Hochschule Düsseldorf
University of Applied Sciences

Fachbereich Medien
Faculty of Media

Interaction Optimization through Pen and Touch, Eye Tracking and Speech Control for the Multimodal Seismic Interpretation Workspace

Mehmet Danyel Kemali

657534

Master of Science Thesis Media Informatics

December 2016

Supervision:

Prof. Jens Herder, Dr. Eng. / Univ. of Tsukuba

External Supervision:

Dr. Stefan Rilling

Fraunhofer IAIS

Anmelde-Formular

➔ Hinweis vor der Abgabe: Binden Sie *an Stelle dieser* Seite ein Exemplar des Anmeldeformulars ein, auf dem bei der Anmeldung im Studierendenbüro das Datum der Abgabe eingetragen wurde.

➔ Geben Sie Ihre drei Exemplare der Ausarbeitung sowie das zusätzliche Anmeldeformular im Studierendenbüro ab und lassen Sie das Datum Ihrer Abgabe darauf jeweils eintragen.

➔ Dann geben Sie jeweils ein Exemplar an den Betreuer/die Betreuerin – persönlich oder über den Postkasten.

➔ Ein Exemplar bleibt bei Ihnen.

STATEMENT OF ORIGINALITY

I hereby certify that I am the sole author of this Master Thesis „Interaction Optimization through Pen & Touch, Eye Tracking and Speech Control for the Multimodal Seismic Interpretation Workspace“ 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.

_____

Date, Signature

Contact

Mehmet Danyel Kemali
Richardstr. 120
40231 Düsseldorf, Germany
danyelkemali@gmx.de

Abstract

This thesis examines the development of a prototype desktop application for the interpretation of visualized seismic data utilizing pen and touch, eye tracking and speech recognition interaction technologies. Particularly, the combination of the interaction technologies to create a multimodal user interface for seismic interpretation is researched by this thesis, considering current developments in the field of research of multimodal interfaces. Issues during the development are identified and possible solutions are described by this thesis.

Based on previous work by Fiedler et al., which investigated a multimodal interface using a Kinect motion sensor and speech control for a room scale application enabling interaction with seismic visualizations, this thesis aims to survey the transportability of the results into a single user workspace (Fiedler et al. 2015). Fiedler et al. observed a significantly improved efficiency for the execution of complex tasks. In a similar approach, the prototype implemented for this research is evaluated by ten participants during a user study, hinting that the results might be dependent on the task and therefore, not transmissible per se. The necessity of further development, more user studies, perhaps over a longer period, and altered goals for further research are detected and presented by this thesis.

Kurzfassung

Diese Masterarbeit beschäftigt sich mit der Entwicklung eines Prototyps für die Interpretation von visualisierten, seismischen Daten. Die Anwendung nutzt Stift- und Berührungseingaben, eine Technologie zur Augenverfolgung, sowie eine Schnittstelle zur Spracherkennung, um eine möglichst natürliche Interaktion zu erzeugen. Insbesondere die Kombination der Interaktionstechnologien, um eine multimodale Interaktionsschnittstelle zu entwickeln, wird in dieser Arbeit untersucht. Probleme, die während der Entwicklung identifiziert wurden und mögliche Lösungen, werden im Rahmen dieser Arbeit präsentiert.

Aufbauend auf einer vorangegangenen Arbeit von Fiedler et al., die eine multimodale Schnittstelle, unter der Nutzung eines Kinect Sensors in Kombination mit Sprachsteuerung, zur Interaktion mit seismischen Daten untersucht haben, zielt diese Arbeit darauf, die Übertragbarkeit der Ergebnisse von Fiedler et al. in eine Arbeitsumgebung für eine einzelne Person festzustellen (Fiedler et al. 2015). Fiedler et al. konnten eine während der multimodalen Nutzung des Systems verbesserte Effizienz für die Ausführung einer komplexen Aufgabe feststellen. In einem ähnlichen Versuch wurde im Rahmen dieser Arbeit eine vergleichbare Nutzerstudie mit zehn Personen durchgeführt. Die Ergebnisse zeigen auf, dass das Resultat von Fiedler et al. sich nicht per se übertragen lässt. Die Notwendigkeit sowohl von weiteren Forschungen, als auch die Möglichkeit weiterer Untersuchungen, z.B. über längere Zeiträume hinweg, sowie veränderte Ziele für die zukünftige Forschung, werden von dieser Arbeit aufgezeigt.

Table of Contents

Table of Figures

Table of Codes

Table of Tables

I. Introduction

1 Introduction

After an introduction to the terminology used in this thesis, this chapter presents the motivation for this work by listing its research goals, concluding with a short introduction to seismic interpretation.

1.1 Terminology

Abbr.	Description
API	Application Programming Interface. “A set of functions and procedures that allow the creation of applications which access the features or data of an operating system, application, or other service.” Taken from the Oxford online dictionary (Oxford 2016).
IT	Interaction Technique. A Technique is “A way of carrying out a particular task, especially the execution or performance of an artistic work or a scientific procedure”. Taken from the Oxford online dictionary (Oxford 2016).
IDE	Integrated Development Environment. A software application for software development.
IAIS	Fraunhofer Institute for Intelligent Analysis and Information Systems.
JSON	JavaScript Object Notation. Open-standard data-interchange format.
GUI	Graphical User Interface.
MUI	Multimodal User Interface.
MMSIW	Multimodal Seismic Interpretation Workspace.
SDK	Software Development Kit. Set of software development tools.
XML	Extensible Markup Language. “A metalanguage which allows users to define their own customized markup languages, especially in order to display documents on the Internet.” Taken from the Oxford online dictionary (Oxford 2016).

1.2 Motivation

For years, the mouse has been the main input device for desktop computers. With developments on the mobile sector, especially the rise of smartphones, new interaction technologies emerged within a decade and became available for the broader public. In our everyday life, touch input is being used mainly on mobile devices while the combination of mouse and keyboard is still preferred for desktop computers. Jakob Nielsen from the Nielsen Norman Group rates the different input techniques, comparing them to each other. A mouse delivers a better performance in terms of precision, but per Nielsen it fails when it comes to "Direct engagement with screen and 'fun' to use in comparison to touch" (Jakob Nielsen 2012). The author concludes his article by urging the reader to emphasize the strength of the available input, e.g. using tooltips when a mouse is available.

To identify the strength of natural interaction technologies, to combine newly developed interaction technologies and to evaluate the research with regards to seismic interpretation, the research in Human-Computer Interaction has been and still is a keen interest of the VR Geo Consortium. The VRGeo Consortium supports research on different fields to find new innovations for the oil and gas exploration (VRGeo 2016). The motivation for this project is to research how new interaction technologies could improve seismic interpretation. A previous work, called the Multimodal Seismic Interpretation Workspace, has been part of the Human-Computer Interaction research since late 2015. The creation of a new prototype builds on previous work to optimize the workspace during further research. Main aspects for the optimization are the inclusion of additional interaction technologies respectively, the usage of already implemented technologies in a more multimodal manner, e.g. the combination of speech and eye tracking.

1.3 Goals

Implementing multimodal interfaces into interactive systems aim to increase the available capabilities for interaction. Matthew Turk from the Department of Computer Science of the University of California reviews multimodal interaction in an article, stating that "Multi-

modal interfaces describe interactive systems that seek to leverage natural human capabilities to communicate via speech, gesture, touch, facial expression, and other modalities, bringing more sophisticated pattern recognition and classification methods to human–computer interaction." (Turk 2014).

This thesis' aims assume that multimodal interaction can optimize the interaction with a seismic interpretation workspace. Based on previous work from Fiedler et al., it is assumed that the usage of multimodal interaction could improve the efficiency in seismic interpretation when complex tasks are carried out (Fiedler et al. 2015). Fiedler et al. determined that the use of the developed synergistic, simultaneous fusion engine method eases the cognitive load for complex tasks. Therefore, one goal of this thesis is to find out if the results of Fiedler et al. can be transferred to a different setup for seismic interpretation like the presented workspace. Furthermore, Fiedler et al. could not determine a significance in favor of multimodal interaction compared to unimodal interaction in terms of user acceptance and "fun" during the usage of the system. To confirm, that this result also applies for the specific workspace used in this setup, the initiated user study also includes a similar questionnaire like the one Fiedler et al. used. Nevertheless, the conducted user study is carried out differently, letting all participants execute each task multi- and unimodal respectively with ITs using new technologies like gaze, speech, pen and touch and ITs using common technology, e.g. mouse and keyboard. The idea for the altered approach assumes, that the user study participants might have a different perspective on the system, e.g. in terms of how enjoyable working with it is, when they had the chance to try it out with both new and commonly know technologies.

Derived from the previous work, this leads to two hypothesis statements as the goals of this thesis.

(I) Hypothesis H_1

Using multimodal interaction in complex tasks for a seismic interpretation workspace increases the user's efficiency.

(II) Hypothesis H_2

The usage of the multimodal fusion engine to complete tasks leads to an increased acceptance and is more enjoyable to use than using a common interaction technology, e.g. a mouse.

To confirm or to reject the statements, a user study is conducted to evaluate a new prototype implementation of the Multimodal Seismic Interpretation Workspace.

1.4 Outline

This work is split into three big sections. The first chapters revolve around some mandatory introduction to the topic, related work and the design process of the workspace. The second part describes the implementation process and responds to issues which occurred during the development and how these issues have been solved. In the last part, the user study is examined in detail. The results are discussed and an outlook on the future direction concludes this thesis.

1.5 Seismic Data Interpretation

The process of interpreting seismic data to detect petroleum reservoirs may be also described as the interpretation of seismic reflections retrieved from hydrocarbon exploration. In the Schlumberger Oilfield Glossary, reflection is defined in terms of geophysics as "[…] the return or rebound of particles or energy from the interface between two media" (Schlumberger 1998d). In principle, waves of energy are sent into the subsurface of Earth. The waves are reflected partially by the subsurface geological layers. The reflections are recorded over a period by geophones on land or hydrophones under water and the retrieved data is processed into electrical impulses and results in seismic data (Schlumberger 1998c). In Figure 1, a schematic of marine seismic acquisition shows the necessity of two vessels for the acquisition of data. The source vessel is the boat on the left, which sends the energy waves while the recording boat uses groups of hydrophone to retrieve the reflected data.

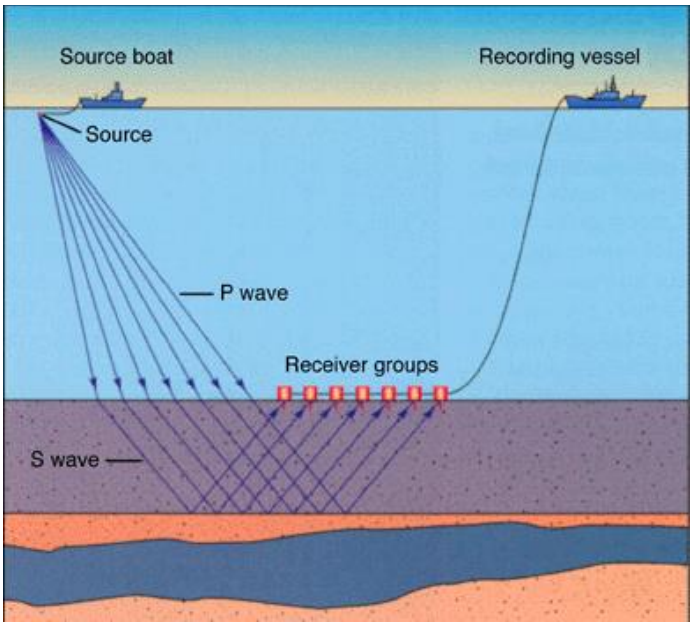


Figure 1: Schematic of marine seismic acquisition, taken from the Schlumberger Oilfield Glossary (Schlumberger 1998a)

Seismic data can be visualized like it is done by the in-house development of Fraunhofer IAIS. An example for visualized seismic data can be seen in Figure 2, showing two slices with different attributes. A slice represents the seismic volume data in a direction and position.

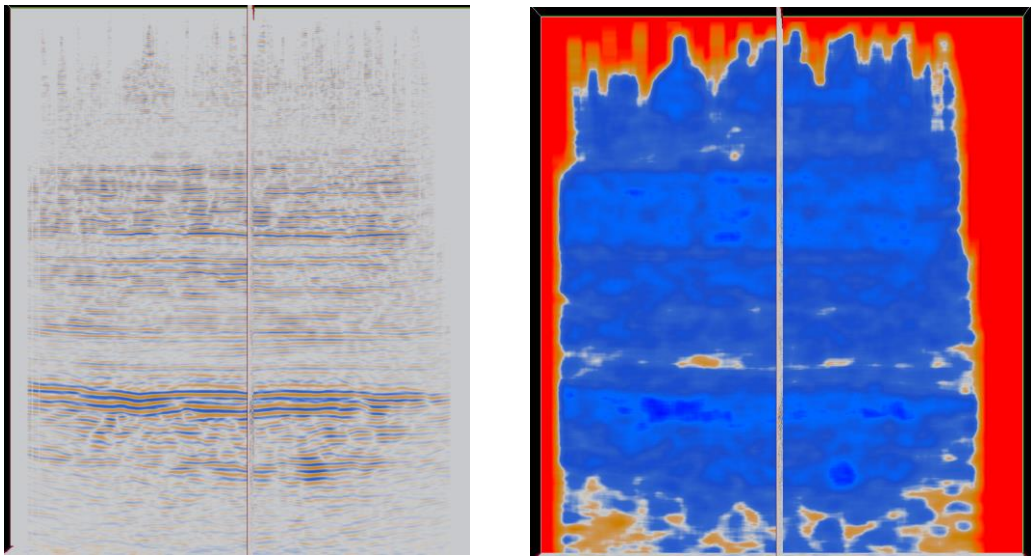


Figure 2: Visualization of seismic data by the in-house development of Fraunhofer IAIS, on the left attribute "Amplitude", on the right attribute "Entropy".

Seismic volume data offers visualized seismic data in a three-dimensional manner using the seismic lines in which the data was acquired. These seismic lines are called crossline, in-line

and timeline and are the results of a 3D survey, which is run during acquisition. The lines are "[…] perpendicular to the direction in which the data were acquired" (Schlumberger 1998e). As seen in Figure 3, the retrieval of Seismic volume data Slices is visualized.

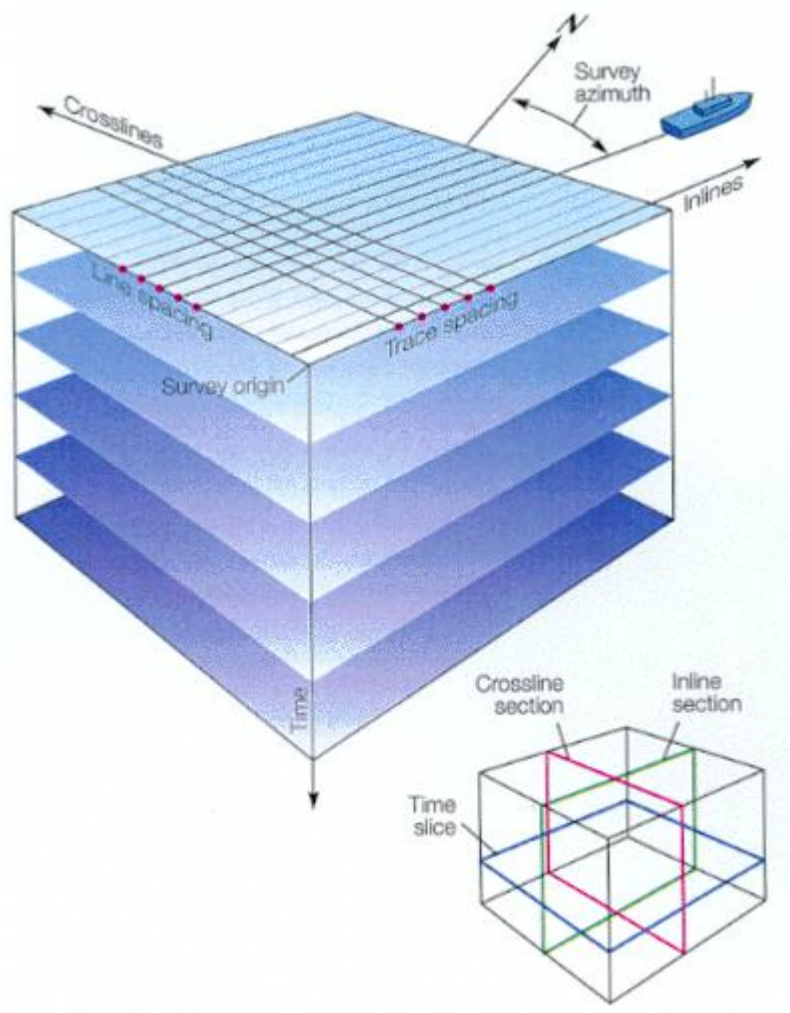


Figure 3: Diagram of seismic lines, taken from the Schlumberger Oilfield Glossary (Schlumberger 1998e)

The measured properties of seismic data are called attributes (Schlumberger 1998b). The attributes possess geological significances which are used during the seismic data interpretation process to gain a different perspective on the data.

For the Multimodal Seismic Interpretation Workspace, the interpretation process is realized by enabling the user to paint with attributes directly on a slice of seismic volume data. Painting attributes directly onto a slice reveals a different view of the seismic data, enabling the interpreter to draw conclusions, whether further actions should be initiated in the interpreted area.

I. Related Work

2 Related Work

The Related Work section presents briefly the important developments on the human-computer interaction field of research involving multimodal interaction. Furthermore, this chapter introduces the previous work which is the foundation of the application developed for this thesis.

2.1 History

Several research projects in the past investigated the combination of different interaction technologies. One of the first projects was Richard A. Bolt's contribution to the ACM SIGGRAPH Conference on Computer Graphics in 1980. The issue presented the combination of speech and gesture inputs (Bolt 1980). The interface was named "Put-That-There", enabling the user to point at different areas of a projection in a media room, which is the gestural part of the system, and triggering actions by saying commands. For example, the user can create basic shapes like squares and circles. Figure 4 illustrates the usage of the system.

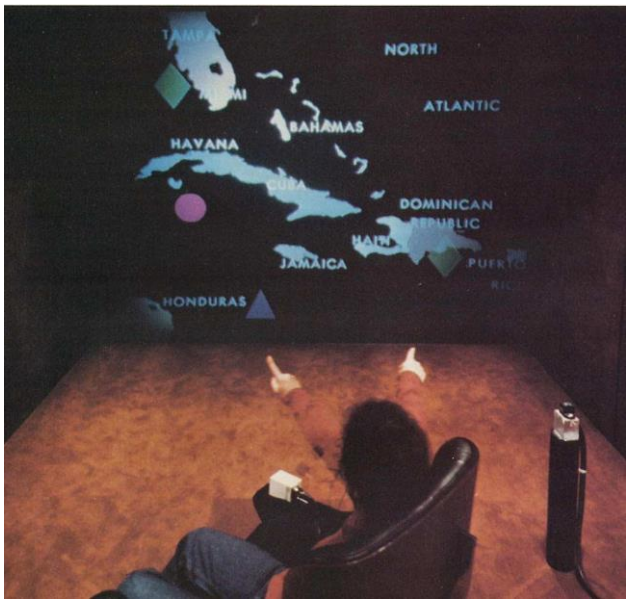


Figure 4: Media Room used for the "Put-That-There" demonstration (Bolt 1980).

The system utilizes a Connected Speech Recognition System (CPRS) developed by NEC (Nippon Electric Company) and was therefore able to recognize connected speech. For the

gestural part of the system, Bolt used a position and orientation sensing technology based on measurements with nutating magnetic fields. This technology requires the usage of a small sensor as a pointing device, which can also be wrist-mounted (Bolt 1980, pp. 264–265). With commands like “Move that to the right of the green square”, the two independent input streams, speech and gesture, are combined to determine the user’s intention. Per Bolt, using the pronoun leads to a mode where the user does not need to know what he is actually pointing at and is still capable of achieving his aim. This process of “pronomialization”, as Bolt it named, leads to the name giving speech command “Put-that-there”.

Bolts system introduced true multimodality in Human-Computer-Interaction, and many research projects followed his lead during the past 36 years. For instance, in 1989, Neal et al. introduced the CUBICRON project, aiming on the development of knowledge-based interfaces including speech and gesture. The stress point of the publication from 1989 was the part on NL (Natural Language). The presented system offered speech in- and output creating a dialogue system. The pointing gestures, realized with a common mouse input device, in combination with the spoken commands is used to dynamically create a multimodal language. Air Force mission planning was the application domain of the project. An example for NL which the researchers present is the possible command “What is the status of this <point> airbase?”. Based on the data of the pointing device, the information gap “<point>” is filled during the search process in the knowledge base.

