«. Especially the missing possibility to constantly show the pie menu for mode switches was criticized. The muscular tension for keeping the pie menu opened was described as tedious. However, users honored the achievable workflow acceleration through the pie menu and one user described it as »innovative«.

08 // Conclusion & Future Work

8 CONCLUSION & FUTURE WORK

There are three major contributions of this thesis. First, the thesis established novel interaction techniques, based on the fusion of pen and touch input, as viable means to coordinate bimanual interaction in the professional content-creation and manipulation tasks of the seismic interpretation workflow. Second, the thesis presented a disciplined approach in fulfillment of an effective utilization of natural human motor skills by seamlessly integrating novel multimodal capabilities into a standard desktop application to benefit from pen and touch based interaction techniques. Finally, this thesis has shown that users prefer multimodal pen and touch based interaction techniques over traditional desktop based interaction techniques.

The literature review included in this thesis substantiates the lack of an effective utilization of acquired human motor skills, not only for the domain of seismic interpretation, but also for more general daily desktop computing. Despite the natural aptitude of humans to use multiple parallel ways to communicate with each other, the way humans broadcast information to a computer or interact in a computer-mediated environment has largely been limited to singularity. As computing devices permeate an ever-growing portion of the daily live, new interaction methodologies raised to meet the challenges of overcoming the limitations erected by today's mouse-and-keyboard paradigm. Multimodal interaction seems to be a promising solution to this. A framework for multimodal interaction inherits the potential to allow developers filling the gap between existing HCI techniques and the advantage of novel devices. A leverage of those devices speeds up development and simplifies standard tasks for developers. Software engineers and designers can collaborate to focus on development of multimodal interfaces with the goal of incorporating more human senses in daily desktop computing. The aim must be to make interaction tasks simpler and more fluid by reducing the need for many interaction modes and switches between them. Multimodal interaction is most effective when taking advantage of the innate qualities of natural human interaction. In general, if well-chosen in accordance to the capabilities of the incorporated modalities a reduction of the gulf of execution between the users' intention and the systems response is possible.

The developed plugin for the seismic interpretation framework OpendTect, called PenTouch Plugin, introduced interactive displays into the mice and keyboard dominated realm of desktop computer environments. The implemented algorithms demonstrated a way to fill the gap between existing human computer interaction techniques and the capabilities of novel devices. This resulted in a potential approach to be prepared for the forecasted commercial shift of using unified input and output devices caused by the emerge and success of mobile devices. Although encapsulated in one single plugin, the PenTouch Plugin demonstrates the benefits from a careful, analytic and empirical design process to achieve maximal effectiveness. The design method for novel pen and touch based interaction techniques presented in this thesis provides state-of-the-art solutions to control desktop application tasks. The principles behind this design method are well founded and based on an extensive literature review. The integration into an established framework proves its the real-world applicability. The implemented bimanual interaction techniques have been shown to be superior to their one-handed counterparts, as they increase parallelism, reduce mode-switching time and help users in perceptual challenging tasks. The conducted pilot user study was a first step to prove this and has increased the understanding of human factors involved in bimanual multimodal interaction. This knowledge will help interface designers better understand which tasks are suitable for multimodal interaction and how to best assign parameter control to the degrees of freedom of involved input modalities. In compliance with advocated design considerations for multimodal interfaces throughout this thesis, the implemented system architecture is modular, easy maintainable and extendable. Although currently depending on specific hard- and software requirements, the encapsulation of the main logic in an independent library not only allows its integration in other seismic interpretation applications, but also in other domains for computer-aided engineering. The underlying architecture of the PenTouch Plugin can act as an example to surmount the lack of a unified multimodal interaction framework for future works. It demonstrated how to benefit from pen and touch input under consideration of their usage in physical world interaction by selective mapping according to their capabilities regarding naturalness, adaptability and dexterity. Thus, the work of this thesis can be seen as a starting point for providing guidelines for future research in this area.