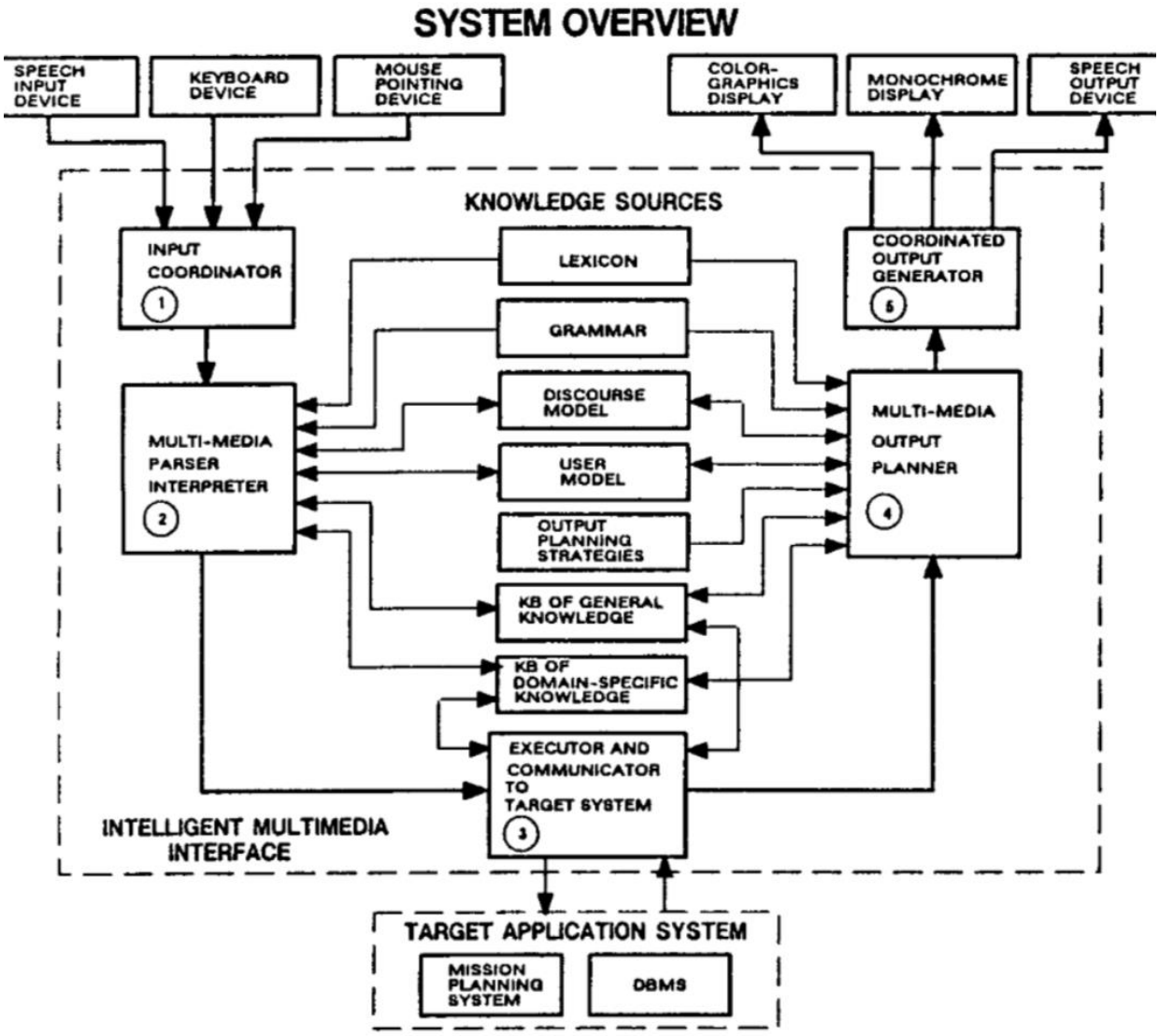


Figure 5: System Overview CUBICRON Project, (Neal et al. 1989, p. 413)

In Figure 5, Neal et al. illustrate the architecture of the system. The system already offers multimodal in- and output, using two units, the "Multi-Media Parser Interpreter" and "Multi-Media Output Planner" in conjunction with the knowledge sources to compose the reply for the user. The process of combining input- and output streams has often received names like Fusion Engine. The Fusion Engine of a multimodal system are "[t]he mechanisms used for combining information […]" (Lalanne et al. 2009) and has also received different names in the past, e.g. combining or integration (Lalanne et al. 2009, p. 164). The Fusion Engine of a multimodal system can therefore be described as a key component. The CUBICRON project by Neal et al. included the interaction with geographic maps in a multimodal manner (Neal

et al. 1989). The approach of the project, creating multimodal interaction, to interact with large map data, is comparable to the Multimodal Seismic Interpretation workspace, although, the application domains differ greatly.

Neal et al. already incorporated NL in their CUBICRON project in combination with a pointing device. Therefore, the CUBICRON projects seems to have a similar initial position in comparison to the Multimodal Seismic Interpretation Workspace, but the MMSIW combines speech, gaze, pen and touch interaction in terms of seismic interpretation. Therefore, a closer look to the multimodal developments of the past decades including the fusion of speech and gaze is required.

In 1993, Koons et al. approached the integration of simultaneous input from speech, gaze and hand gestures (Koons et al. 1993). In their publication, Koons et al. present a prototype capable of handling input from speech, gaze and hand gestures and processing the data into a "[…] single integrated meaning" (Koons et al. 1993, p. 261). Koons et al. introduce gaze as an additional pointing input. The user is not forced to use the eye tracker to select objects on the map. Koons et al. justify this with their non-intrusive approach to include eyes into the interaction process. An interesting part of the project is the usage of timestamps to connect the input streams to the frame-based system. This allows the researchers to create a parsing tree to interpret commands which might be declared as "natural". Koons et al. exemplify that process with the speech command "That blue square below the triangle". After the speech command has been parsed, an evaluation method for each frame, created by the other input streams, is being called. Koons et al. illustrates the process in Figure 6, e.g. the evaluation method attached to OBJ2 tries to find an object in the propositional system, which features a red color and the shape of a triangle. When an object with the mentioned features is detected, it becomes available for the evaluation method of OBJ1, all together with information on its spatial representation, which is the frame evaluating the speech command "below". If multiple items with the desired features are determined, the system runs problem-solving methods taking gesture and eye modes into account with their temporal relation to the frame (Koons et al. 1993, pp. 266–267).

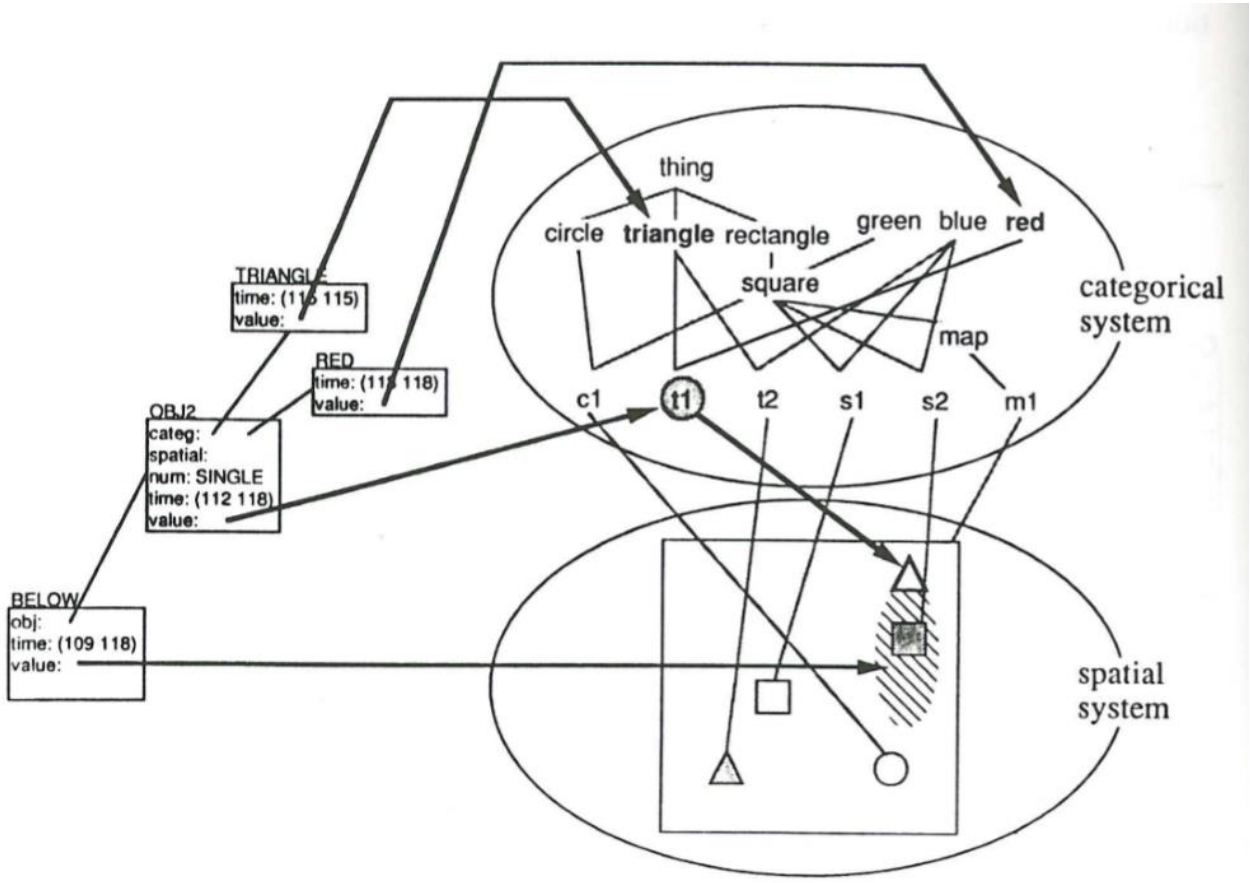


Figure 6: Evaluation of "below the red triangle" (Koons et al. 1993, p. 266).

The process presented by Koons et al. shows an interesting attempt on how multimodal input streams could be used to resolve deictic speech commands. The current prototype of the Multimodal Seismic Interpretation Workspace relates on more simple strategies to resolve speech commands, excluding deictic references, and solely relying on the gaze position, though deictic commands may be used by the user optionally during some commands. With further evaluations of the Multimodal Seismic Interpretation Workspace prototype, a more complex routine to resolve speech commands and the inclusion of more deictic interaction needs to be considered.

A more recent work presented in 2013 by Schulz et al. combines speech, eye tracking, gesture, pen interaction and a see-through interface as the working environment of a radiologist (Schulz et al. 2013). The aim of the research project seems to be comparable to the Multimodal Seismic Interpretation Workspace, researching on how the working environment in a specific field could be improved with the usage of multiple-devices as input. The see-through

interface in combination with speech enables the user to interact with the working environment hands free. Furthermore, additional information is directly overlaid in to the view of the user, see Figure 7.

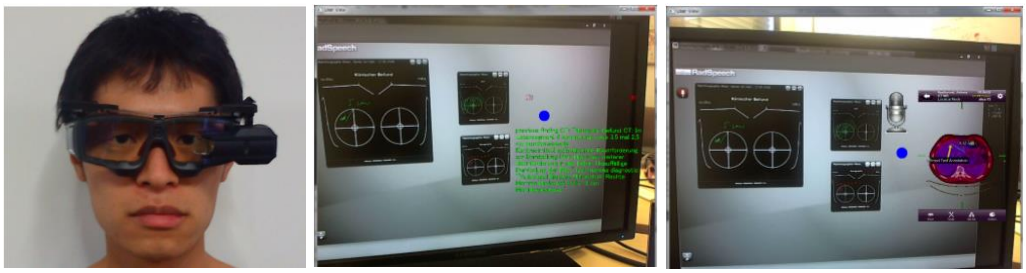


Figure 7 : See-through Interface combining Eye Tracking with an HMD, blending additional information directly into the view of the user (Schulz et al. 2013)

Comparable to the idea of Schulz et al., one approach for the Multimodal Seismic Interpretation Workspace is to investigate how interaction with the Overview Area can be realized, without the requirement to move pen and touch interaction into this area.

There have been also several projects evaluating multimodal interaction by conducting user studies. Oviatt et al. researched the differences in multimodal integration patterns over a longer period of several weeks. Per Oviatt et al. the differences in multimodal integration patterns are consistent and stable over time, therefore, sequential and simultaneous integrators keep their chosen type of multimodal interaction for most of the time (Oviatt et al. 2005). Hence, the designing of multimodal interfaces should also consider the differences in the integration of multimodal integration patterns as well.

Ömer Genç conducted research on novel pen & touch interaction techniques for seismic interpretation in 2013, which resulted in the integration of multimodal interaction into an existing framework. Genç's research, focusing on the integration of seamless ITs using pen & touch is not the only research conducted by Fraunhofer IAIS in multimodal human-computer interaction (Ömer Genç 2013). A comparison between unimodal and multimodal interaction is presented in 2015 by Fiedler et al. The user study evaluates the interaction with an interactive visualization application, displaying seismic data (Fiedler et al. 2015). The application includes the usage of speech, gesture, and features 3D Stereo for the visualization of the seismic volume data, see Figure 8.

Figure 8: Multimodal interaction with speech and gesture input technology (Fiedler et al. 2015).

For the user study, Fiedler et al. divided the participants into two equally large groups, performing similar tasks. While the participants of one group carried out all tasks unimodal, the other group's participants executed all tasks multimodal. The user study has been conducted under observation, and the time each user needed to perform a task, the errors he made and the amount of help he needed, were recorded by an overseer. The results of the user study suggest increased efficiency when complex task where executed multimodally. A similar significance could not be detected for less complex tasks (Fiedler et al. 2015).

2.2 Preliminary Work

After the presentation of the interactive visualization application during a VRGeo consortium meeting, which is part of the work of Fiedler et al. from 2015, the members of the consortium started a project to investigate multimodal interaction for seismic interpretation workspaces. The first prototype implements natural interaction technology with pen and touch support and the visualization of the seismic volume data is achieved with an in-house development by Fraunhofer IAIS. Only three slices are available for interaction, each for one direction. Main features of the workspace consist of the interpretation of seismic data

by painting with a variety of attributes, which are selectable from a GUI, on the visualized slices and the possibility to change the position of a slice. The first prototype already offers a multi-display setup, composed of a Wacom Cintiq 27 QHD Pen & Touch Device (Wacom 2015) and a common monitor, stacked on top of the Wacom Cintiq. Figure 9 illustrates the setup.

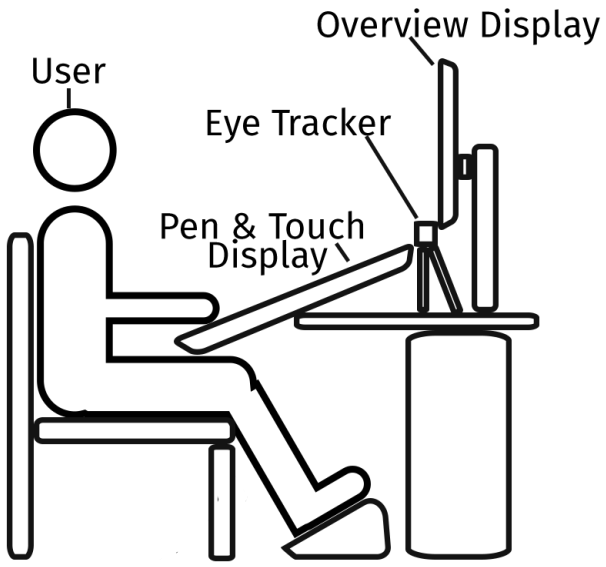


Figure 9: A schematic illustrating the setup of the Multimodal Seismic Interpretation Workspace.

On the Overview Display, the user can view the seismic data visualization with additional direct volume rendering, which can be seen in Figure 10. Direct volume rendering uses ray casting to create a 3D volume based on the seismic data. Rays are cast travelling into the scene. A test for intersections with the scene data determines which color a pixel needs to display, resulting in the desired image. The incorporated visualizer also supports stereoscopic 3D, but requires dedicated graphics hardware like Nvidia Quadro graphic cards, supporting quad-buffered stereo with OpenGL (NVIDIA 2016) .

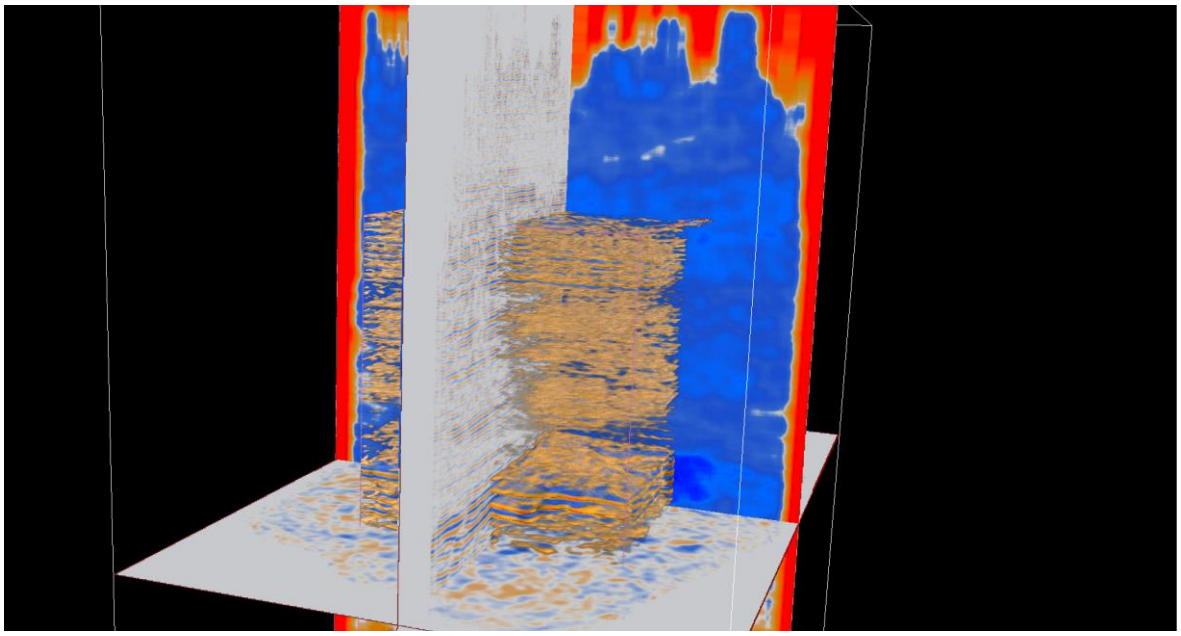


Figure 10: Direct volume rendering output by the Fraunhofer IAIS development.

For the prototype, an eye tracker is implemented to reduce the number of samples used for the direct volume rendering in non-gazed areas. The feature is called *Focused Volume Rendering* and is motivated by the fact that direct volume rendering requires high performance hardware due to the demanding ray casting calculations. The samples property of a ray casting technology determines the number of rays casted in order to test for intersections with the scene data. Besides other possible solutions for performance improvements, e.g. early ray cancellation, the reduction of the sample value improves the overall system performance significantly. Utilizing the eye tracker, the visualizer reduces the samples in areas which are not focused by the user, see Figure 11.

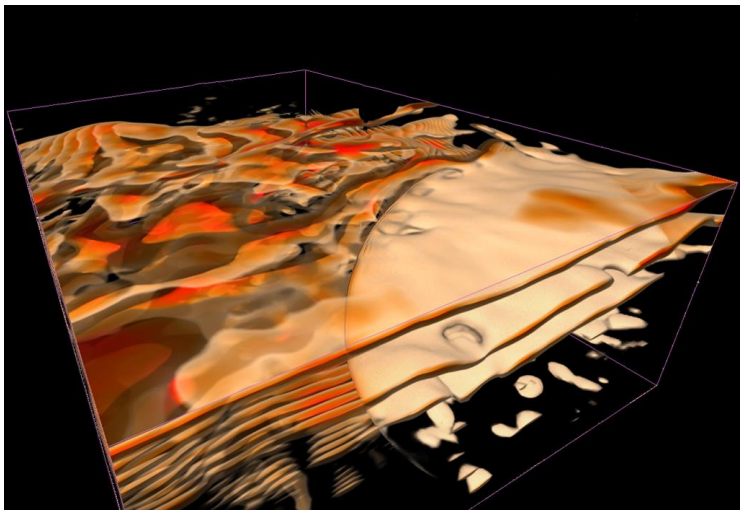


Figure 11: Focused Volume Rendering reduces the samples in areas, where the gaze of the user is not focusing.

The workspace application is written in C++ using the IDE Visual Studio on computers running the Windows operating system by Microsoft. For the visualization, the in-house development is primarily using the OpenSceneGraph library, which is open-source software (OpenSceneGraph). OpenSceneGraph is a 3D graphics API and widely used in scientific visualization. It is also written in C++ and uses the graphics API OpenGL (Khronos Group 2016). Therefore, OpenSceneGraph requires graphics hardware with OpenGL support.

II. Design

3 Design

The optimization of the Multimodal Seismic Interpretation Workspace is conceptualized based on discussions with VRGeo consortium members. This section introduces the initial draft of the system's concept and the design of new ITs.

3.1 Concept

In March 2016, a VRGeo consortium meeting was held on site at Fraunhofer IAIS. The consortium members tested the then current prototype iteration. The prototype equated to the system described in the Preliminary Work section of this thesis. See Figure 12 for a photograph of the setup, showing the two areas of the workspace.

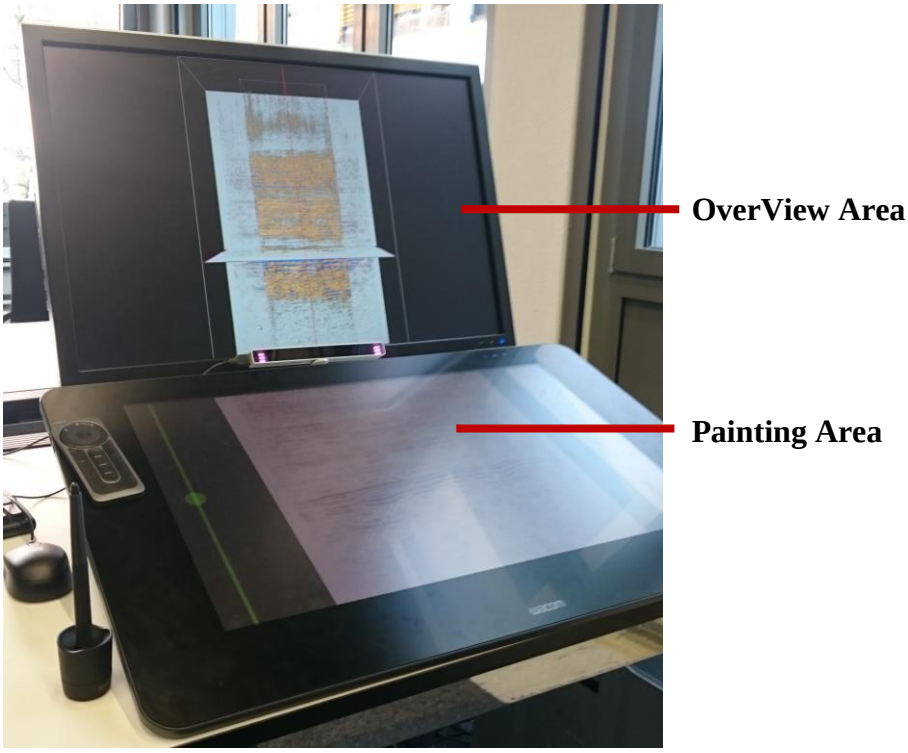


Figure 12: The first prototype of the Multimodal Seismic Interpretation Workspace.

During a so-called Feedback Session, the feedback of all members was collected. The multimodal aspect of the workspace was limited to a combination of pen, touch, and eye tracking interaction techniques. The chairman of this research project, Mark Dobin, suggested to expand the multimodal interaction of the workspace by adding speech control as a new interaction paradigm. To upgrade the Multimodal Seismic Interpretation Workspace closer to common seismic interpretation software, Dobin proposed a redesign of the software, including an increased feature set and a new mode for the second display. Dobin recommended to combine speech control commands with the gaze of the user tracked by an eye tracker to dynamically change the action a speech command triggers. The usage of speech control as a main interaction paradigm for the second display is justified by the inevitable break of the used interaction technology, which occurs when the user switches from the Painting Area to the OverView Area for interaction purposes. Mark Dobin's suggestions were supported by the other consortium members and therefore included into the research and development endeavors for the project.

Based on the feedback the draft in Figure 13 is created showing the concept for the new prototype. The most significant difference in comparison to the existing prototype is the introduction of a new operating mode for the OverView called *SplitView*. In this mode, it is planned to list saved slices on the left half of the display, while offering the possibility to preview a slice on right side. Besides, it is also created with the idea of featuring a "pushing" of the previewed or selected slice to the Painting Area.

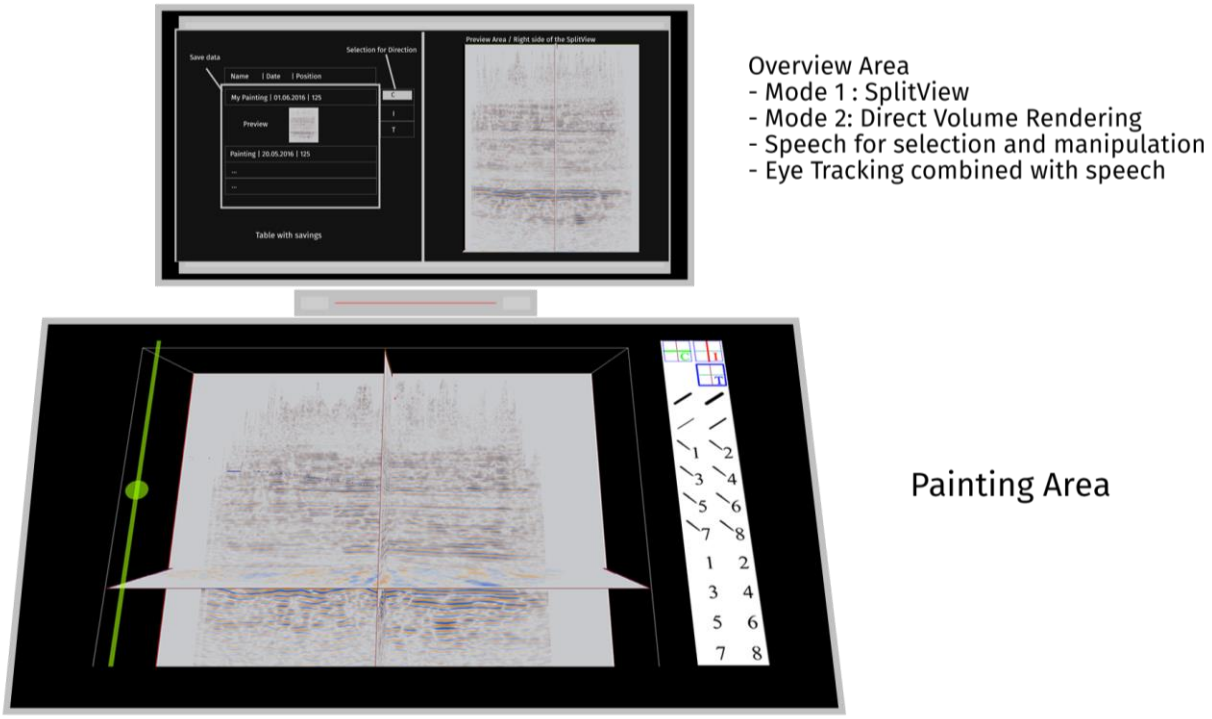


Figure 13: Concept for the new prototype of the MMSIW.

In conclusion, the concept for the next iteration of the prototype consists of an altered usage of the utilized eye tracker, includes speech and optimizes the overall interpretation process by adding new features to the workspace.

3.2 Design process & Multimodal Interaction

The design process for the workspace is organized iteratively to refine the tasks, as proposed by Jakob Nielsen (Nielsen 2008) and the iterations are coordinated with Dr. Stefan Rilling, who is the technical manager of the VRGeo consortium and therefore possesses increased experience in seismic interpretation.

Based on the feedback received during the consortium meeting, a new concept for the further development of the workspace is created, see 3.1. During the first step, a set of user stories is developed which are used to determine the features of the workspace. A user story is often constructed under consideration of the following scheme:

As a (role) I want (something) so that (benefit) (Ambler 2014)

For the Multimodal Seismic Interpretation Workspace, this leads to user stories like:

As a user I want to use a voice command to switch the current view direction (Crossline, Inline, Timeline), so that I don't need to use a different interaction technology.

A full list of all user stories can be found in the appendix section of this thesis. A so-called Kanban board is used to plan the development of the workspace loosely (leankit 2016).

The most influential tasks derived from the user stories include multimodal interaction. Dumas et al. visualized the process of multimodal input and output, providing an overview on multimodal interaction (Dumas et al. 2009). Per Dumas et al., multimodal interaction can be split into several states, see **Fehler! Verweisquelle konnte nicht gefunden werden.**. The main character in the process is of course the user, deciding on how to act next based on emotions, intentions etc. The actions are perceived by hardware devices, being the interfaces between the human and the computer. All perceived actions are interpreted in the fusion process, leading to the computation of an answer, which is then transferred back to the user with the same or other devices.

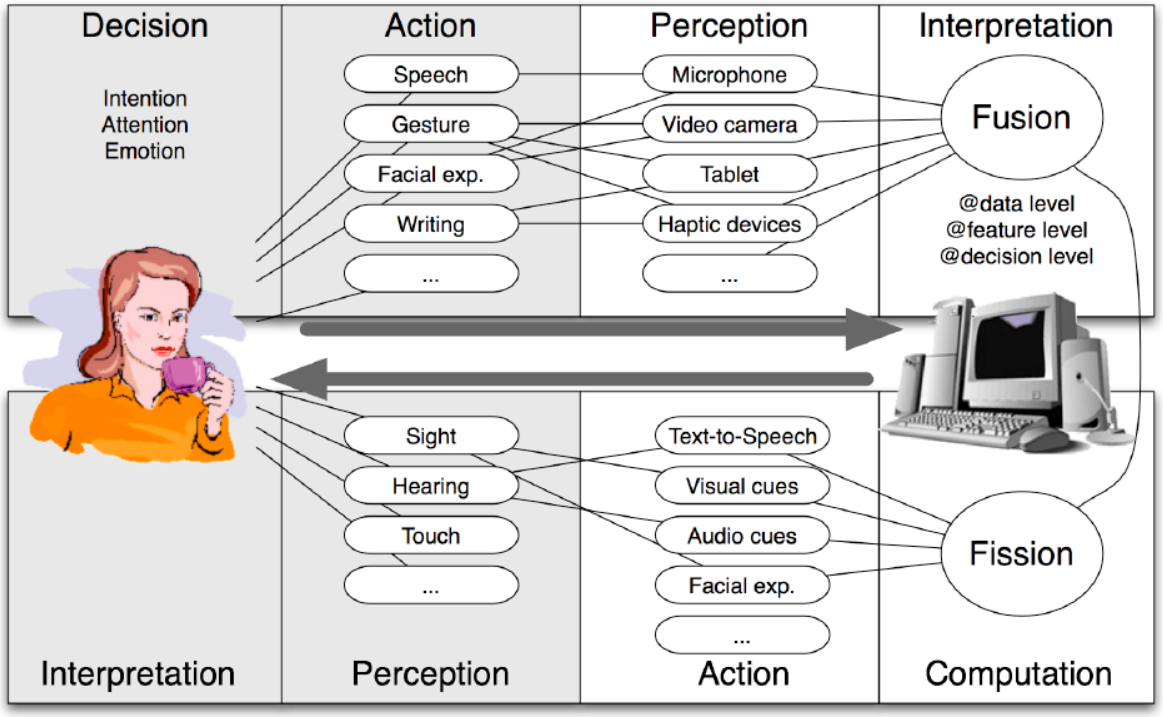


Figure 14: Multimodal interaction process, (Dumas et al. 2009, p. 8).

To transfer the ideas of the multimodal tasks into concrete interaction techniques, these should be contemplated as well. In their book "3D User Interfaces. Theory and Practice" Bowman et al. split 3D interaction techniques into the categories of selection/manipulation (Bowman 2005, 139 pp.) and navigation (Bowman 2005, 183 pp.).

Some interaction paradigms included in the MMSIW already follow commonly known paradigms. Bowman et al. also suggest to stick to existing manipulation techniques, proposing only to change to other techniques when the outcome is assumed to be beneficial (Bowman 2005, p. 179). In the Multimodal Seismic Interpretation Workspace, e.g. multi-touch is used to enable the user to pinch-to-zoom, which has become a very common IT across all touch-based devices. For the most part, the interaction techniques included in the Multimodal Seismic Interpretation Workspace are selected considering the design guidelines by Bowman et al (Bowman 2005, p. 179). Exceptions are made when it comes to new ITs which are utilizing multimodal interaction.

Figure 15 visualizes the idea to transfer the touch input from the Painting Area to the Over-View Area, based on the gaze of the user. Thus, removing the necessity to switch the input device to manipulate or navigate in the Overview Area.

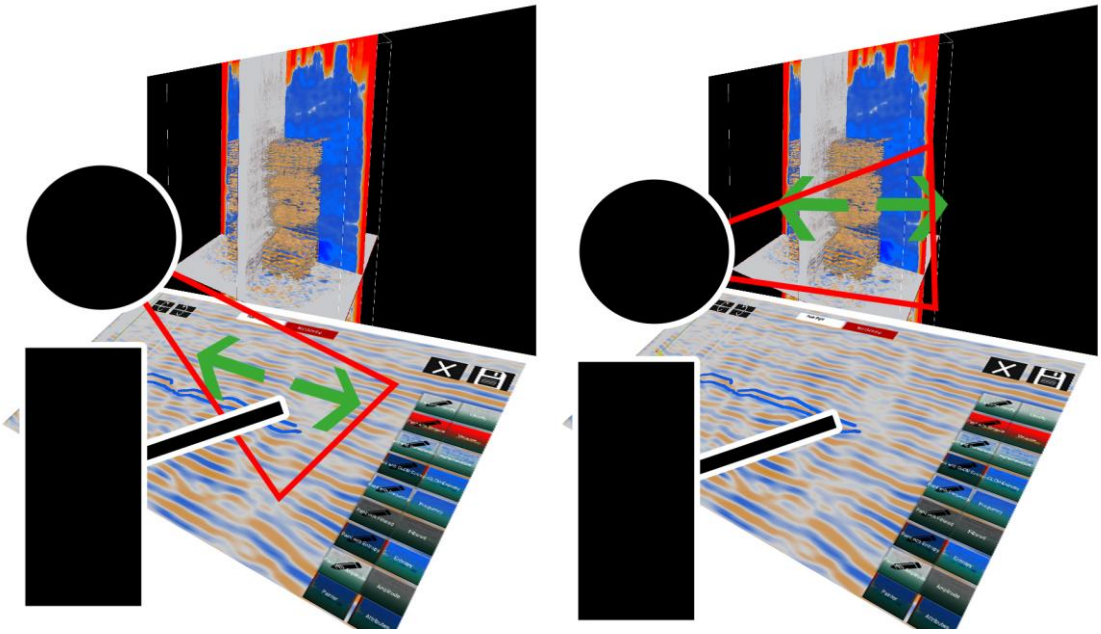


Figure 15: Usage of the Eye Tracker to determine when the touch input should be transferred to the OverView Area.

To describe the planned multimodal interaction in a more formal principle, Finite-state machines (FSM) are created to design the required interaction features of the workspace for the most complex tasks. Using the free UML (Unified Modelling Language) tool UMLet (UMLet 2016), an example for an FSM is created and can be seen in Figure 16, which visualizes the same process as seen in Figure 15.

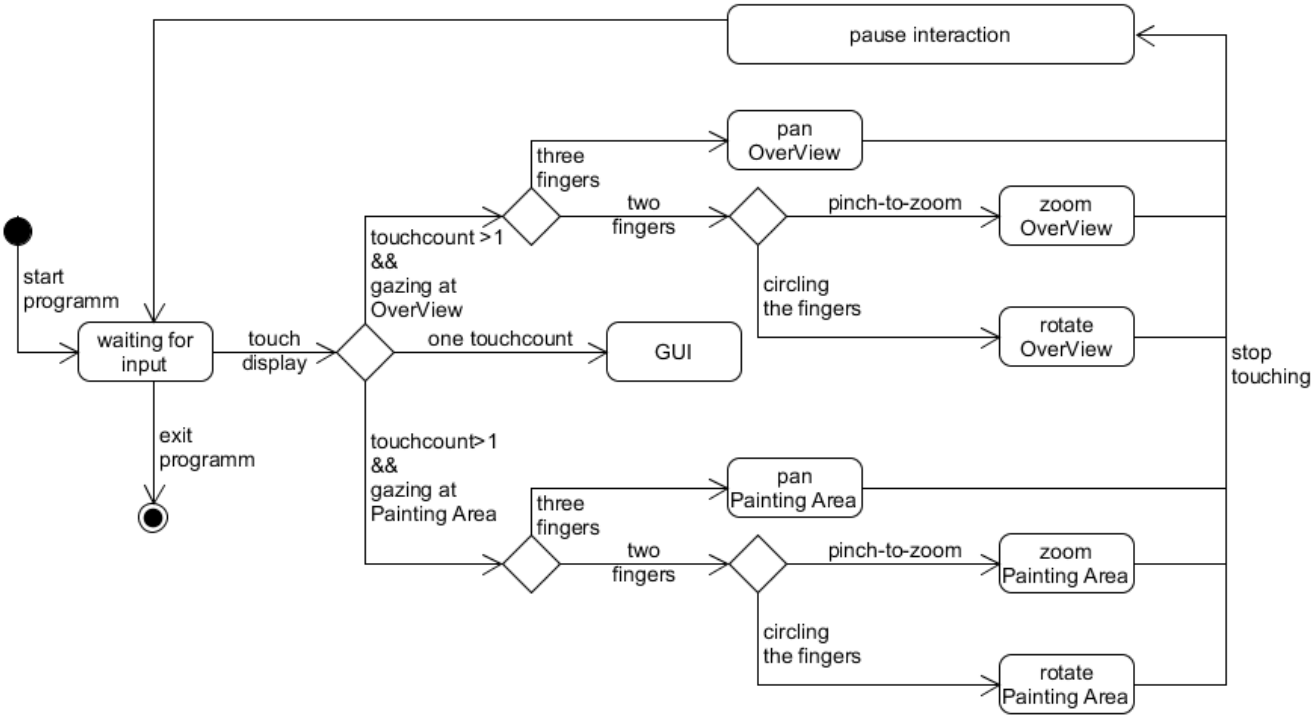


Figure 16: FSM visualizing the proposed touch interaction in combination with eye tracking.

Naturally, it is possible to create an FSM visualizing a process with all its facets, but this would lead to very complex figures. Therefore, parts of the process which could be split into more subtle representations are centralized.

One requirement, resulting from the user stories is the design and implementation of a completely new mode for the OverView. The requested mode needs to be designed to display a table on the left side of the OverView and a slice preview on the right side. Therefore, this mode is named *SplitView*. A table with saved slices is demanded for the left side. The user should be able to select a slice from the table and view it on the right side or in the Painting Area. The feature to “push” slices vocally is one of the main features of the new prototype. Figure 17 illustrates the procedure in a simplified visualization with an FSM.

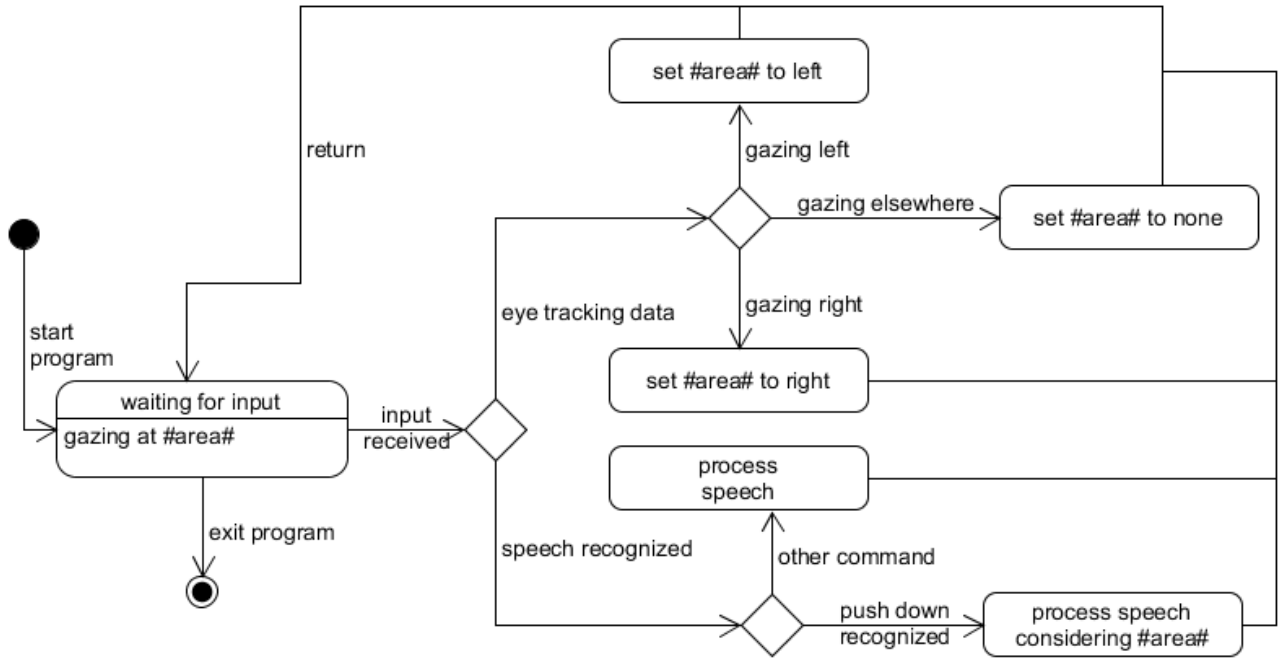


Figure 17: FSM visualizing the multimodal interaction for "pushing" a Slice via speech involving the gaze of the user.