Future developments of the system itself can focus on the implementation of algorithms rather than the integration into an established framework. Moreover, novel techniques, such as those implemented in this thesis, can act as stimulus for other platforms too. For example, the implemented lens-mode is suitable for integration in a collaborative mobile usage scenario, where each user is equipped with an own device. A single user ought to be able to highlight a point of interest on her or his device, causing a magnifier view to be shown on a shared display increasing the interplay of the team while maintaining personal user preferences. However, the major issue, that has to be addressed in future work is the currently limitation of the implemented techniques to the workflow of horizon tracing. A consistent and rewarding user experience can only be achieved by offering similar interaction techniques for most or all tasks of the system. Currently, it is only possible to manipulate a small subset of all available data types in OpendTect. For example, the implemented interaction technique to copy a slice using a compound gesture of pen and touch should be adapted to further data types for a consistent ease of use of the system. In the end, disruptive and tedious workflows should be replaced with seamless techniques, which decrease the users' cognitive load. This was also suggested by several volunteers of the user study. Furthermore, the problem regarding the interaction with the in-place pie menu must be addressed. Users complained about a missing feature to constantly show the pie menu relieving the muscular tension for keeping the menu visible, which was implemented for the sake of acceleration. Although the majority of the participants honored the possible increase of productivity and efficiency of the in-place pie menu, future implementations should address the unconformities by compromising between acceleration and user satisfaction.

This thesis showed that the seismic interpretation workflow benefits from the versatility of novel pen and touch based interaction techniques. An approach that has not been done so far. In this regard, the implemented techniques meet the claims of an effective utilization of natural human motor skills and include the potential to be an integral part of future user interfaces beyond the domain of seismic interpretation.

Bibliography

BIBLIOGRAPHY

[Alb82] Alan E. Albert. “The Effect of Graphic Input Devices on Performance in a Cursor Positioning Task.” In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 26*.1 (1982), pp. 54–58.

[BFW+08] Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. “Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces.” In:*Proceedings of the working conference on Advanced visual interfaces.* AVI ’08. Napoli, Italy: ACM, 2008, pp. 154–161.

[BKLP04] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 3D User Interfaces: Theory and Practice. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004.

[BM86] W. Buxton and B. Myers. “A study in two-handed input.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI ’86. Boston, Massachusetts, USA: ACM, 1986, pp. 321–326.

[Bro04] A.R. Brown. *Interpretation of Three-Dimensional Seismic Data.* Aapg Memoir / Seg Investigations in Geophysics Nr. 42. Published jointly by The American Association of Petroleum Geologists and the Society of Exploration Geophysicists, Geological Society Publishing House, 2004.

[BRP05] Ragnar Bade, Felix Ritter, and Bernhard Preim. “Usability comparison of mouse-based interaction techniques for predictable 3d rotation.” In: *Proceedings of the 5th international conference on Smart Graphics.* SG’05. Frauenwörth Cloister, Germany: Springer-Verlag, 2005, pp. 138–150.

[BS06] Jasmin Blanchette and Mark Summerfield.*C++ GUI Programming with Qt 4.* Ed. by 2nd. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2006.

[BSR03] M. Bacon, R. Simm, and T. Redshaw.*3-D Seismic Interpretation.*Cambridge University Press, 2003.

[CMS88] Michael Chen, S. Joy Mountford, and Abigail Sellen. “A study in interactive 3-D rotation using 2-D control devices.” In: *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*. SIGGRAPH ’88. New York, NY, USA: ACM, 1988, pp. 121–129.

[CSH+92] Brookshire D. Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, and Andries van Dam. “Three-dimensional widgets.” In: *Proceedings of the 1992 symposium on Interactive 3D graphics*. I3D ’92. Cambridge, Massachusetts, USA: ACM, 1992, pp. 183–188.

[Dah06] Markus Dahm.*Grundlagen der Mensch-Computer-Interaktion*. Informatik : Software-Ergonomie. Pearson Studium, 2006.

[dAn12] David d’Angelo. *Touch & Pen - A fruitful combination*. Presentation at the VRGeo Meeting, Sankt Augustin, June 2012.

[dG12] David d’Angelo and Ömer Genç. *Pen + Touch Interaction - A fruitful combination (Update)*. Presentation at the VRGeo Meeting, Sankt Augustin, December 2012.

[dGB12] dGB Earth Sciences B.V. *Introduction to OpendTect V. 4.4*. dGB Earth Sciences B.V., 2012.

[DL01] Paul Dietz and Darren Leigh. “DiamondTouch: a multi-user touch technology.” In: *Proceedings of the 14th annual ACM symposium on User interface software and technology*. UIST ’01. Orlando, Florida: ACM, 2001, pp. 219–226.

[DLO09] Bruno Dumas, Denis Lalanne, and Sharon Oviatt. “Human Machine Interaction.” In: ed. by Denis Lalanne and Jürg Kohlas. Berlin, Heidelberg: Springer-Verlag, 2009. Chap. Multimodal Interfaces: A Survey of Principles, Models and Frameworks, pp. 3–26.

[Ech09] Florian Echtler. “Tangible Information Displays.” PhD thesis. Technische Universität München, 2009.

[EK08] Florian Echtler and Gudrun Klinker. “A multitouch software architecture.” In:*Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*. NordiCHI ’08. Lund, Sweden: ACM, 2008, pp. 463–466.

[EMA97] D.W. Eaton, B. Milkereit, and E. Adam. “3-D Seismic Exploration.” In:*Proceedings of Exploration 97: Fourth Decennial International Conference on Mineral Exploration.* 1997, pp. 65–78.

[FHD09] Mathias Frisch, Jens Heydekorn, and Raimund Dachselt. “Investigating multi-touch and pen gestures for diagram editing on interactive surfaces.” In:*Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces.* ITS ’09. Banff, Alberta, Canada: ACM, 2009, pp. 149–156.

[Fri12] Mathias Frisch. “Interaction and Visualization Techniques for Node-Link Diagram Editing and Exploration.” PhD thesis. Otto-von-Guericke-Universität Magdeburg, 2012.

[GHJV95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides.Design patterns: elements of reusable object-oriented software. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[GS04] J. Gluyas and R. Swarbrick.*Petroleum Geoscience.* Wiley, 2004.

[Gui87] Yves Guiard. “Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model.” In: Journal of Motor Behavior19.4 (1987), pp. 486–517.

[Har00] B.S. Hart.*3-D seismic interpretation: a primer for geologists.* SEPM short course. SEPM (Society for Sedimentary Geology), 2000.

[HCS98] Ken Hinckley, Mary Czerwinski, and Mike Sinclair. “Interaction and modeling techniques for desktop two-handed input.” In: *Proceedings of the 11th annual ACM symposium on User interface software and technology.* UIST ’98. San Francisco, California, USA: ACM, 1998, pp. 49–58.

[Hei12] A.M. Heinecke. *Mensch-Computer-Interaktion: Basiswissen Für Entwickler und Gestalter.* Springer, 2012.

[Hou92] Stephanie Houde. “Iterative design of an interface for easy 3-D direct manipulation.” In:*Proceedings of the SIGCHI Conference on Hu-man Factors in Computing Systems.* CHI ’92. Monterey, California, USA: ACM, 1992, pp. 135–142.

[HPP+97] Ken Hinckley, Randy Pausch, Dennis Proffitt, James Patten, and Neal Kassell. "Cooperative bimanual action." In:*Proceedings of the ACM SIGCHI Conference on Human factors in computing systems.* CHI '97. Atlanta, Georgia, USA: ACM, 1997, pp. 27–34.

[HSH04] Knud Henriksen, Jon Sporring, and Kasper Hornbäek. "Virtual Trackballs Revisited." In: *IEEE Transactions on Visualization and Computer Graphics10.2 (Mar. 2004)*, pp. 206–216.

[HTP+97] Ken Hinckley, Joe Tullio, Randy Pausch, Dennis Proffitt, and Neal Kassell. "Usability analysis of 3D rotation techniques." In:*Proceedings of the 10th annual ACM symposium on User interface software and technology.* UIST '97. Banff, Alberta, Canada: ACM, 1997, pp. 1–10.