A first draft of the SplitView Area is created based on the requirements, which can be seen in Figure 18. The draft provides a list of saved slices with various information. The proposed design filters the slices based on their seismic line property and also suggests a small preview image directly within the table.

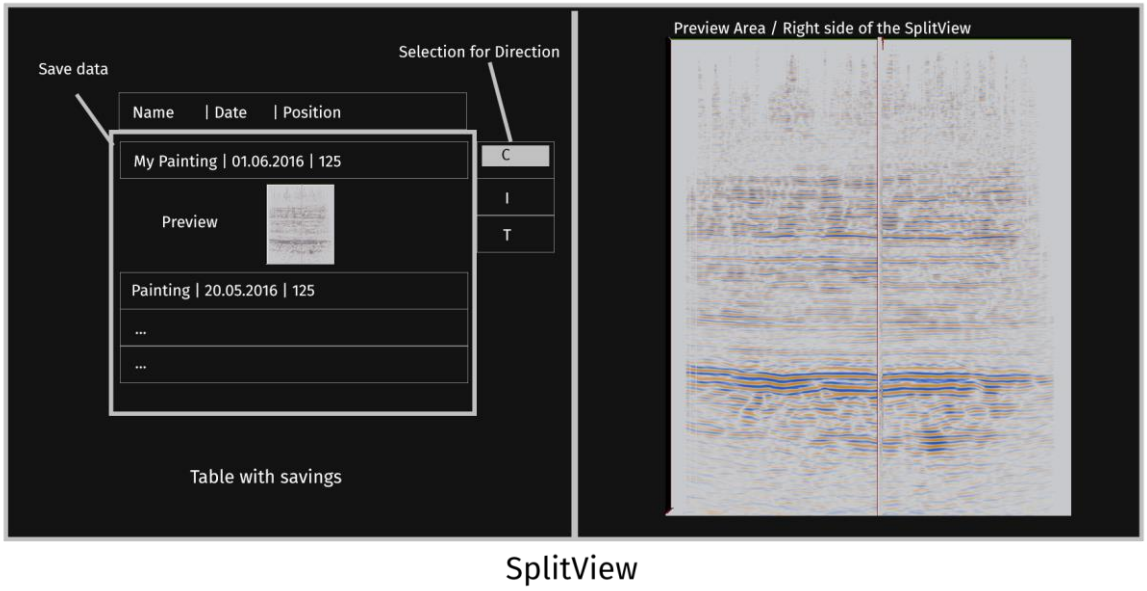


Figure 18: First draft for the so-called SplitView.

Nevertheless, for the next iteration of the prototype, the proposed design is reduced leaving out the preview in the table to quicken the implementation. Also, the seismic line is not used as a filter and appears as additional information directly in the row.

3.3 Optimization

For a more sophisticated interpretation process and to elevate the workspace closer to professional software, a set of further features is designed. All identified requirements are described in the user stories, but the most important features, concerning the main interpretation tasks, are:

1. A slice should be selectable and moved independently from the selected view direction (crossline, in-line, timeline).
2. The image should be bound to the position, changing the position of the slice should not result in transferring the painting to the new position.
3. The image should not be erased when the attribute of the slice is altered.
4. It should be possible to have multiple slices in a seismic line.
5. The user should be able to rotate the view.
6. It should be possible to erase the image.

To implement the listed features, the workspace requires a process of redesign, where existing techniques are adapted to the identified demands or recreated from scratch.

3.3.1 Selection and Manipulation

Most of the selection is designed to be achievable either with pen and touch or speech. Per Bowman et al. “[…] in a selection task, the interaction technique should provide the user with a means to indicate an object to select and to confirm the selection; and provide visual, haptic, or audio feedback while performing the task” (Bowman 2005, p. 148). For the extension of the prototype, already available selection tasks are designed to be executable via speech, increasing the flexibility of the system, e.g. selecting a different painting attribute could be done with the GUI or via speech. This sort of duality is also considered for new

features, e.g. the system is designed to activate the eraser using either speech or the designated button. Furthermore, manipulation tasks are also created including speech interaction, e.g. it is possible to remove a slice which is selected as active by touching the trash-icon in the Painting Area or by simply saying the command “Slice remove”. This sort of manipulation interaction naturally excludes some of the tasks, e.g. the interpretation process is achieved with the painting of attributes directly onto a slice, see Figure 19, and is not extended with speech features, though the already mentioned options, e.g. to select a different attribute, can be carried out by the user during the painting process.

Figure 19: Main task of the workspace, the painting of attributes on a slice.

Nevertheless, the usage of speech for interaction reshapes the concept of the Multimodal Seismic Interpretation Workspace. Following Dobin’s suggestions and the created user stories, speech commands can be split into three categories for the new concept:

1. Permanently available speech interaction, e.g. select a different attribute for the viewed slice.
2. Speech interactions which are only available when the user is gazing at a specific area, e.g. select a row in the table.
3. Speech interactions requiring the same command, but resulting differently based on the gaze of the user, e.g. transfer of the selected slice from the table area while looking left or the previewed slice from the right to the Painting Area.

Deictic references are not planned to be part of the fusion process to simplify the implementation, but could be included during further development as the possibilities to recognize and use deictic words in speech are designed to be available in the system.

3.3.2 Navigation

The navigation in the Painting Area is adopted from the previous work, and only slightly modified in its conceptualization. Thus, the user is enabled to pinch-to-zoom in the Painting Area with two fingers, and as a new feature, to rotate the view by circling one finger around another finger. When the user touches the screen with three fingers and moves his hand around, the view moves respectively resulting in panning, see Figure 20.

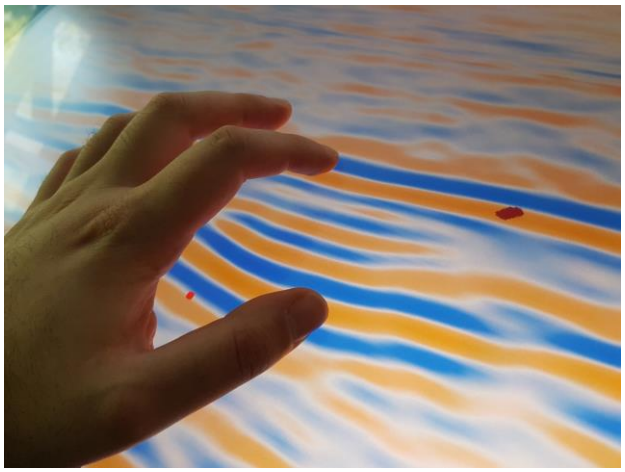


Figure 20: Panning the view with three fingers in the Painting Area.

The feature to move a slice in its seismic line is kept from the previous work, but expanded requiring the user to select a slice as active, which is a new feature. When the user selects a slice as active tapping with the pen on the slice, the lice becomes highlighted, see Figure 21.

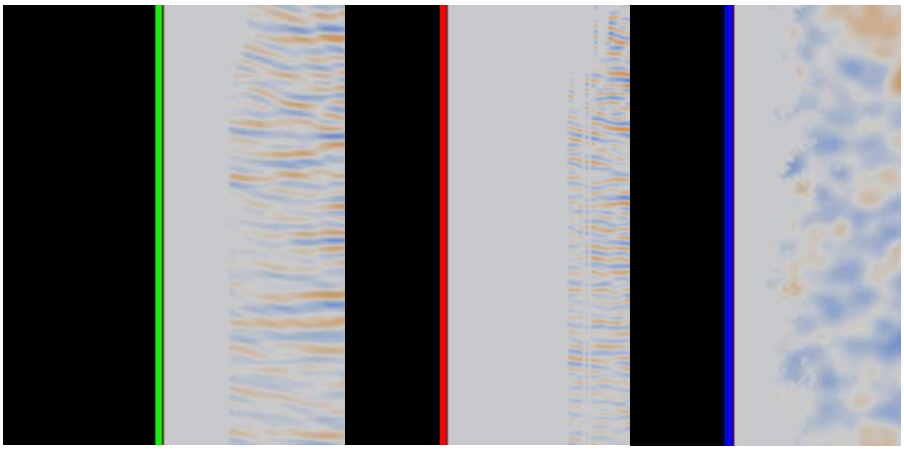


Figure 21: The edges of a slice are highlighted in the color of the seismic line (crossline is green, inline is red and time-line is blue) when the slice is selected as active.

When a slice is selected as active, the slider on the left of the Painting Area is updated and moves the selected slice henceforward.

3.3.3 Graphical User Interface

Jakob Nielsen presented ten heuristics for user interface design back in 1995 (Jakob Nielsen 1995). Some of these heuristics are:

- *Visibility of system status: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.*
- *Aesthetics and minimalist design: Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of Information in a dialogue competes with the relevant unit of information and minishes their relative visibility.*
- *Consistency and standards: Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow plattform conventions.*
- *User control and freedom: Users often choose system functions by mistake and will need a clearly marked "emergency exit"to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.*
- …

Considering all of the heuristics might be not possible for the optimization of the workspace, e.g. the implementation of undo and redo functionalities like it is postulated by the heuristic

"*User control and freedom"*, nevertheless, the heuristics proposed by Nielsen are respected during the design process.

The basic GUI implemented in the first prototype is replaced during the development as it had major issues. Figure 22 shows a screenshot of the GUI used in the first prototype. First of all, the GUI does not feature any feedback. This issue relates to the heuristic "*Visibilty of system status"* by Nielsen. No elements are highlighted after a selection, making it impossible for the users to figure out the current system status and selected options. Moreover, no other feedback mechanisms are included. Furthermore, the interaction via touch does not respond on every icon, revealing an technical issue which needs to be solved. An additional problem of the GUI is the usage of fixed values to position the menu, therefore, the system can only be operated at a predefined output resolution of 1920x1080 pixels.

As the negative aspects of the implementation presented with the first prototype predominate, the GUI is recreated from scratch.

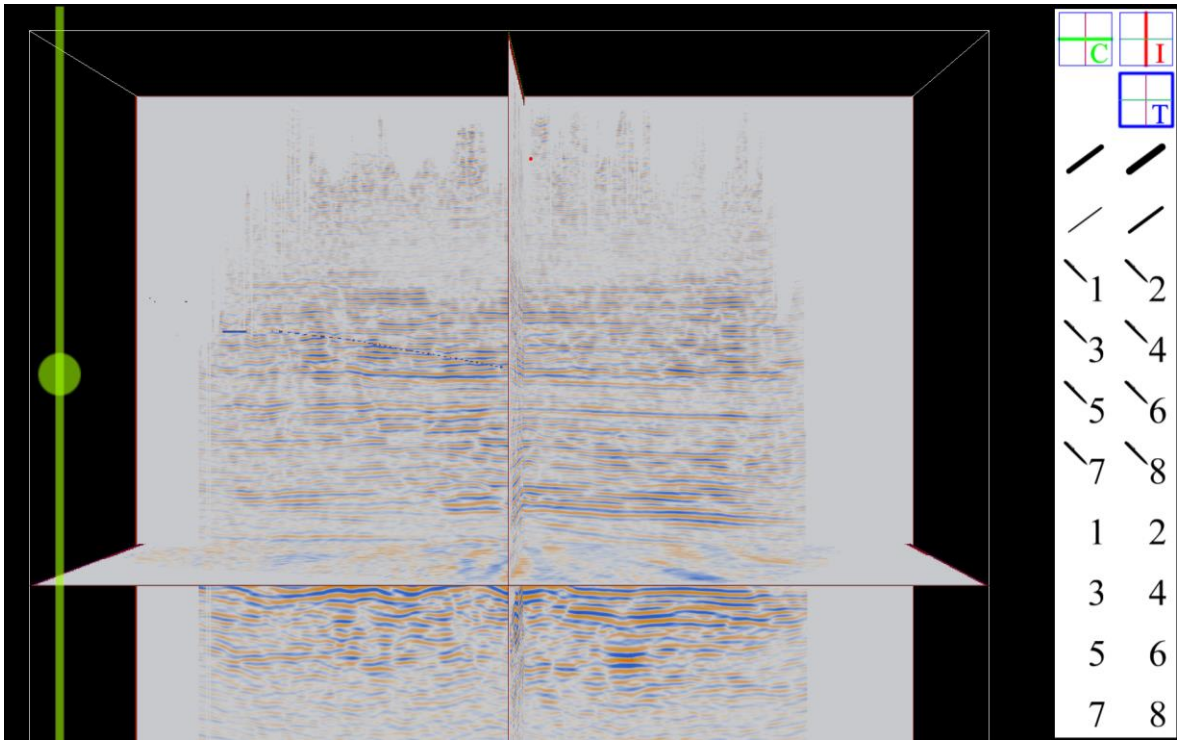


Figure 22: The GUI used in the first prototype.

To optimize the GUI, the options for the attribute selection for the pen and for the slices are split into two sub-menus by the draft. As more features are planned for the new prototype of

the Multimodal Seismic Interpretation Workspace, the sub-menus are designed to be collapsible. Thus, the user can hide the sub-menus revealing more of the workspace. Besides, the buttons are designed to show a preview of the attribute and are also labeled with the name of the attribute. In comparison, the first GUI labeled the buttons only with numbers, presuming that the user already knows which attribute is connected to the selected button.

For the rest of the GUI, the elements are decoupled from the single menu area. Most buttons are redesigned, using simple icons which already indicate the functionality, e.g. the commonly used save icon showing a floppy disk is implemented to save the active slice. Thus, the redesign of the GUI sticks to the "*Aesthetics and minimalist design*" heuristic by Nielsen.

The first prototype of the Multimodal Seismic Interpretation Workspace does not deliver any feedback to the user's selection, e.g. the selection of a different attribute to paint with can only be confirmed as soon as the user starts to paint after the selection task is completed. As a solution to this issue, a notification area is identified as a requirement for the next iteration of the prototype, see Figure 23 for a snippet showing the implemented notification area.

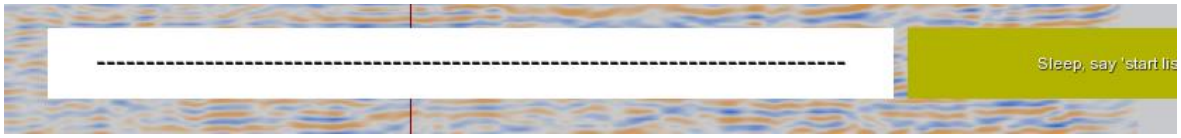


Figure 23: The notification area enables the user to receive feedback messages.

Furthermore, to increase the visual feedback to the user, displaying the status of the system permanently is necessary and therefore the notification area is designed to be a feedback solution reacting to the overall interaction. Providing feedback can be considered as one of the most basic guidelines for improving the usability of the system, according to Jakob Nielsen (Nielsen 2008). To optimize the overall usability of the Multimodal Seismic Interpretation Workspace, the selected attributes, seismic line, stroke width etc. should be highlighted for the user. Therefore, separate labels are implemented reflecting the current system status in terms of the mentioned values. In Figure 24 the user can easily read the status of the system, including the selected seismic line, the thickness of the drawing line and the selected attribute.

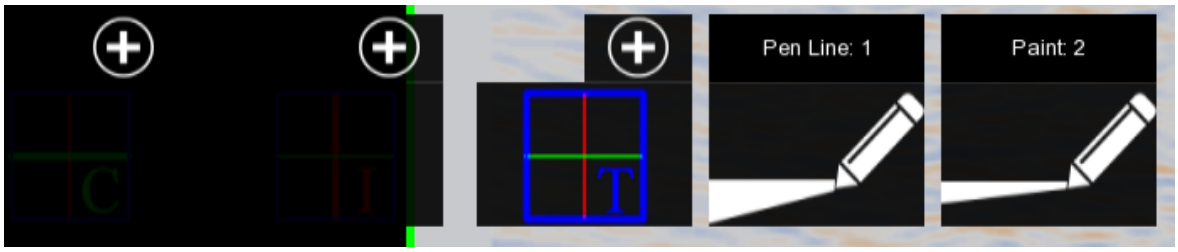


Figure 24: Visual feedback on the system status.

Visual feedback is also a requirement for the OverView Area. Therefore, symbols showing the gaze position are identified as necessary during the development, to recognize if the eye tracker is tracking the gaze of the user properly.

III. Implementation

4 Implementation

After a brief introduction to the overall system architecture, this section pictures issues and their solution during the implementation process. This section also presents the fusion engine, which is one of the core components of a system utilizing multimodal interaction.

4.1 System Architecture

For the implementation of the MMSIW, common programming patterns, e.g. the Observer-Observable Pattern, are included into the existing system with modifications. The system is mainly frame-based and all events are processed during the main loop call. Figure 25 visualizes parts of the workspace system as a flow-chart indicating the process of in- and output offered and delivered by the MMSIW. Each interaction technology is encapsulated with its own class implementation, using and integrating the corresponding API, e.g. voices recorded by the microphone are processed by the Microsoft Speech Recognition and are accessed via the Microsoft Speech API (SAPI) by an instance of the *SpeechListener* class.

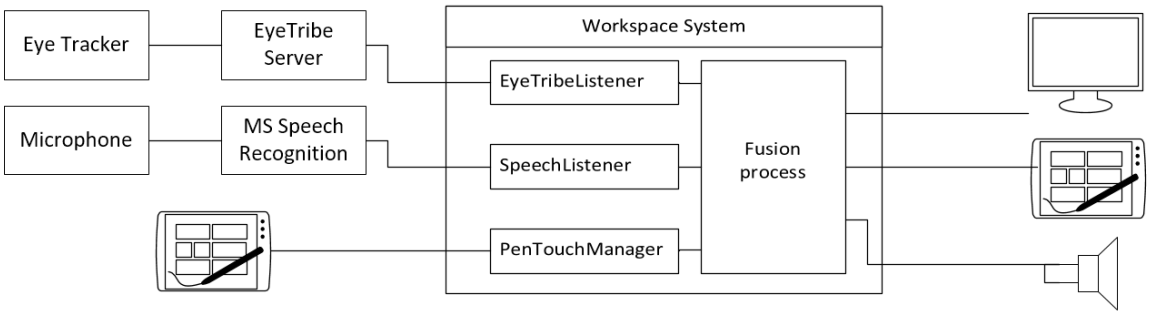


Figure 25: Abstract graphic showing the system architecture of the MMSIW.

The representation in Figure 25 is a very simplified graphic to illustrate the main flow. As a common and complete class diagram (e.g. created with UML) of the system would be too complex, multiple class diagrams for the most important components of the architecture are

created and presented in the upcoming sections. Nevertheless, it is important to get an overview of the structure of the system. Therefore, a graphic visualizing the software architecture is created in an unconventional way to illustrate software composition, see Figure 26.