[HYP+10] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. "Pen + touch = new tools." In:*Proceedings of the 23nd annual ACM symposium on User interface software and technology.* UIST '10. New York, New York, USA: ACM, 2010, pp. 27–36.

[JCG08] F. Jahn, M. Cook, and M. Graham. *Hydrocarbon Exploration & Production. Developments in Petroleum Science.* Elsevier Science, 2008.

[JGH+08] Robert J.K. Jacob, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. "Reality-based interaction: a framework for post-WIMP interfaces." In:*Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI '08. Florence, Italy: ACM, 2008, pp. 201– 210.

[JTSS94] Huw James, Mark Tellez, Gaby Schaetlein, and Tracy Startk. "Geophysical Interpretation: From Bits and Bytes to the Big Picture." In:Oilfield Review 6 (1994).

[KBH02] P. Kearey, M. Brooks, and I. Hill. *An Introduction to Geophysical Exploration. Geoscience texts.* Wiley, 2002.

[KFBB97] Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. "The design of a GUI paradigm based on tablets, two-hands, and transparency." In: *Proceedings of the ACM SIGCHI Con-ference on Human factors in computing systems.* CHI '97. Atlanta, Georgia, USA: ACM, 1997, pp. 35–42.

[Kin12] Kenrick Kin. “Investigating the Design and Development of Multi-touch Applications.” PhD thesis. University of California at Berkeley, 2012.

[LBE04] Anatole Lécuyer, Jean-Marie Burkhardt, and Laurent Etienne. “Feeling bumps and holes without a haptic interface: the perception of pseudo-haptic textures.” In:*Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI ’04. Vienna, Austria: ACM, 2004, pp. 239–246.

[LGF10] G. Julian Lepinski, Tovi Grossman, and George Fitzmaurice. “The design and evaluation of multitouch marking menus.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI ’10. Atlanta, Georgia, USA: ACM, 2010, pp. 2233– 2242.

[LHGL05] Yang Li, Ken Hinckley, Zhiwei Guan, and James A. Landay. “Experimental analysis of mode switching techniques in pen-based user interfaces.” In:*Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI ’05. Portland, Oregon, USA: ACM, 2005, pp. 461–470.

[ML89] Robert Mack and Kathy Lang. “A Benchmark Comparison of Mouse and Touch Interface Techniques for an Intelligent Workstation Windowing Environment.” In:*Proceedings of the Human Factors and Ergonomics Society Annual Meeting 33*.5 (1989), pp. 325–329.

[NC93] Laurence Nigay and Joëlle Coutaz. “A design space for multimodal systems: concurrent processing and data fusion.” In: *Proceedings of the INTERACT ’93 and CHI ’93 Conference on Human Factors in Computing Systems.* CHI ’93. Amsterdam, The Netherlands: ACM, 1993, pp. 172–178.

[Nor02] D.A. Norman. *The design of everyday things.* Basic Books, 2002, p. 156.

[OCW+00] Sharon Oviatt, Phil Cohen, Lizhong Wu, John Vergo, Lisbeth Duncan, Bernhard Suhm, Josh Bers, Thomas Holzman, Terry Winograd, James Landay, Jim Larson, and David Ferro. “Designing the user interface for multimodal speech and pen-based gesture applications: state-of-the-art systems and future research directions.” In: *Hum.-Comput. Interact.* 15.4 (Dec. 2000), pp. 263–322.

[Pos09] S. Poslad. *Ubiquitous Computing: Smart Devices, Environments and Interactions.* John Wiley & Sons, 2009.

[PWS88] R. L. Potter, L. J. Weldon, and B. Shneiderman. “Improving the accuracy of touch screens: an experimental evaluation of three strategies.” In:*Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’88. Washington, D.C., USA: ACM, 1988, pp. 27–32.

[RC91] D. Rubine and Carnegie-Mellon University. Information Technology Center. Specifying Gestures by Example. CMU-CS. Carnegie Mellon University, Computer Science Department, 1991.

[Rek02] Jun Rekimoto. “SmartSkin: an infrastructure for freehand manipulation on interactive surfaces.” In:*Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’02. Minneapolis, Minnesota, USA: ACM, 2002, pp. 113–120.