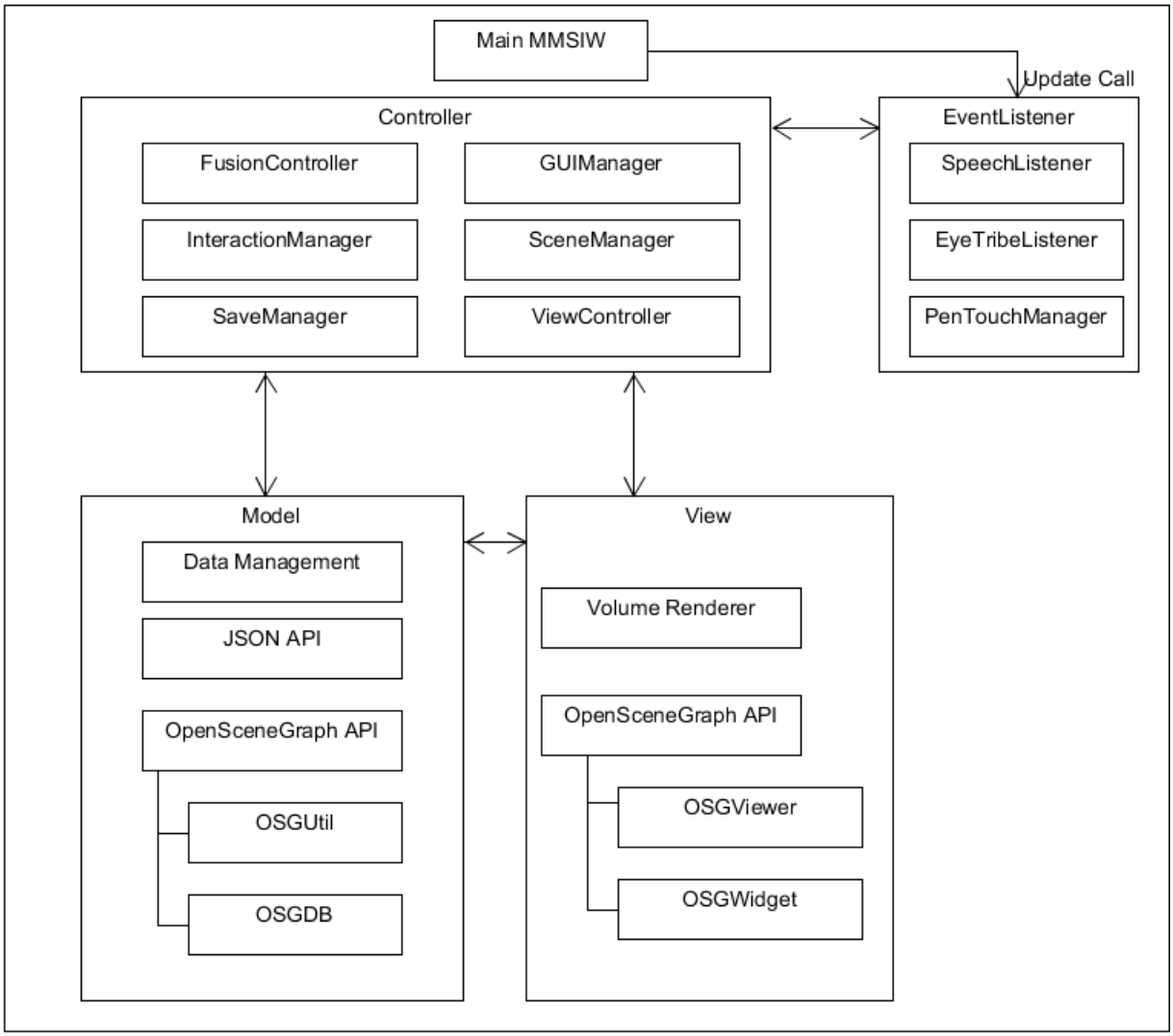


Figure 26: Software architecture of the MMSIW.

MMSIW uses two in-house developments of Fraunhofer IAIS, consisting of components for the import of the SEG Y data files, which is a file format by the Society of Exploration Geophysicists (SEG 2016), called *DataManagement* and a component which is used for the visualization, named *VolumeRenderer*. Both components are also integrated into the software architecture in Figure 26. The implementation is structured according to the Model-

View-Controller pattern. Multiple instances of controller classes regulate the system's behavior based on the events received during the main loop. Each controller class is created with a distinct purpose in mind, e.g. only the GUIController is in control of the GUI elements.

4.2 Painting Area

The main task, which is the interpretation of seismic data, is carried out in the Painting Area. The implementation of new features is discussed in this chapter.

4.2.1 Pen and Touch

The pen and touch features are implement using hardware from the Wacom company. A Wacom Cintiq HD Touch device with 27" display is utilized for the Multimodal Seismic Interpretation Workspace. The device offers multi-touch interactions, which can be incorporated to implement gestures. The relatively large display area of the Wacom Cintiq unit is surrounded by thick edges, enabling the user to rest his arms comfortably on the device (Wacom 2015). A product shot of the Wacom Cintiq QHD 27 can be seen in Figure 27.

Figure 27: The Wacom Cintiq QHD 27 Touch, (Wacom 2015)

Wacom offers resources and documentation for the low-level API to retrieve the data from the device in custom programs for further processing (Wacom 2016b). The API is also supported by a set of recent Wacom products supporting pen and touch, e.g. the Intuos Art series (Wacom 2016a), which are also used during the development. A dynamic library already offering the possibility to receive the pen and touch data has been developed at Fraunhofer IAIS in the past and has been already included in the solution.

4.2.2 PenObserver

An update to receive available pen and touch data packets from the Wacom API is executed during each main loop iteration of the application. Available pen data packets are then processed by the instance of the *PenObserver* class by executing the method *processPenData.* In order to determine the pixels the user hits with the pen, an intersection testing is carried out, see Code 1.

```
1.  osgUtil::LineSegmentIntersector::Intersection* pIntersection = NULL;
2.
3.     pIntersection = mInteractionManager->pickSliceIntersectionAtNormalized
    ScreenCoords((penPosX + 1.0f) / 2.0f, (penPosY - 1.0f) / - 2.0f);
4.
5.     if(pIntersection != NULL)
6.            {
7.            mAttributePainter->paintAttributeAtIntersection(pIntersection);
8.
9.            mCurrentDelay = mDelay;
10.            }
11.            else
12.           mAttributePainter->stopPainting();
```

Code 1: Retrieving intersections for the painting process, PenObserver.cpp.

In line 3 of Code 1, the program tries to retrieve a slice using normalized screen coordinates, which are extracted from the pen data packet. If the result contains an intersection object , see line 5, the name of the intersection is then used by the object of the *AttributePainter* class to get the required slice object from the *SceneManager* instance, see line 7, for the actual painting. Therefore, the painting process involves several objects to be realized.

The painting process runs fine if the workspace is executed with a single-display setup, but the multi-display setup of the Multimodal Seismic Interpretation Workspace requires a re-programming, as the already implemented solution for the *Direct Volume Rendering* does not work as expected. Both instances of the class *osgViewer::View* are used to display the same scene graph in the OverView Area and in the Painting Area, but with separate cameras

for the rendering. A scene graph is a common data structure and used in most graphics applications. Scene graphs structure and represent a scene with a tree of nodes, where each node represents an object in the scene. These can be visible geometry nodes, like the slices in the Multimodal Seismic Interpretation Workspace, or invisible objects with different functions than the representation of geometry, e.g. cameras, transformation nodes etc. Please refer to the article “Scenegraphs: Past, Present, and Future” by Avi Bar-Zeev for further information on scene graphs (Bar-Zeev 2007).

Changes made on the geometry, like the altering of the texture when the user paints, become visible in both views, which is an advantage for this solution. In comparison, if two different scene graphs would be used, the changes would require a synchronization between the two scene graphs for each node. Using only one complex scene graph is also considered to be preferable when it comes to performance, but this assumed advantage is not further investigated.

Using only one scene graph for both views also entails a disadvantage. When the user draws something in the Painting Area, the formerly described painting process, including an intersection detection, is conducted. Basically, the intersections between a ray and geometry instances in a scene are used to determine a list of nodes (OpenSceneGraph 2007a). As two different instances of the osgViewer::View class are used to realize the multiple output, the traverse results in a list of nodes, as if the ray was shot from the perspective of both cameras. Therefore, as the first intersection with a slice is being used to paint, the painting becomes visible at the slice which is closest to the cameras in the scene. To remove this unwanted behavior, the exclusion of the camera used in the OverView Area from the intersection traversal is required. This can be done with the use of so-called node masks. Using node masks, the sub-graphs of the scene graph can be traversed selectively (OpenSceneGraph 2007b). The difficulty which emerges with the inclusion of node masks is, that the camera which is used by the OverView section needs to be excluded from the intersection testing, but still be forced to render the scene, as node masks are also used to cull objects from the rendering process. The issue becomes even more complex, as the node mask solution is already in place, removing the Direct Volume Rendering in the Painting Area from the rendering process. To solve this issue, the Volumerenderer used for the visualization of the seismic data, also an in-house development of Fraunhofer IAIS, needs to be adjusted.

The node masks of each node are compared to the node mask of the camera, when a traversal is conducted. The result of the logical AND comparison, the official tutorial on the web site of OpenSceneGraph mentions logical OR by mistake (OpenSceneGraph 2009), excludes all sub-nodes of a node, if the result is Boolean false. Visualizing the scene graph of the system reveals that numerous cameras are used for the overall rendering, see Figure 28.

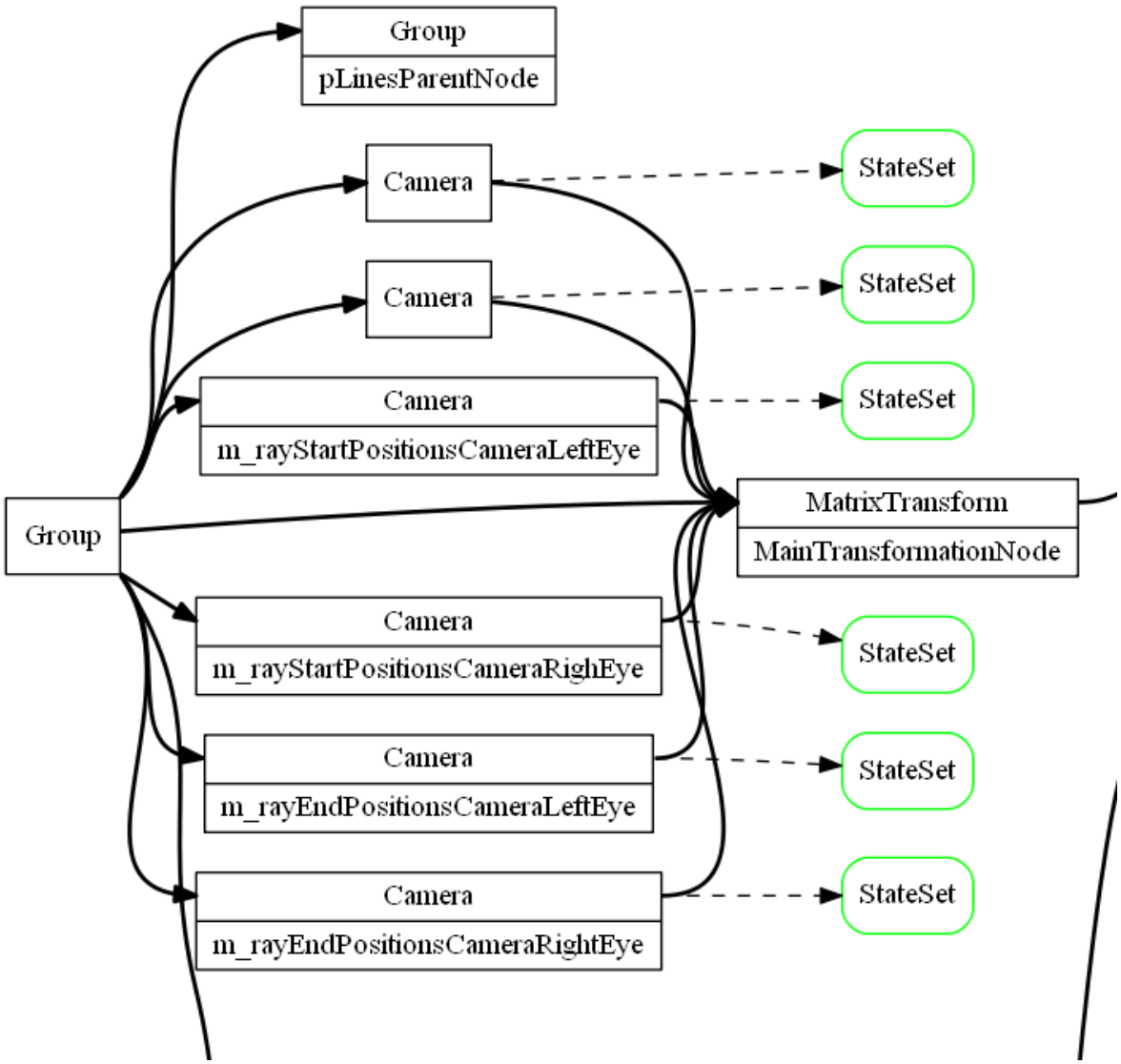


Figure 28: The highest levels of the scene graph, revealing the usage of multiple cameras for the rendering,

Four cameras are used for the ray starting and ending points in the Direct Volume Rendering process. As these cameras are on the same level as the main cameras, named *camera* in the figure, the ray starting and ending point cameras must be excluded from the process as well, to achieve the desired behavior. With the logical AND being used, the node representing the

Direct Volume Rendering receives the node mask with the hexadecimal value FB. To cull the Direct Volume Rendering in the Painting Area, the camera of the Painting Area receives a value of ~FB, ~ negates FB. Therefore, the logical AND results in zero and the Direct Volume Rendering is not rendered. To exclude the nodes from the intersection traversal, the intersection testing is run with the node mask value of 4, see Code 2.

```
1. if (mWacomView->computeIntersections(mWacomView->getCamera(),
          osgUtil::Intersector::PROJECTION,normalizedScreenPosX ,normalized-
          ScreenPosY, intersections, 0x4))
```

Code 2: Intersection testing

Logical AND 4 has a result of zero with FB and 4 with ~FB, therefore, by setting all node masks of all cameras which are used in the OverView Area to 4 and for the Direct Volume Rendering to ~FB leads to the desired painting behavior of the system.

4.2.3 Preserving Paintings

When a user paints a slice in the first prototype and moves the slice in the seismic line direction, the painting is preserved over all positions. Furthermore, changing the attribute of the slice results in the deletion of the painting. The observed behaviour is unintended for the new prototype. Thus, requiring a new approach to save a painting coupled to its position. To achieve the expected behaviour of the system, separate textures are created for each image and evaluated during the loading process for the slice. Therefore, it is possible for every available slice in the scene to create buff textures holding the information on the image. It is also possible to place multiple slices in the scene, leading to an increased amount of buffered textures within a few interaction steps. This sort of behaviour needs to be treated like an issue, as it might decrease the systems performance easily. A solution would be to remove the texture buffering from the RAM, keeping only a few textures, e.g. which have been manipulated recently. All other available textures would then be stored on the hard disk, which would result in longer loading times when these textures would be required. As the MMSIW application is a research project with non-commercial focus, this issue is rated as low-priority and pushed back in the concept. In addition, the system could also be limited to a number of slices in the scene, or in terms of painted textures stored in the background process, but no restrictions are implemented in the current iteration.

4.2.4 Touch

The *TouchObserver* uses the instance of the *InteractionManager* to trigger the manipulation of the ViewMatrix based on recognized and processed touch input.

4.2.5 Palm Rejection

A solution to reject the palms of the user and therefore to reduce unwanted touch interaction was already implemented by the previous work. The class *PalmRejection* is altered during the development to support dynamic resolutions. During the development, an unwanted behaviour of the system is observed, the rejection of palms behaves unexpectedly in some situations. This leads to the detection of palms by the system, although none are touching the Wacom device. Therefore, the system stops reacting to touch input in some areas of the display. The reason for this issue cannot be detected, as its occurrence does not follow any pattern. To keep sure that the system stays operational, the information collected on detected palms in the *PalmRejection* Class instance are deleted after a short time without any touch inputs being received by the MMSIW application.

4.2.6 GUI

To optimize the Graphical User Interface of the workspace, the GUI is developed from scratch. This procedure includes the realization of a new class to handle the GUI, which is named *GUIManager*. The *GUIManager* handles the elements displayed on all available views, including the OverView Section. Although an application framework for UI development, e.g. QT (Qt 2016), could be used for the workspace to develop a GUI, a simple widget library for objects like labels, tables, buttons etc. called *osgWidget* is already included into the OpenSceneGraph API (OpenSceneGraph 2010) and therefore used for the implementation. A quick review on the library, by testing some examples, surfaces sufficient functionality for the development of a GUI. The GUI of the workspace consists of buttons, labels, a table, objects hinting the gaze position and a slider for the repositioning of the active slice.

4.3 Overview Area

Features and principles implemented for the realization of the Overview Area are discussed during this section, including the saving feature, which is a requirement for the SplitView concept.

4.3.1 Saving Process

One feature which is required in order to implement the SplitView is the possibility to save a painted slice. To keep this feature as simple as possible, a file containing JSON data is used to keep track on the slice data. JSON is a common data-interchange format (JSON 2016). A different approach to save the necessary information could be the usage of a SQL database. An entry to the saved data file consists of parameters like the attribute of the slice, the date, the seismic line and the position in the seismic line, see Code 3.

```
1. {
2.  "Slice" : [
3.  {
4.  "attribute" : 1,
5.  "date" : "13-11-2016 20-41-39",
6.  "id" : 1,
7.  "lineType" : 2,
8.  "position" : 375,
9.  "src" : "./data/saves/save1479066099.bmp"
10. },
11. ...
12. }
```

Code 3: Save entry, savedata.json.

The most important entry is the relative path to a bitmap file, which is created during the saving process containing a texture. This texture file is required to restore the painting. To display an loaded attribute at a specific pixel in the texture, the *VolumeRenderer* requires the corresponding integer value to be injected into the first color channel of the pixel. The required values are stored in *Image* files, provided by OpenSceneGraph via the osg::Image class, which is basically a class for the encapsualtion and storage of texture image data (OpenSceneGraph 2016). Everytime the user moves to an unvisited position of a slice, a new Image class instance is created and stored in the container *mPaintings*. Thus, the attempt to change the selected attribute of a alice triggers a lookup in the container using the provided position as a key, see line 7 in Code 4.

```
1.  for(unsigned int s = 0; s < mAttributeNumberImage->s(); s++)
2.     {
3.            for(unsigned int t = 0; t < mAttributeNumberImage->t(); t++)
4.            {
5.                  unsigned char* pixel = mAttributeNumberImage->data(s, t);
6.
7.                  std::map<int, osg::Image*>::iterator it = mPaint-
    ings.find(mPosition);
8.
9.                  if (it != mPaintings.end()) {
10.
11.                       unsigned char* buff = it->second->data(s, t);
12.                       if (buff[0] != 99) {
13.  pixel[0] = buff[0];
14.                       }
15.                       else {
16.                             pixel[0] = attributeNumber;
17.
18.                       }
19.                 }
20.                 else {
21.                       pixel[0] = attributeNumber;
22.                 }
23.
24. ...
25.
```

Code 4: Retrieving saved painting information in the image, Slice.cpp.

When a saved painting Image is retrieved, a pixel-wise lookup sets the pixel of the displayed texture as expected, see line 9-22 in Code 4. The visualization of the texture is processed in the *VolumeRenderer* component. One possible improvement to this implementation could be the usage of the other color channels available by the allocated *Images* to reduce the required memory.

A full diagram showing all classes involved in the save and load process can be seen in Figure 29.

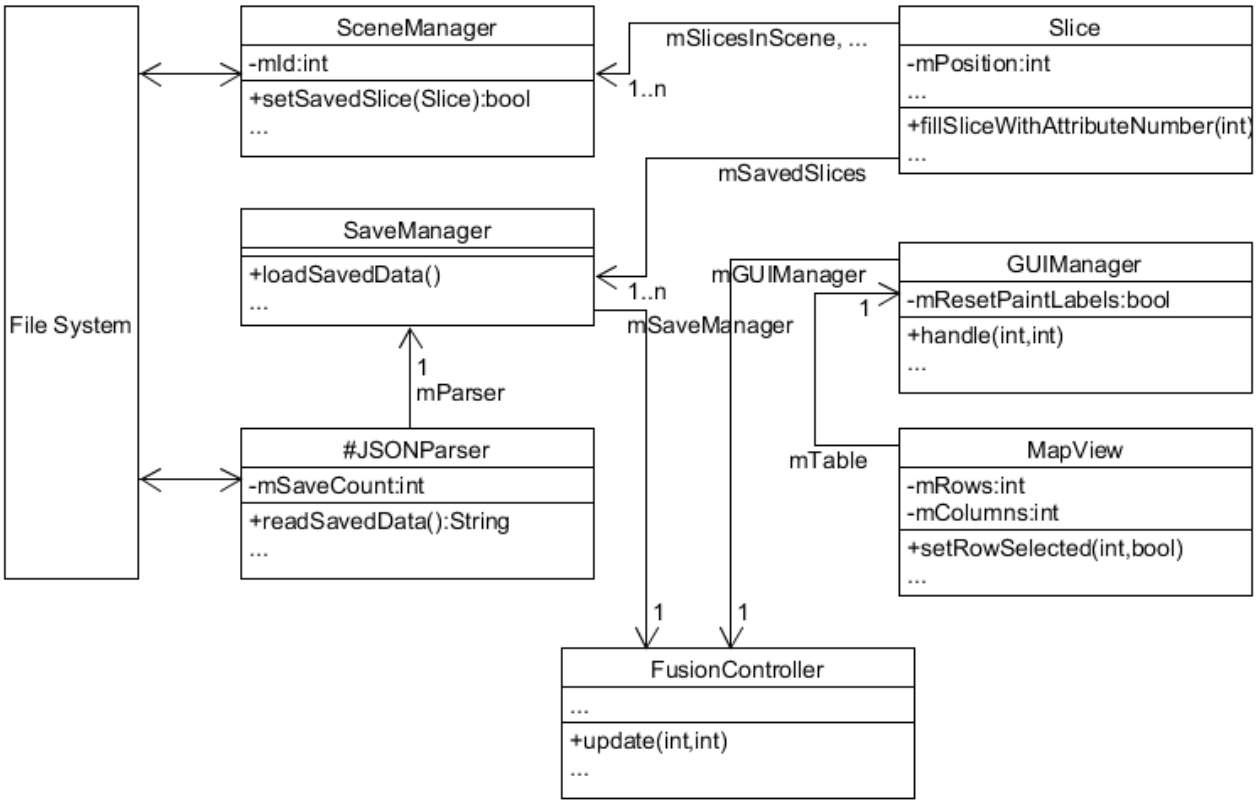


Figure 29: Part of the software architecture with classes involved in the save and load processes.

For the saving process, a new class called *SaveManager* is created, handling the saving and loading of slices in conjunction with the *JSONParser* class, capable of manipulating the local JSON file. Furthermore, an instance of the JSON Parser class can also create a set of slice objects, which are stored in the SaveManager. The SaveManager provides functions to retrieve the loaded slices for further processing, e.g. the FusionController uses the SaveManager to retrieve a pointer on the loaded slice container, forwarding the pointer to the GUIManager, which then updates the *MapView* instance to display the loaded entries.

The implemented *MapView* table differs from the concept presented in chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**, see Figure 30.

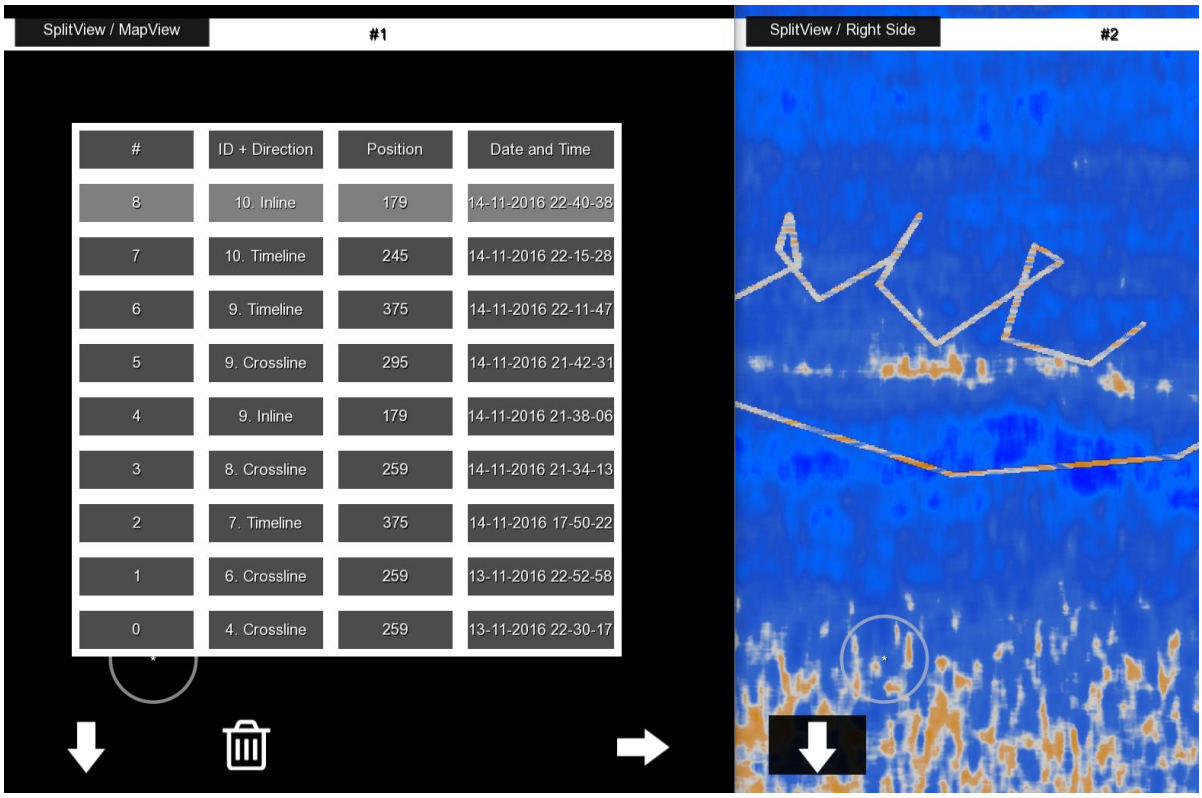


Figure 30: The table "MapView" listing saved slices.

A data row in the table consists of a number, which is used during the selection process, a column including an id and a direction, the position in the seismic line and the date and time when the slice was saved. The selected row is highlighted and the trash icon is used to remove the selection. The buttons showing arrows push the selection to the desired area.

4.3.2 Preview

For the preview on the right side of the *SplitView*, a second instance of the *InteractionManager* class and a second scene graph is created to achieve an independent visualization and interaction. Based on the gaze of the user, the instance of the *EyeTribeListener* class updates a value representing the currently viewed area by the user. The *TouchObserver* uses the gaze information to process the received touch-input in combination with the correct InteractionManager instance. This solution also requires a second VolumeRenderer instance for the visualization, decreasing the performance of the system.

4.3.3 Eye Tracking

On the Overview Area, the user of the Multimodal Seismic Interpretation Workspace is enabled to influence his speech interaction based on his gaze. For this purpose, the so-called Eye Tribe, designed and created by the Danish company The Eye Tribe, is utilized for the project (The Eye Tribe 2016). The Eye Tribe Tracker uses infrared illumination to detect the gaze of the user, see Figure 31. Refer to a review paper by Dubey and Chhabria on currently available eye tracking technologies and their principle functionality for further information on eye tracking (Dubey, Chhabria 2014). For the Multimodal Seismic Interpretation Workspace, the first version of the tracker is implemented. An improved version of the tracker is currently available for preorder.

Figure 31: The EyeTribe eye tracker utilized for the MMSIW

The class *EyeTribeListener* uses the SDK provided by the The Eye Tribe company to connect to a locally executed server application, called the *EyeTribeServer*. Thus, requiring the user to start the server application before opening the MMSIW application to use the eye-tracker. Furthermore, the tracker can be calibrated with an application called *EyeTribeUI* which helps to find the optimal tracking position for the sensor. When everything is set up correctly, the EyeTribeListener class in the MMSIW application receives data packets constantly. The data packets include several tracking information, e.g. the state of the tracking, see line 1 in Code 5, and of course the position of the gaze on the display, which are used to determine the currently viewed area.

```
1.  if (gaze_data.state == gaze_data.GD_STATE_TRACKING_FAIL || gaze_data.state ==
    gaze_data.GD_STATE_TRACKING_LOST)
2.
3.  ...
4.
5.                  if (gaze_data.avg.x >= OVERVIEW_DISPLAY_WIDTH/2)
6.                  {
7.                        mFusionController->update(253,MMWE::RIGHT_VIEW_MODE);
8.                        mFusionController->handleEyeInput(gaze_data.avg.x,
    abs(OVERVIEW_DISPLAY_HEIGHT-gaze_data.avg.y));
9.                        return;
10.                 }
```

Code 5: Accessing the gaze data states, EyeTribeListener.cpp.

Furthermore, the received position values are used to update the hint symbol's position with the *FusionController*, creating feedback to the user on the tracking, see line 8 in Code 5.

4.3.4 Depth Perception by Parallax Movement

One feature available with the Volume Visualizer of the Fraunhofer IAIS in-house development is the possibility to view the Direct Volume Rendering in stereoscopic 3D. One feature which is planned with low priority, but investigated during the development of the prototype is the possibility to use the EyeTribe tracking data in order to reposition the ViewMatrix, thus, leading to increased immersion by creating a depth effect, based on parallax movement. This idea is motivated by the fact that 3D stereo requires active glasses in the current prototype setup which is incompatible with the EyeTracker, as the tracker cannot track the eyes of a user due to the glasses.

The ability of a human being to perceive depth can be created by several mechanics and abilities of the eyes and the brain. The depth effects can be split into motoric and visual effects. Some visual effects, which, furthermore, can be categorized as binocular and monocular perception effects, can be recreated for a viewer. The most common way to achieve images with depth is by using the effect of binocular disparity. Our eyes receive the same image, but due to the position of the eyes, the perspective is viewed by each eye is slightly different. Nevertheless, the images of both eyes are combined to a single image, a process called binocular fusion (Ciccarello, Berger 2016). Binocular fusion results in images for the viewer with depth information. This can be recreated by simply showing two different, but properly repositioned images to the viewer.

3D supporting technologies can be put into two main groups, the active and the passive technologies. The shutter technology is widely used, as it only requires a display with a refresh rate of at least 120 Hz. Centerpiece of the active technologies are glasses; the viewers need to wear these in order to perceive a 3D image. The lenses are darkened in synchronization with the shown images on the display, which are showing different perspectives in alteration. As the main application utilizes OpenSceneGraph and therefore, the OpenGL API, active stereo is supported only with quad-buffered stereo. Quadbuffered Stereo can be viewed in a window, while the Direct3D APIs do not necessarily support the technology. Only graphics cards with the support of quad-buffered 3D Stereo can execute the application in 3D stereo, e.g. graphics cards of the Quadro series by Nvidia. As active 3D stereo requires the user to wear glasses, the EyeTribe Tracker is not able to track the users gaze.

In comparison to active technologies, one of the most common passive technologies is using polarization filters, which single out lines of the display, and therefore split one image into two separate per each eye. One interesting approach in passive technologies are auto-stereoscopic 3D displays, removing the necessity of wearing glasses for 3D stereo. Although the handheld video game console 3DS by Nintendo introduced an auto-stereoscopic display to the mainstream using a parallax-barrier display (Nintendo 2016), large displays are still very uncommon. Some vendors like Tridelity (Tridelity 2016), offer displays with auto-stereoscopic 3D in larger dimensions. Nevertheless, to overcome the constraints imposed by the usage of active stereo glasses in the MMSIW, a monocular implementation using the effects of parallax movement is proposed in order to create depth in the visualization.

The underlying mechanics of motion parallax are described by Kelnnhofer et al., who illustrated the translation of a pair of points in Figure 32, producing motion parallax. In the figure, A is assumed to be a fixation point with a specific retinal position. As the fixation points moves, the retinal position remains the same, as the eye pursuits the fixation point by rotating itself by $d\alpha$. The velocity of this change is therefore $\frac{d\alpha}{dt}$. As the fixation point B is closer to the eyes, the angular distance of the retinal positions is θ, which produces a retinal motion with velocity $\frac{d\theta}{dt}$. The relation for motion parallax is $\frac{d\theta}{d\alpha} \approx \frac{\Delta f}{f}$, allowing the recovery of the depth with Δf.

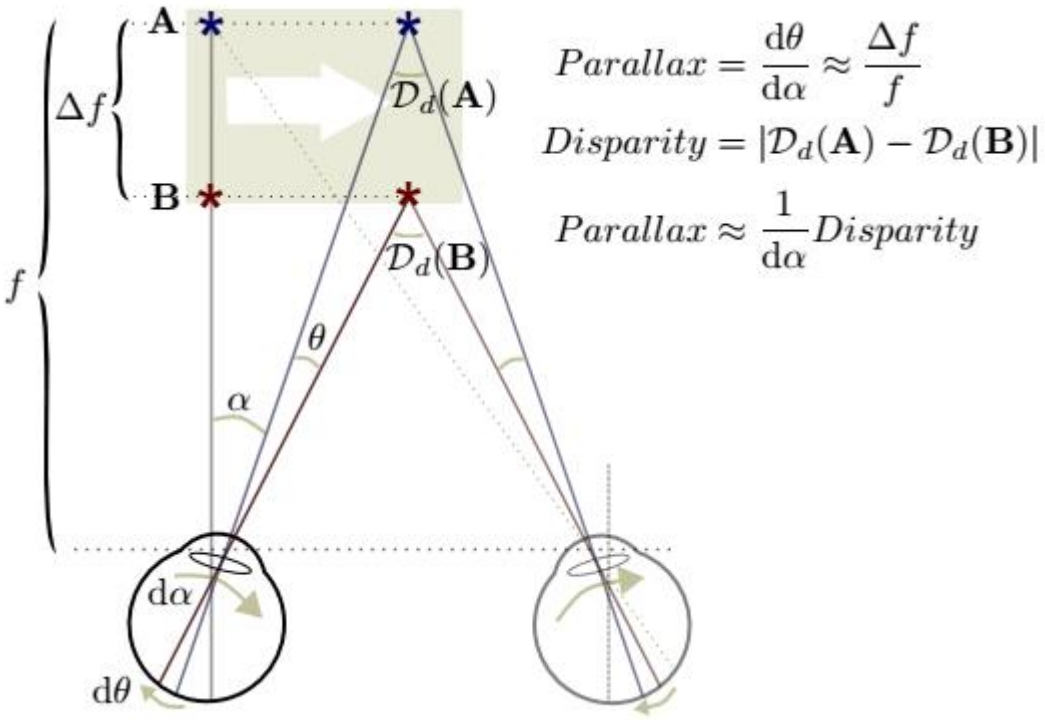


Figure 32: The mechanics of motion parallax, (Kellnhofer et al. 2016).

Several projects introduced the usage of the parallax motion. One project utilizes the Wii-motion controller for head-tracking and is developed by Johnny Chung-Lee (Chung Lee 2008). The program offered by Chung-Lee uses some parameters like the size of the display and the aspect ratio and requires the user to initially calibrate the program. Based on the movement of the detected Wii motion sensor, the projection matrix is recalculated creating the parallax movement effect. Although the idea to use the EyeTribe eye-tracker to implement a parallax motion solution to the MMSIW seems appropriate at first sight, some disadvantages of the hardware make an implementation undesireable. The Wii-Mote solution presented by Chung-lee uses the movement of the sensor, transferring it to the eye-tracker would require the viewer to gaze at one point, moving the fixation point very cautiously. A more sophisticated solution would incorporate head-tracking. Due to the low-priority status of this idea, the implementation is dropped in favor of different hardware as a possible component in future research. An eye-tracker in the same price-range as the Eye Tribe is presented by the company Tobii (Tobii 2016). The newest generation of the Tobii eye-tracker also includes discrete head-tracking and is much more promising for a possible implementation of a parallax movement solution (Sauter 2016).

4.4 Speech Control

Shortly after the VRGeo consortium meeting in March 2016, the search for appropriate speech recognition APIs began. During the past years the availability of speech based services increased. Especially digital assistants like Siri by Apple, Google Assistant or Google Now by Google or Cortana by Microsoft established interaction across multiple device types, incorporating voice recognition. During the Apple Keynote WWDC (Apple Worldwide Developers Conference) 2015, Apple's Senior Vice President of Software Engineering Craig Federighi stated, that the digital-assistant Siri gets over 1 billion requests a week (NDTV 2015). Federighi's statement is supported by a report available from the consumer technology research firm Park Associates, which has been published during the fourth quarter of 2015. Per the article written by Daniel Kobialka on the report, "[…] more than half of iPhone owners use Apple's (NASDAQ: AAPL) Siri voice recognition software, while less than one-third of Android phone owners leverage Google Now (NASDAQ: GOOG)." (Kobialka 2016). The article indicates a low number of Android phone owners using Google's voice assistant, but considering the massive smartphone market, according to IDC 301.9572 million smartphones running the Android OS only in Q2 2016, the potential audience using Google's digital assistant remains very large (Ljamas et al. 2016). Therefore, the large distribution of the digital assistants influences the design process of the voice recognition features provided by the Multimodal Seismic Interpretation Workspace.

4.4.1 Microsoft Speech API and Alternatives

As the workspace is developed in C++, the search for a speech recognition API is focused on libraries using C++. Using a C++ library prevents the need for additional API's connecting the different programming languages and reduces the complexity of the software architecture. Furthermore, Mark Dobin stated during the VRGeo consortium meetings in March 2016 and September 2016, that the multimodal interaction offered by the workspace should not be tied to the program itself. The user should be enabled to interact with other software installed on the system and quickly switch between the Multimodal Seismic Interpretation Software and e.g. a notepad application, an e-mail application etc. One available voice recognition library is Voce. The library offers cross-platform development supporting Java and

C++ and is open source (Streeter 2016). Unfortunately, the integration of Voce to the Multimodal Seismic Interpretation Workspace would only be possible if additional software would be programmed.

A different approach to achieve desktop applications featuring voice control is to use the Microsoft Speech API or SAPI (Microsoft 2016b). SAPI offers engines for text-to-speech (TTS) and speech recognition. The SAPI provides two different types of speech recognition engines. Microsoft recommends the usage of the shared recognizer. Multiple applications can use a shared recognizer, which leads to the possibility to use the speech recognition mode of Microsoft Windows for tasks e.g. opening a WordPad and dictating a text, and to use speech recognition in the Multimodal Seismic Interpretation Workspace application (Microsoft 2016c). For further development, the new Speech Platform included into Windows 10, powering the digital assistant Cortana, should be considered as a newer approach for speech recognition in the Multimodal Seismic Interpretation Workspace (Metulev 2016). As the workspace is currently required to run on PCs running Windows 7 and is not designed as a Windows 10 app, the usage of the Windows 10 speech platform is ruled out.

4.4.2 First Version

The development of the speech recognition features start with the integration of SAPI into the solution. SAPI is already pre-installed on systems running Windows 7 and upwards. For some examples, demonstrating the usage of the API, the download and installation of the Windows SDK is required (Unknown 2010).

To keep the speech recognition feature close to mechanisms already known by most of the users, an initiating keyword is defined in the corresponding grammar file. Saying out loud the word “listen” will start the speech recognition for four seconds, waiting for further commands. After each recognized and executed command, the speech recognition transitions back to the initial state, only reacting to the “listen” command. The reset into the initial state is also processed if the four seconds in active mode pass without any voice commands being recognized by the system, see Figure 33.

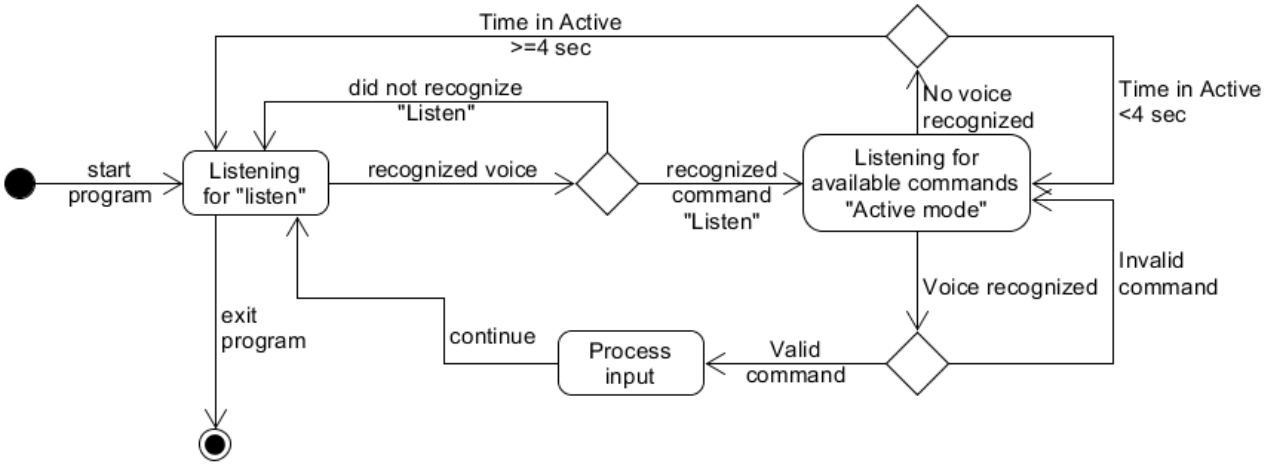


Figure 33: FSM showing the speech recognition process implemented in the MMSIW.