[RLL+04] Leah M. Reeves, Jennifer Lai, James A. Larson, Sharon Oviatt, T. S. Balaji, Stéphanie Buisine, Penny Collings, Phil Cohen, Ben Kraal, Jean-Claude Martin, Michael McTear, TV Raman, Kay M. Stanney, Hui Su, and Qian Ying Wang. “Guidelines for multimodal user interface design.” In: *Commun. ACM* 47.1 (Jan. 2004), pp. 57– 59.

[Saf08] D. Saffer.*Designing Gestural Interfaces*. O’Reilly Media, 2008.

[SBG+11] Hyunyoung Song, Hrvoje Benko, Francois Guimbretiere, Shahram Izadi, Xiang Cao, and Ken Hinckley. “Grips and gestures on a multi-touch pen.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’11. Vancouver, BC, Canada: ACM, 2011, pp. 1323–1332.

[SCS+11] Minghui Sun, Xiang Cao, Hyunyoung Song, Shahram Izadi, Hrvoje Benko, Francois Guimbretiere, Xiangshi Ren, and Ken Hinckley. “Enhancing naturalness of pen-and-tablet drawing through context sensing.” In:*Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ITS ’11. Kobe, Japan: ACM, 2011, pp. 83–86.

[Sho92] Ken Shoemake. “ARCBALL: a user interface for specifying three-dimensional orientation using a mouse.” In: *Proceedings of the conference on Graphics interface ’92*. Vancouver, British Columbia, Canada: Morgan Kaufmann Publishers Inc., 1992, pp. 151–156.

[SIM13] SIM - System In Motion. “Coin3D and Qt: The Solution for Cross-platform 3D Application Development.” In: (2013).

[SPH98] R. Sharma, V.I. Pavlovic, and T.S. Huang. “Toward multimodal human-computer interface.” In: *Proceedings of the IEEE 86*.5 (1998), pp. 853–869.

[SS05] A. Saxena and B. Sahay. *Computer Aided Engineering Design*. Anamaya Publishers, New Delhi, India, 2005.

[Sta11] T. Stapelkamp.*Interaction- und Interfacedesign: Web-, Game-, Produkt- und Servicedesign Usability und Interface als Corporate Identity*. X.media.press / publishing. Springer, 2011.

[Str91] David J. Struman. “Whole-hand Input.” PhD thesis. Massachusetts Institute of Technology, 1991.

[Wer94a] J. Wernecke. *Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor Release 2*. OpenGL Series. Addison Wesley, 1994.

[Wer94b] J. Wernecke. *Inventor Toolmaker: Extending Open Inventor TM, Release 2*. OpenGL Series. Addison-Wesley, 1994.

[WFB+07] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. “Lucid touch: a see-through mobile device.” In: *Proceedings of the 20th annual ACM symposium on User interface soft-ware and technology*. UIST ’07. Newport, Rhode Island, USA: ACM, 2007, pp. 269–278.

[WL06] R. Wilkinson and BHP Billiton Limited.Speaking Oil and Gas. BHP Billiton Petroleum, 2006.

[WSR+06] Mike Wu, Chia Shen, Kathy Ryall, Clifton Forlines, and Ravin Balakrishnan. “Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces.” In: *Proceedings of the First IEEE In-ternational Workshop on Horizontal Interactive Human-Computer Systems*. TABLETOP ’06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 185–192.

[XIW07] Xu Xia, Pourang Irani, and Jing Wang. “Evaluation of guiard’s theory of bimanual control for navigation and selection.” In:*Proceedings of the 2007 international conference on Ergonomics and health aspects of work with computers*. EHAWC’07. Beijing, China: Springer-Verlag, 2007, pp. 368–377.

[YD01a] Ö. Yilmaz and S.M. Doherty. *Seismic Data Analysis: Processing, Inversion, and Interpretation of Seismic Data*. Investigations in Geophysics Series Bd. 1. Society of Exploration Geophysicists, 2001.

[YD01b] Ö. Yilmaz and S.M. Doherty. *Seismic Data Analysis: Processing, Inversion, and Interpretation of Seismic Data*. Investigations in Geophysics Series Bd. 2. Society of Exploration Geophysicists, 2001.

[Yee04] Ka-Ping Yee. "Two-handed interaction on a tablet display." In: CHI '04 Extended Abstracts on Human Factors in Computing Systems. CHI EA '04. Vienna, Austria: ACM, 2004, pp. 1493–1496.

[ZM98] Shumin Zhai and Paul Milgram. "Quantifying coordination in multiple DOF movement and its application to evaluating 6 DOF input devices." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '98. Los Angeles, California, USA: ACM Press/Addison-Wesley Publishing Co., 1998, pp. 320–327.

Online Resources

[@Ali] Alias. *Alias Help*. url: http://www.autodesk.com/techpubs/aliasstudio/2010/images/ALIAS/TenFidy/English/LTSConventions/LTSFoundations2_SomeCommonBasics/markingMenu.png (visited on 05/18/2013).

[@Ano] Anoto. *Anoto-Digital Writing Solutions*. url: http://www.anoto.com/lng/en/pageTag/page:home/ (visited on 03/24/2013).

[@Appa] Apple. *Apple - iPad 2 - View the technical specifications for iPad 2*. url: http://www.apple.com/ipad/ipad-2/specs.html (visited on 07/18/2013).

[@Appb] Apple. Apple - OS X Mountain Lion - Move your Mac even further ahead. url: http://www.apple.com/osx/ (visited on 07/22/2013).

[@Appc] Apple. *Apple Special Event March 2011*. url: http://events.apple.com.edgesuite.net/1103pijanbdvaaj/event/index.html (visited on 07/18/2013).

[@Asi] Wacom Asia. *Wacom Pens: The ergonomic way to use your computer*. url:http://www.wacom-asia.com/rsi/img/rsi_ap.pdf (visited on 04/01/2013).

[@Aut] Autodesk AutoCAD. *AutoCAD | 3D CAD Design Software | Autodesk*. url: http://www.autodesk.com/products/autodesk-autocad/overview (visited on 04/26/2013).

[@Bux] Bill Buxton. *Multi-Touch Systems that I Have Known and Loved*. url: http://www.billbuxton.com/multitouchOverview.html (visited on 07/09/2013).

[@CMa] CMake. *CMake - Cross Platform Make*. url: http://www.cmake.org/ (visited on 04/03/2013).

[@COM] COM-ON. *Wacom Cintiq 24 HD touch und Intuos 5 touch*. url: http://com-on-online.com/wacom-cintiq-24-hd-touch-und-intuos-4-touch/ (visited on 07/05/2013).

[@Cor] Corel. *Digital Art Software - Corel Painter12* url:http://www.corel.com/corel/product/indexjsp?pid=prod4030123&cid=catalog20038&segid=78&storeKey=us (visited on 04/26/2013).

[@dGB] dGB Earth Sciences. *About us*. url: http://www.dgbes.com/ index.php/about-us.html (visited on 05/18/2013).

[@Diga] Digia. *Qt*.url:http://qt.digia.com/ (visited on 05/19/2013).

[@Digb] Digia. *Signals & Slots | Documentation | Qt Project*. url: http://qt-project.org/doc/qt-4.8/signalsandslots.html (visited on 05/19/2013).

[@Dor] Bob Dormon. *Samsung Galaxy Note 2 hands-on review*. url: http://www.theregister.co.uk/2012/09/04/ifa_2012_samsung_galaxy_note_2_hands_on_review/ (visited on 05/03/2013).

[@Eur] Wacom Europe. *Studie der TU Darmstadt untersucht Stifttablett als ergonomische Alternative zur Maus am Computer*. url: http://www.wacom.eu/int/use-it/ergonomics/download/Ergonomics_report_de.pdf (visited on 04/01/2013).

[@Fra] Fraunhofer IAIS. *Fraunhofer IAIS: Fraunhofer IAIS*. url: http: //www.iais.fraunhofer.de/ (visited on 04/03/2013).

[@Hol] Christian Holz. *Modelling touch input and making it really accurate*. url:http://www.dcs.gla.ac.uk/~darylw/christianslides.pdf (visited on 07/12/2013).