A grammar file is basically a XML-File which is processed by the executable program *gc.exe*, resulting in a *cfg*-File. The cfg-File is included as a resource into the Visual Studio solution and used during the initialization of the recognition context. Only words and compositions included in the grammar file lead to events in the created context. Code 6 shows the definition of grammar entries. Each entry consists of a name and a value. In line 1, an ID for the proposed language needs to be declared. The system needs to use the same language as defined in the grammar file for speech recognition to keep sure that commands can be recognized.

```
1. <grammar LANGID="409">
2.  <DEFINE>
3.  <ID NAME="VID_Push" VAL="201"/>
4.  <ID NAME="VID_Push_Options" VAL="101"/>
5.  <ID NAME="VID_Push_down" VAL="16"/>
6.  <ID NAME="VID_Push_right" VAL="17"/>
7.  <ID NAME="VID_Pen" VAL="202"/>
8.  <ID NAME="VID_Pen_Options" VAL="102"/>
9.  <ID NAME="VID_Pen_thicker" VAL="18"/>
10. <ID NAME="VID_Pen_thinner" VAL="19"/>
11. ...
```

Code 6: Definition of grammar entries.

The defined entries are connected to rule entries by using the name ID, see line 2 in Code 7. A rule consists of optional (<O>) and non-optional (<P>) elements. A rule can also require the recognition of a property or options, see line 14 to 19 in Code 7.

```
1. ...
2. <RULE ID="VID_Push" TOPLEVEL="ACTIVE">
3.  <O>Please</O>
4.  <P>
```

```
5.  <L>
6.  <P>Push</P>
7.  <P>Show</P>
8.  </L>
9.  </P>
10. <O>on</O>
11. <O>to</O>
12. <RULEREF REFID="VID_Push_Options" />
13. </RULE>
14. <RULE ID="VID_Push_Options" >
15. <L PROPID="VID_Push_Options">
16. <P VAL="VID_Push_down">down</P>
17. <P VAL="VID_Push_right">right</P>
18. </L>
19. </RULE>
20. ...
```

Code 7: Rule definition in the grammar file.

Therefore, the example shown in Code 7 leads to a recognition of the rule "VID_Push", when the user says at least one valid composition, which are (for the example "push"):

- "Please push on down" or "Please push on right"
- "Please push to down" or "Please push to right"
- "Push on down" or "Push on right"
- "Push to down" or "Push to down"
- "Push down" or "Push down"

When the recognizer detects one of the proposed combinations, the defined values can be extracted from the event and used for further processing.

Obviously, not all combinations make linguistically sense. This issue should be solved with more complex grammar files, including more rules. Nevertheless, the presented grammar file is sufficiently complex for the targeted research. A full list of all available speech commands can be found in the appendix section of this thesis.

The results of the speech recognition tend to be more accurate when the system is being used by the same person over a longer period, thus training the system. Hence, speech recognition offers individual recognition profiles, which should be considered for user studies, see Figure 34.

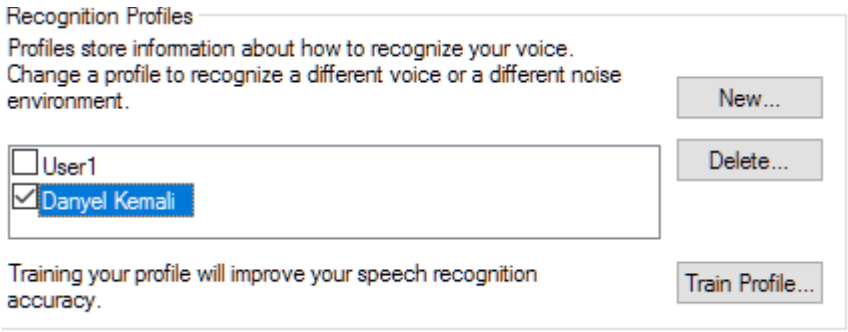


Figure 34: Individual training profiles for speech recognition.

Although the recognition of speech might be insufficient at first sight, the quality of the recognition might be increasable over a period of usage.

4.4.3 Second Version

The first version of the implemented speech feature received negative feedback from some users who were asked to try out parts of the system during the development process. The need to activate speech with the command “listen” for each command is criticized as being “annoying” and “flow-breaking”, as it is not possible to chain commands. Furthermore, the speech recognition system keeps running in the background, awaiting commands to control the operating system, although the MMSIW indicates that speech control is turned off. Indeed, the MMSIW application only waits for the command “listen” when it is “turned” off in the application, but this approach requires speech to be kept running in the background to wait for the command “listen”. Therefore, some users experience unwanted behavior during their short testing’s, e.g. by monologizing, triggering interaction with the operation system, e.g. opening the start menu of Windows. To optimize speech, the concept is altered during the second iteration. The idea is to link the speech interaction of the workspace with the speech recognition service of Windows. Thereby, turning off speech in the application would turn off speech for the whole operating system.

Microsoft offers three different states for speech recognition in Windows. Speech recognition can be active, listening for all commands, it can be put into sleep mode, only listening for the command “start listening” and the last operation mode turns speech control off.

Figure 35: The three operating modes of speech recognition as presented in Microsoft Windows.

To connect the speech interaction in the MMSIW with the speech recognition service, the SpeechListener class is altered, using functions provided by SAPI to control the service's operation mode. This procedure requires the removal of the implemented feature of waiting for the command "listen", replacing it with the already available command by Microsoft "start listening". Unfortunately, Microsoft missed including a command to change the operating mode to sleep. Hence, "start listening" and "stop listening" are the only voice commands to manipulate the operating mode, thus requiring a custom solution to put speech recognition to sleep in Microsoft Windows.

Eric C. Brown from Microsoft released a blog entry in 2010 describing the usage of SAPI to detect if speech recognition is in sleep mode, included with Windows 7 into SAPI (Eric C. Brown 2010). The example presented by Brown reveals that SAPI has been extended by this feature with the inherited class *ISprecognizer3* using *ISpCategories*. These new classes are introduced with SAPI 5.4, leveraging the minimum requirement for the MMSIW to Windows 7. A brief look into the API documentation does not indicate the purpose of *ISprecognizer3* and *ISpCategories* (Microsoft 2016a). Without the example provided by Brown, a proper implementation of the desired behavior seems to be impossible, as also the presented examples by Microsoft do not include the access of the operating mode. Nevertheless, an examination of the blog entry by Brown leads to Code 8.

```
1.     bspct = SPCT_SLEEP;
2.           if (SUCCEEDED(hr))
3.           {
4.                 cpReco3->GetCategory(bspct, &bCategory);
5.                 cpReco3->SetActiveCategory(bCategory);
6.                 mRecognizer->SetRecoState(SPRECOSTATE::SPRST_ACTIVE_ALWAYS);
7.                 CHANGE_LISTEN_MODE = 99;
8.           }
```

Code 8: Putting speech recognition to sleep programmatically, SpeechListener.cpp.

To change the current operating mode of speech, an instance of the *ISpRecognizer3* needs to be retrieved from the created recognizer and altered by it is property *Category*. Although, the recognition state of the recognizer can only be put into *active* or *inactive* mode, the category determines if the active mode equals to the sleep mode, see Figure 36.

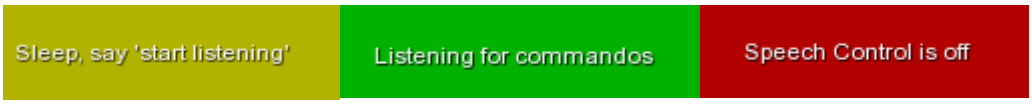


Figure 36: Speech recognition operation modes represented in the MMSIW.

By using the category parameter of SAPI, the speech recognition process can be put into sleep for the whole system, reaching the desired behavior for voice control.

4.5 Fusion Engine

The prototype from the previous work which is developed further for this thesis was implemented in a frame-based approach. Therefore, all input events are processed on a single frame iteration. A different approach would be to decouple the events from the frame loop by applying time-stamps on every registered event, thus enabling the system to process the event during the process of a different frame. Or, events could also be buffered for a period. Different approaches have been presented in the past for the integration of multimodal interfaces using fusion engines. The process of fusion input streams for multimodal interaction can contain different solutions. Most research projects in the past 30 years offered different approaches on solving the process of fusion.

In 2009 Lalanne et al. came up with a survey on fusion engines (Lalanne et al. 2009). The paper offers a closer look on different approaches to fusion engines, but besides this, it also covers the evolution of multimodal fusion concepts. For this reason, the survey also delivers an overview of some research projects on the field of multimodal fusion. In Figure 37, a timeline is used to visualize an overview on fusion strategy developments of the past 30 years. While fusion engines started off in a sequential manner, the simultaneous and synergistic approaches emerged within a decade and became popular with researchers during the

1990s. As we entered the new millennium, research shifted towards context-based models, e.g. to remove ambiguities during the fusion process.

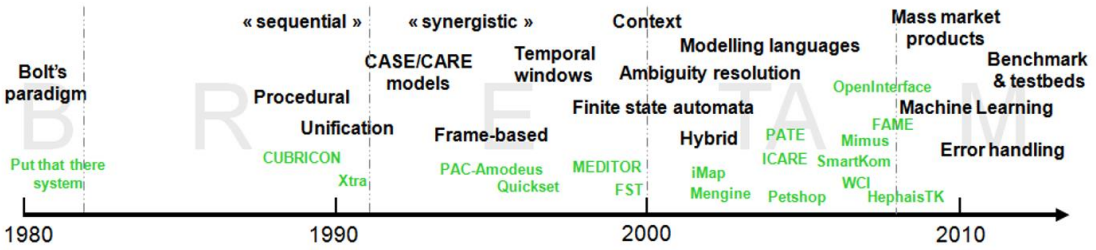


Figure 37: Overview on research projects with multimodal interaction and fusion engine concepts (Lalanne et al. 2009).

The timeline in Figure 37 uses some categories proposed by Nigay et al. in 1993, which is called the CASE-model. It uses two dimensions to split multimodal interaction based on the use of the modalities, sequential and parallel, and on type of fusion implemented, combined or independent. For the MMSIW a synergistic, frame-based approach is implemented, to fusion the input-streams.

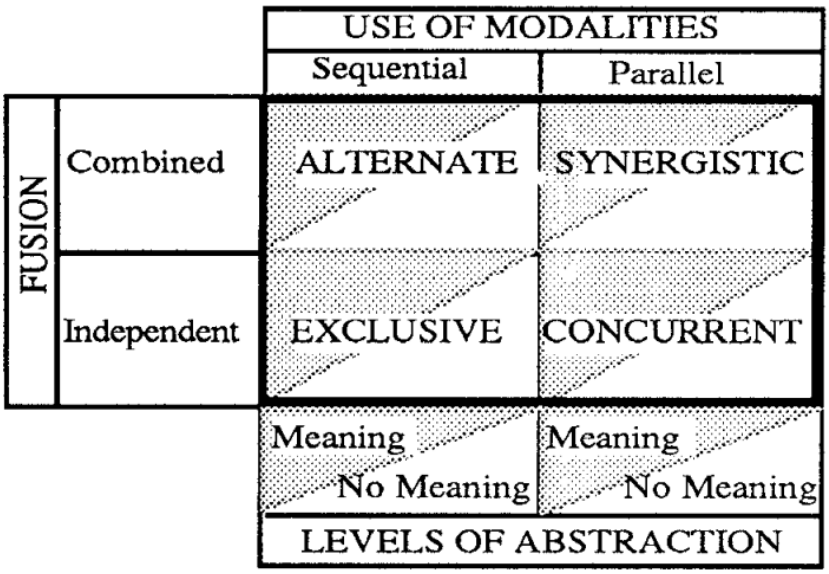


Figure 38: The CASE-Model showing different principles for the fusion of modalities (Nigay et al. 1993).

In each processed frame, the input received from the diverse listener classes are used during the fusion process. Therefore, the sequence in which the updates are processed are of importance.

In Figure 39, parts of the fusion process are visualized with a sequence diagram. The combination of input streams is mainly done with the eye tracking and speech recognition, thus, being processes at first.

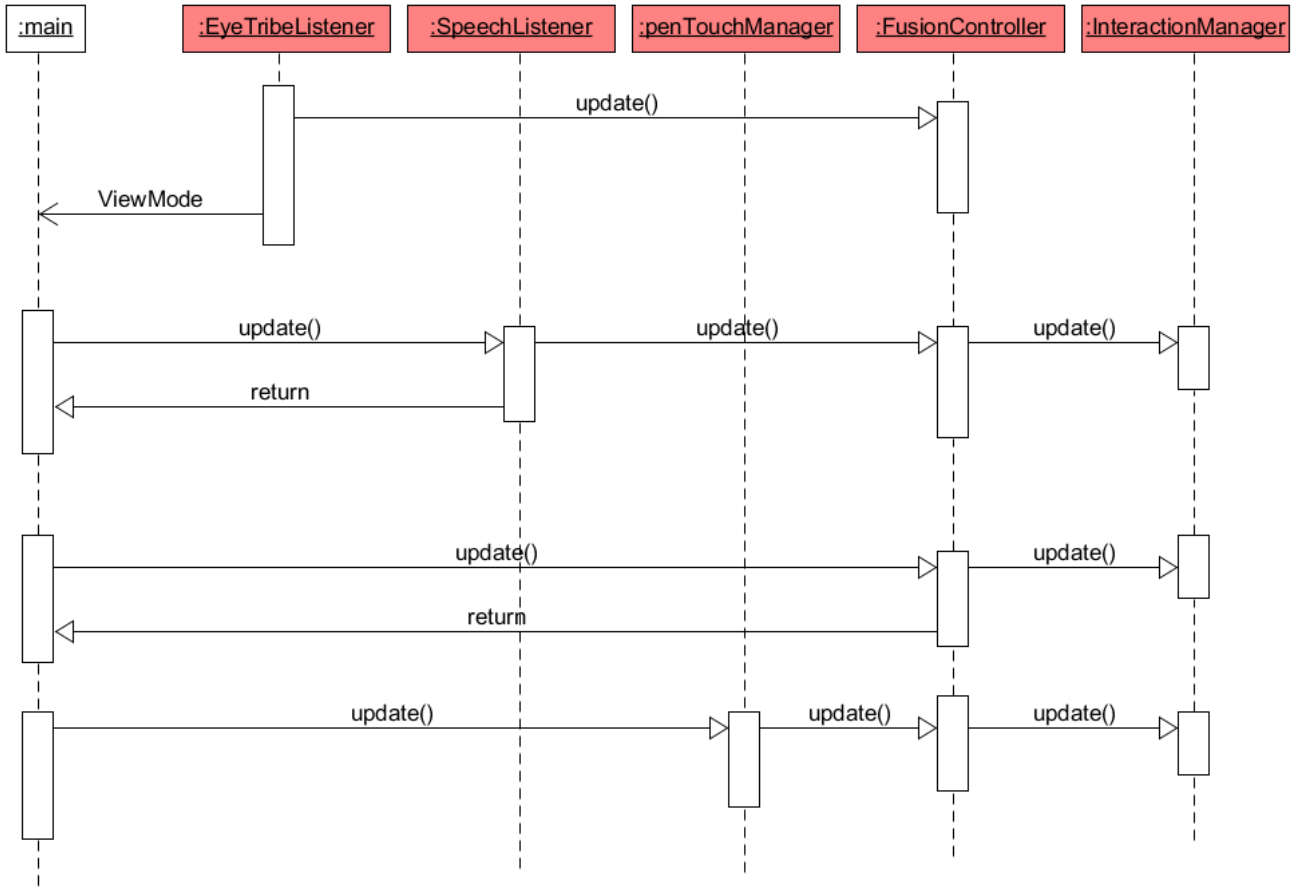


Figure 39: Sequence diagram illustration the fusion process

The control of the fusion process is completely programed in the *FusionController* Class. Although the FusionController operates the combination of the input streams, the overall interaction is encapsulated in the *InteractionManager* Class, which manipulates the *ViewMatrix* based on the processed input. Therefore, the FusionController receives updates on the user's gaze and recognized speech, evaluates the received input considering system status, e.g. the active slice, and triggers actions based on the result of the interaction incorporating controller classes like the InteractionManager, the ViewManager or the GUIMananger.

The presented process can be viewed as an altered *Observer programming pattern*, where the FusionController is the observer being informed by multiple subjects, being the listeners in this case.

IV. User Study

5 User Study

To confirm the hypotheses of this work, a user study is being conducted. This section delineates the setup of the test and concludes with the results.

5.1 Motivation

A concluding user study is conducted to confirm or reject the hypotheses which have been stated in chapter 1.3. The null hypothesis Fiedler et al. evaluated during a user study, "[…] declare that a synergistic fusion engine has more user acceptance and is easier to master and therefore causes less cognitive load for users than no combination of modalities" (Fiedler et al. 2015). Per Fielder et al., the null hypothesis, stating that multimodal interaction does not result in an increased efficiency, could only be rejected for a complex task. To find out if this result can be transferred to the multimodal interaction integrated in the MMSIW, the user study is conducted similar to the approach of Fiedler et al. (Fiedler et al. 2015) and Oviatt et al. (Oviatt et al. 2005).

5.2 Methodology and Metrics

The results of the user study are probed with Student's t-test to reject the null hypotheses. A t-test can be used to identify significances between two data sets. For the Multimodal Seismic Interpretation Workspace, the data sets are collected during the user study and consist of three metrics, the time a user requires to carry out a task, the amount of errors he commits during the execution and the amount of assistance he needs to complete a task. The values are collected per each task and result in two data sets for each task.

The t-test significance testing is a statistical hypothesis test which can be used to determine if one of three cases are present for parameter z and its desired value z_0 :

(I) z meets the desired value z_0

(II) z does not overrun the desired value z_0

(III) z does not fall below the desired value z_0,

one of these cases becomes the null hypothesis H_0 (Stingl 2009, 668 pp.). For the Multimodal Seismic Interpretation Workspace, the null hypothesis states that the time, the amount of errors and the need for assistance are equal in both data sets. It is assumed that the data sets are normally distributed by

$N(\mu: \sigma^2)$.

And the related null hypothesis is defined as follows,

$H_0: \mu = \mu_0$,

with μ as the mean value for the data set which resulted from the evaluation of the common interaction technology and μ_0 being the mean value for the data set results of the new interaction technologies.

As an estimated value for μ and μ_0, μ=$\bar{x}$ and μ_0=$\bar{x}_0$ are used, where $\bar{x}$ and $\bar{x}_0$ equate to the empirically determined mean values of the data sets. The variable for testing T is then defined as

$$T = \frac{\bar{X} - \mu_0}{S/\sqrt{n}}$$

A critical value c determines if the null hypothesis can be rejected and is calculated based on a mistake value α. Leading to the calculation of probability P with

$P(|T| > c|\mu = \mu_0) = \alpha$ Or $P(|T| \leq c|\mu = \mu_0) = 1 - \alpha$ (Stingl 2009, p. 668)

For the data sets retrieved from the user testing, the t-test is realized with Microsoft Excel. Deciding on the correct type of the t-test is very important, as the usage of the wrong t-test type may lead to incorrect results and therefore reject the null hypothesis falsely. The data sets for each task consist of ten samples. It is assumed that for complex tasks the mean values for using new interaction technologies will be significantly better. Therefore, a one-tailed two-sample t-test is carried out. The t-test can be executed assuming equal- or unequal variances. According to Charles Zaiontz, who authored an article on two sample t-testing on the web site “Real Statistics Using Excel”, a rule of thumb for the testing allows the usage of the equal variance assumption in most cases, as “[…] even if one variance is up to 4 times

the other, the equal variance assumption will give good results" (Zaiontz 2016). Thus, leading to the usage of the t-test version whenever the variances do not break the stated rule of thumb.

5.3 Test Environment

The testing environment for the first five tests are performed with the setup as seen in Figure 40. Each participant receives the same introduction to the system by watching a short tutorial video. The whole procedure is observed by an overseer, who may also be approached for assistance by the test person.

To receive a subjective opinion on the system with a questionnaire, every participant performs each task with the new interaction technologies like pen and touch, gaze etc., and with a common mouse, but for every person the order of the technologies is altered to remove the effect of customization on the test results. The testing setup is only changed according to the input devices, e.g. the participant is asked to hand over the pen and is instructed to use a mouse for further interaction or the microphone is turned off.

Figure 40: Testing setup for the user testing

Due to a sudden failure of the Wacom Cintiq QHD 27 device during the user test period, five of the tests are executed with a Wacom Intuos Pen & Touch tablet (Wacom 2016a). The touch interaction involving the GUI is more complicated with the Intuos device and the users are asked to carry out these tasks using the pen. Other actions like panning the view, zooming in and out, rotating and painting are not affected.