[@Kona] Kongsberg Oil & Gas Technologies. *Coin: Main Page*. url: http://doc.coin3d.org/Coin-3.1/ (visited on 05/19/2013).

[@Konb] Kongsberg Oil & Gas Technologies. *Coin3D/Coin/wiki/IntroductionToCoin3D*. url:https://bitbucket.org/Coin3D/coin/wiki/IntroductionToCoin3D (visited on 05/20/2013).

[@Konc] Kongsberg Oil & Gas Technologies. *SoQt: Main Page*. url: http://doc.coin3d.org/SoQt/ (visited on 07/05/2013).

[@Lan] Landmark Hallibuton. *DecisionSpace Platform*. url: https://www.landmarksoftware.com/Pages/DecisionSpacePlatform.aspx (visited on 07/26/2013).

[@May] Autodesk Maya. *Maya | 3D Animation Software | Computer Animation | Autodesk*. url: http://www.autodesk.com/products/autodesk-maya/overview (visited on 05/11/2013).

[@Mica] Microsoft Research. *Pen-and-Touch Interaction for Touch-Screen Displays of All Sizes*. url: http://research.microsoft.com/ apps/video/default.aspx?id=173722 (visited on 04/27/2013).

[@Micb] Microsoft Windows. *Windows 8 - Microsoft Windows*. url: http://www.netmarketshare.com/ (visited on 07/18/2013).

[@Mud] Autodesk Mudbox. *Digital Painting and Sculpting Software | Mudbox | Autodesk*. url: http://www.autodesk.com/products/mudbox/overview (visited on 04/26/2013).

[@Mul] Laboratorio Multimediale. *3D Rotation User Interfaces*. url: http: //medialab.di.unipi.it/web/IUM/Waterloo/node57.html (visited on 05/17/2013).

[@Net] NetMarketShare. *Market share for mobile, browsers, operating systems and search engines | NetMarketShare*. url: http://www.netmarketshare.com/ (visited on 07/18/2013).

[@Opea] OpendTect. *Appendix D. Wacom Digitizing Tablets*. url: http:// www.opendtect.org/rel/doc/User/base/appendix_wacom.htm (visited on 04/02/2013).

[@Opeb] OpendTect. *Open Source Seismic Interpretation System*. url:http://www.opendtect.org (visited on 04/01/2013).

[@Pet] Petrel. *Petrel E&P Software Platform, Schlumberger*. url:http://www.slb.com/services/software/geo/petrel.aspx (visited on 07/28/2013).

[@Pho] Adobe Photoshop. *Image editor software | Adobe Photoshop CS6*. url:http://www.adobe.com/products/photoshop.html (visited on 04/03/2013).

[@Pro] Professional Darts Play Association. *The PDPA | Professional Darts Players Association*. url: http://www.pdpa.co.uk/2012/07/28/betfair-world-matchplay-semi-finals/darts-pdcdartsphil-taylor-terry-jenkins-q_finals-3/ (visited on 05/03/2013).

[@Rus] Jon Russell. *Last Week in Asia: GREE Splashes Out, Xiaomi's Big Profits and More*. url: http://thenextweb.com/asia/2012/05/06/last-week-in-asia-gree-goes-shopping-xiaomi-outs-huge-revenues-india-shows-mobile-potential-and-more/ (visited on 05/03/2013).

[@Sch] Schlumberger. *horizon - Schlumberger Oilfield Glossary*. url: http: //www.glossary.oilfield.slb.com/en/Terms/h/horizon.aspx (visited on 07/01/2013).

[@Sie] Siemens. *CAE / Computer-Aided Engineering: Siemens PLM Software*. url: http://www.plm.automation.siemens.com/en_us/plm/cae.shtml (visited on 05/01/2013).

[@Sil] Silicon Graphics, Inc. *SGI - Developer Central Open Source | Open Inventor*.url: http://oss.sgi.com/projects/inventor/(vis-ited on 05/20/2013).

[@The] The Fitness Hack. *Muscle size doesn't equal strength | THE FITNESS HACK*.url: http://thefitnesshack.wordpress.com/2011/06/08/muscle-size-doesn't-equal-strength/ (visited on 05/03/2013).

[@USE] U.S.Energy Information Administration. *Cougar Land Services-Landoner Information*. url: http://cougarlandservices.net/landowners (visited on 07/01/2013).

[@VRG] VRGeo. *VRGeo: Home*. url: http://www.vrgeo.org/ (visited on 04/03/2013).

[@Waca] Wacom. *FAQ - Wacom Feel Multi-Touch API*. url: http://www.wacomeng.com/touch/WacomFeelMulti-TouchFAQ.htm (visited on 03/24/2013).

[@Wacb] Wacom. *The ergonomic way to use your computer*. url: http://www.wacom.eu/index2.asp?pid=186 (visited on 04/01/2013).

[@Wacc] Wacom Technology Corp. *Wintab Interface Specification 1.4: 16-bit and 32-bit API Reference*. url: http://www.wacomeng.com/windows/docs/Wintab_v140.htm (visited on 07/15/2013).

[@Wacd] Wacom Technology Corp. *Cintiq 24HD touch.* url: http://www.wacom.com/en/creative/products/pen-displays/cintiq/cintiq-24hd-touch (visited on 07/15/2013).

[@Weia] Eric W. Weisstein. „*Multimodal.*“ *From MathWorld–A Wolfram Web Resource.* url: http://mathworld.wolfram.com/Multimodal.html (visited on 07/09/2013).

[@Weib] Eric W. Weisstein. „*Unimodal.*“ *From MathWorld–A Wolfram Web Resource.* url: http://mathworld.wolfram.com/Unimodal.html (visited on 07/09/2013).

A // Questionnaire Analysis

A QUESTIONNAIRE ANALYSIS

User Nr. ______________________ Age: ____________ Date: ______________

Questionnaire

Average rating

I am using a PC	Never	Sometimes	Daily	Average rating
At my workplace	☐---☐---☐---☐---🟥			5,0
In my spare time	☐---☐---☐---☐--🟥--☐			4,4

I am using the following tools	Never	Sometimes	Daily	Know none of them	Average rating
Graphic design tools (e.g. Photoshop, InDesign, Corel…)	☐---☐---☐-🟥--☐---☐			☐	3,33
Presentation software (e.g. PowerPoint, Keynote…)	☐---☐---☐--🟥--☐---☐			☐	3,53
3D Applications (e.g. Maya, Blender, AutoCAD…)	☐---☐---🟥☐---☐---☐			☐	2,86

1. Judge how well and effective the specified tasks could be solved

   a. With the pentouch interaction techniques?

☐-🟥---☐---☐---☐---☐		1,26
Very good	Not at all	

   b. With the desktop interaction techniques?

☐---☐--🟥--☐---☐---☐		2,6
Very good	Not at all	

2. Which of the two interaction paradigms would you prefer?

☐--🟥--☐---☐---☐---☐		1,46
PenTouch paradigm	Desktop paradigm	

3. Judge how well and effective the PenTouch interaction supported you in the horizon tracing workflow?

☐--🟥--☐---☐---☐---☐		1,46
Very good	Not at all	

4. According to your opinion, how intuitive and natural was the multitouch based navigation?

☐-----🟥----☐----------☐----------☐----------☐	
Very natural / intuitive	Not natural / intuitive all

1,53

5. How closely does the differentiation between touch input and pen input resemble real-life interaction?

☐----------🟥----------☐----------☐----------☐	
Very natural	Not natural all

2,0

6. Were the interaction techniques easy to learn?

☐--------🟥--☐----------☐----------☐----------☐	
Very easy	Very complicated

1,8

7. Was the system fun to use?

☐--🟥--------☐----------☐----------☐----------☐	
I agree	I do not agree

1,2

8. I can imagine using the system on a regular basis?

☐--------🟥--☐----------☐----------☐----------☐	
I agree	I do not agree

1,8

9. Here is a plenty of space for your annotations and suggestions. Do you have any ideas or recommendations? What was good? What was bad?