Ten participants, eight males and two females, attend the testing, aged with a mean value of 29.8 years.

5.4 Results

The most important tasks for the hypothesis H_1 are the complex tasks, involving multiple steps and the usage of the simultaneous fusion engine to be performed. Tasks meeting these requirements are the tasks with numbers 6, 9, 12, 13 and, with limitations, task 10. The null hypothesis cannot be rejected for most of the simple tasks in terms of time taken by the participants, which are respectively the tasks with the numbers 2, 3, 4, 5, 7, 8, 11 and 14. One exception is task 8, indicating a high significance in favor of old interaction technologies. Task 8 asks the user to select a slice as active and to remove it from the scene. Although the t-test probability for the errors does not allow the rejection of the null hypothesis for errors, the mean value for this metric indicates that errors occurred more often when the participants used the new interaction technologies. The observations, which have been made during the execution of task 8, permit the assumption that a usability problem could be the reason for the results. Therefore, a further investigation and improvement of the paradigm needs to be considered in further research.

5.4.1 Task 6 and 9

For task 6, all participants were asked to select specified attributes and to paint a circle. When new ITs are used, the users are requested to carry out the task with speech control. Task 6 is also the first of the tasks to instruct speech for the selection. Although the participants were slightly faster in achieving the desired result with the new ITs, the t-test results in Table 1 shows that a significance could not be measured for the time. Interestingly, the

probability determined for errors show a high significance for this metric. Therefore, the users made less errors by using the new ITs.

Task 6	New Interaction Tech. (Time)	Old Interaction Tech. (Time)
Mean Time	33.7	35.8
Mean (Error)	0	0.4
Variance	165.5666667	81.28888889
Observations	10	10
P (Time)	0.338772388	
P(Errors)	0.018393749	
P(Assistance)	0.336661018	

Table 1:Results for task 6: "Change the displayed attribute of the selected Slice to Attribute "Entropy" and the drawing attribute to Amplitude. Paint a circle" (selection, manipulation)".

To execute task number 9, the participants were asked to select a previously saved slice from the MapView and to preview it on the right side of the SplitView. To achieve this goal with new ITs, the users have to gaze on the left side of the SplitView and say the command "Select #", where "#" is the row number of the table entry, and then "Push right". The mean values in Table 2 for this task shows that the participants required more time to carry out the task by using the new ITs. The t-test even results in a highly significance for the data sets, indicating that the presented implementation with the new ITs fail to optimize this task.

Task 9	New Interaction Tech. (Time)	Old Interaction Tech. (Time)
Mean	14.8	11
Variance	33.51111111	8.666666667
Observations	10	10
P (Time)	0.043250351	
P(Errors)	0.177920573	
P(Assistance)	0.5	

Table 2: Results for task 9:" Select your painting in the MapView and preview it on the right side of the Split View." (selection, manipulation)".

5.4.2 Task 12 and 13

To carry out task 12, the participants must gaze on the right side of SplitView and use the speech command "Push down" to view the previewed slice in the Painting Area. As task 12

requires the simultaneous usage of speech control and eye tracking, it is a more complex task. In comparison, the user must click on the designated button using the old IT. The results in Table 3 show that the users were slightly faster when they used the old IT, but the data sets must not be considered as highly significant.

Task 12	New Interaction Tech. (Time)	Old Interaction Tech. (Time)
Mean	10.1	6.7
Variance	37.43333333	4.011111111
Observations	10	10
P (Time)	0.061656098	
P(Errors)	0.083925328	
P(Assistance)	0.083925328	

Table 3: Results for task 12 " Return to your last painting by pushing the preview on the right to the painting area." (navigation, manipulation)"

For the last complex task, the users are asked to select their first saved slice from the table on the left side of SplitView and remove it using speech in combination with the eye-tracker. The results collected from the last task using new ITs are almost identical to the results using the old ITs. A significance could not be detected. The result might be related to the task, but could also indicate customization to the new ITs.

Task 13	New Interaction Tech. (Time)	Old Interaction Tech. (Time)
Mean	6.5	6.4
Variance	2.944444444	1.822222222
Observations	10	10
P (Time)	0.443267029	
P(Assistance)	0.083925328	

5.4.3 Questionnaire

To receive a more subjective perspective from the participants, each user is asked to fill in a questionnaire after all tasks are carried out successfully.

The questionnaire also includes some questions regarding the expertise of the users, e.g. the question "How often do you use a computer?", which can be rated using values reaching from one to five. A value of one indicates that the subject almost never uses a computer and

a value of five indicates that a computer is used “almost every minute”. This sort of rating is also applied to the other questions in Figure 41.

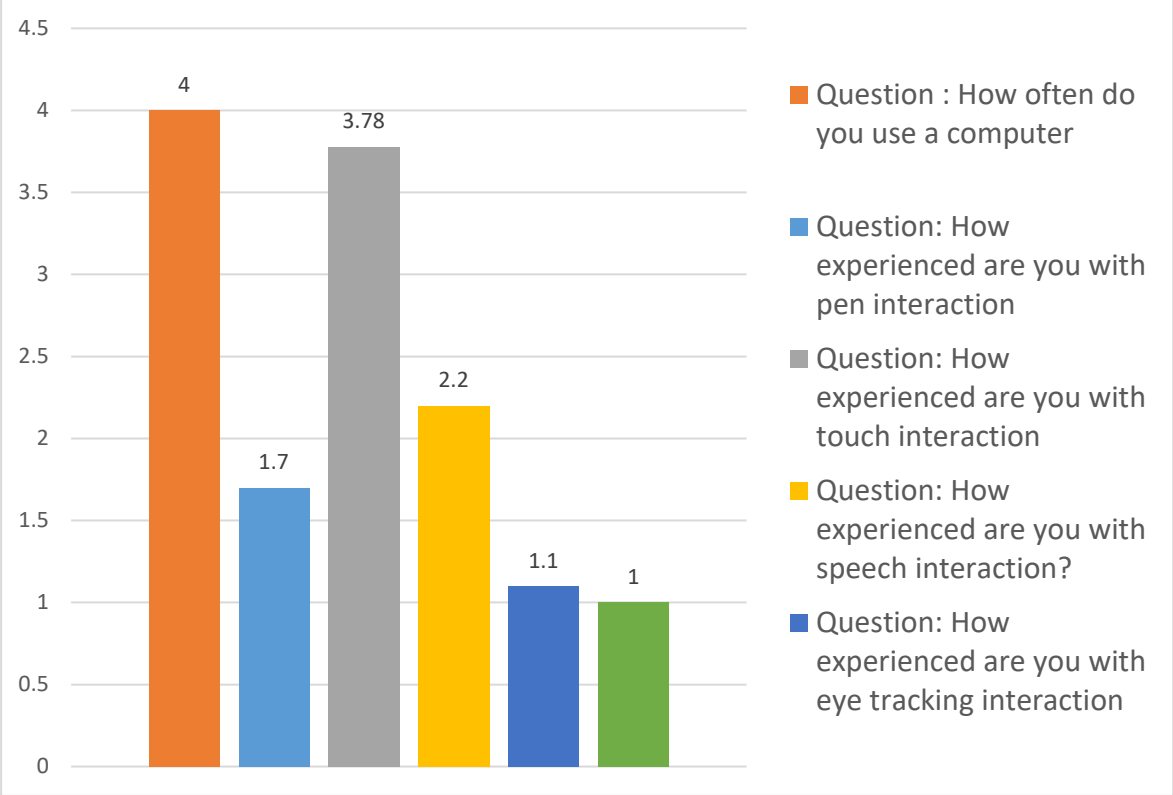


Figure 41:Introductory questions to determine the expertise of the participants.

The tasks have been categorized into navigation, manipulation and selection tasks. Therefore, questions regarding the tasks are grouped by the categories. One essential question to answer is whether the new ITs are more precise than the old ITs. All participants are asked to rate the precision on a scale from one to five, with higher values indication a more favorable rating. The ratings in Figure 43 show that the old ITs are evaluated better than the new ITs across all tasks.

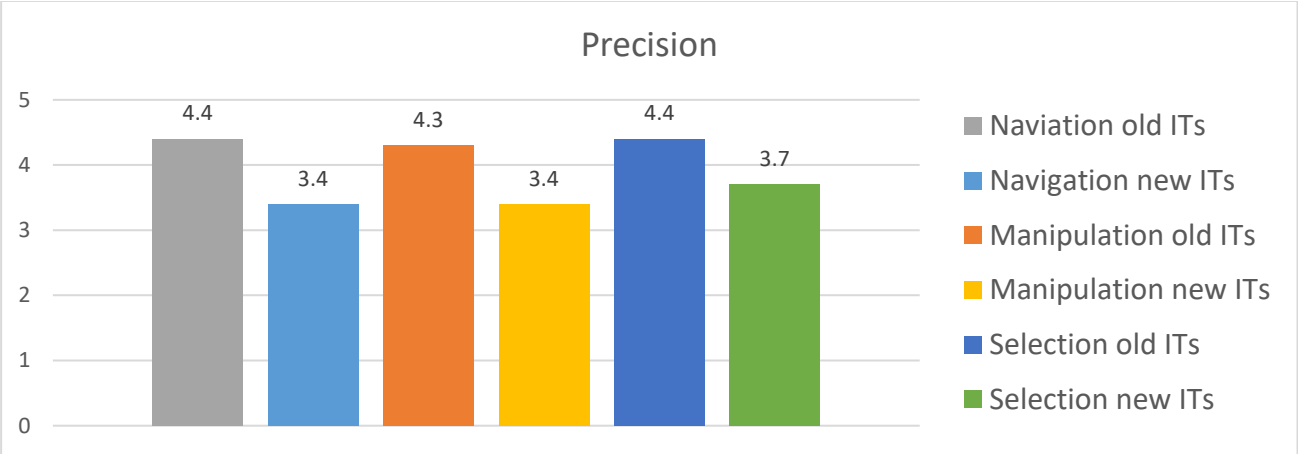


Figure 42: Ratings on how precise the execution of the tasks felt by the participants.

The conducted t-Test reveal a high significance between the rating data sets, see Table 4. Though, the results indicate a low acceptance for the implemented ITs, it is required to contextualize the determined values using the information regarding the expertise of the participants.

Precision	S(old ITs)	S(new ITs)	p
Navigation	0.51639778	0.63245553	0.03204468
Manipulation	0.48304589	0.51639778	0.00079449
Selection	0.48989795	0.48304589	0.0058036

Table 4: t-Test results on precision

Overall ratings for the implementations of the interaction technologies are positive altogether, with speech being the interaction technology receiving the least approval from the users, Figure 43.

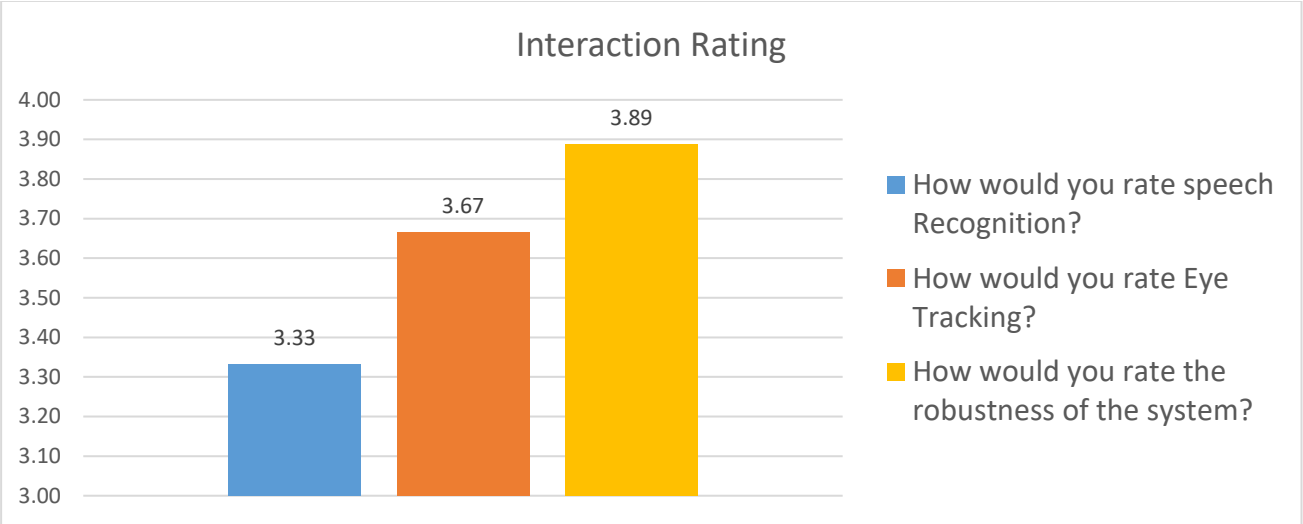


Figure 43: Interaction ratings

A reason for this result can be interpreted from the feedback users gave both verbally and written by filling in the feedback box at the end of the questionnaire. Several users had difficulties to activate speech control into listening mode via speech, when the system was sleeping and only waiting for the command “start listening”. Apart from this, most of the other commands worked properly. This issue could be related to the fact that the speech recognizer is not trained and thus adapted to the voices of the participants before the test is being conducted to decrease the testing time. Therefore, the recognition of the command “start listening” might be more successful when a user trains his unique profile over a longer period instead of using a default profile.

A similar technical difficulty is encountered for the implementation of the eye tracker. For some participants, the eye tracking does not work as expected resulting in the frequent loss of tracking. One participant complains about the eye tracker, as the device seems only to recognize his eyes when they are opened widely, which he himself relates to his Asian background. It might be possible to increase the quality of the tracking by calibrating the eye tracker individually, but for the experiments, this procedure has been skipped by using the default profile to shorten the testing time. Several participants wearing glasses encountered similar difficulties concerning the proper recognition of their gaze.

Some rare crashes could not be finally fixed before the study is conducted. Therefore, the participants are given the opportunity to rate the prototypes robustness, which also hints how influencing crashes are perceived by the users. Most of the users did not experience any

crashes during their experiments. The mean value presented in Figure 43 points to a positive result for the robustness. Nevertheless, bugs leading to crashes need to be fixed before further development and research is initiated.

A different part of the questionnaire revolves around some statements to which the participants can approve of or not. The most interesting result is the overall disapproval of speech to select different attributes and other options, see Figure 44. Based on the observations made during the study, this sort of selection technique seems not to be familiar to any of the participants. Although the result is clearly negative, a longer period of interaction using speech for selection might lead to an increased acceptance. Besides, the mentioned technical issues, e.g. the missing training for the speech profile, might also have been influential to the received output.

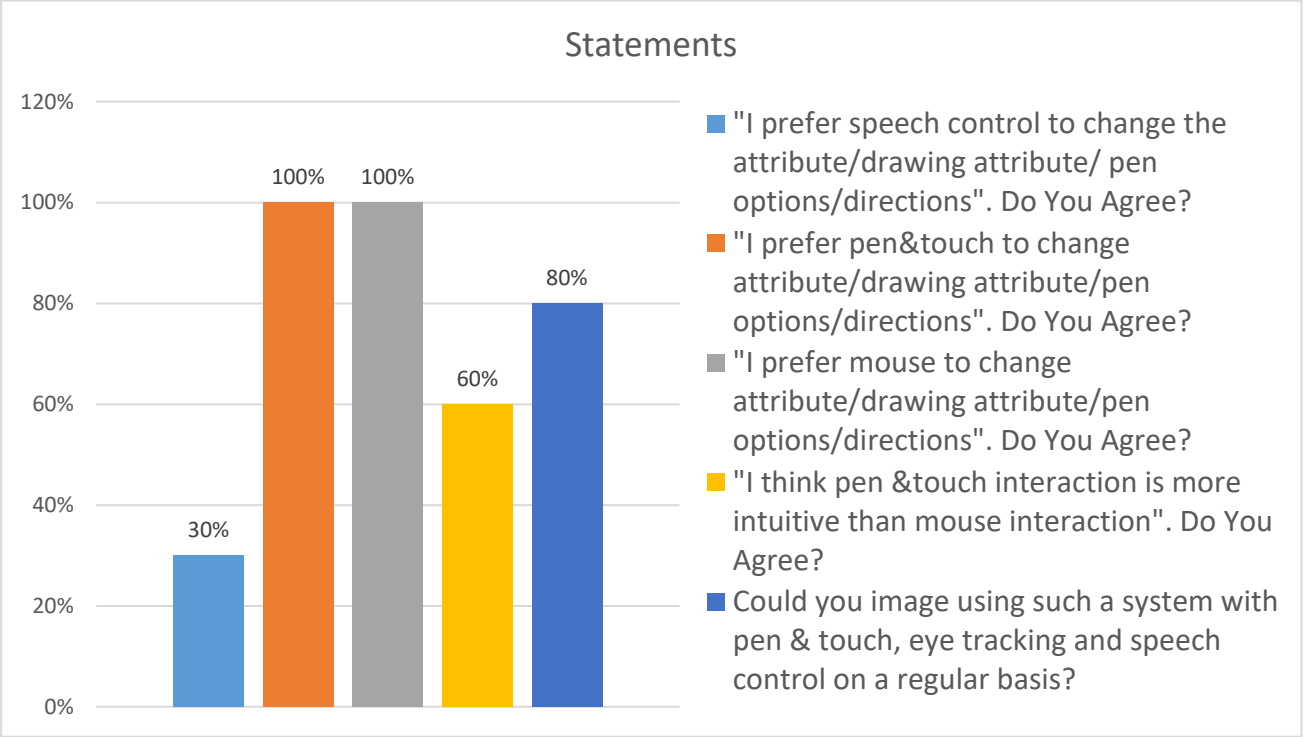


Figure 44: Results on the approval or disapproval of statements stated in the questionnaire.

Statements on more common interaction techniques, e.g. the utilization of pen and touch for selection or navigation tasks, are completely supported by the subjects. A slight majority of the participants believes pen and touch to be more intuitive in comparison to interaction using a mouse. Therefore, the acceptance for both ITs seem to be equally high for selection and navigation tasks.

V. Discussion And Conclusion

6 Discussion And Conclusion

This thesis presents the optimization of a prototype application enabling the user to interpret seismic data, incorporating pen and touch, speech and eye input. The ideas inherited by the design are described and the most important difficulties which emerged during the development of the prototype are introduced by this thesis, offering solutions to the issues. Furthermore, the results of a conducted user study are discussed by this thesis, partially negating the hypotheses of this work.

Although most findings do not result in high significances for the metrics time, error and assistance, the calculated mean values are often similar for new and old ITs. While task 9 has been executed in significantly less time by the users, significantly more errors have been made by the participants during the realization of task 6. And while the subjects required more time to carry out task 12, the time metric settled down to be almost identical for task 13, which utilized a similar IT like task 12, incorporating speech and gaze. Alongside with the technical difficulties mentioned in Chapter 5.4.3, further challenges could be observed during the testing. Although most tasks have been described as precisely as possible, most users carried out the tasks differently, e.g. the drawing of a circle in task 6. To retrieve comparable values for the time, the observer had to consider these discrepancies during the measurement of time, which cannot be balanced by the standard deviation. Therefore, the overseer tried to remove the time users spent individually, and only counted the time the users required to carry out the task. For task 6, the measurement of the time stopped as soon as the user started to paint the circle. Furthermore, some participants seemed to be insecure using the new ITs during the first tasks, but appeared to be more self-confident towards the end of the testing. This observation cannot be concluded from the collected data, highlighting the necessity of investigations over a longer period. For the Multimodal Seismic Interpretation Workspace, professional seismic interpreters could be accompanied over several weeks using the workspace. This approach would be like the conducted study by Oviatt et al., investigating the individual differences in multimodal integration (Oviatt et al. 2005),

For the simple tasks, e.g. selecting an attribute from the GUI via mouse or pen and touch, no significances could be detected at all, which supports the results described by Fiedler et al.

(Fiedler et al. 2015). The acquired results support the research of Sharon Oviatt, who released an article on myths surrounding multimodal interaction. In this case, one myth which needs to be negated when it comes to the integration of multimodal interfaces, is the assumption that the main advantage of multimodal interfaces is the enhanced efficiency. Although a speed-up of 10% was detected for pen and voice interaction in comparison to speech-only interfaces for spatial domains, this results cannot be transferred to other domains without further investigations, according to Oviatt (Sharon 1999, pp. 80–81). Nevertheless, other advantages which have been identified in the past, e.g. the flexibility multimodal interfaces offer to the user, need to be considered stronger as a future direction in research.

In conclusion, using new ITs for more complex tasks with multiple steps do not necessarily lead to increased efficiency, at least not for the tasks tested in this study. Therefore hypothesis H_1cannot be confirmed by the conducted user study.

As stated in 1.3, this research project is motivated by two hypotheses H_1 and H_2. Per H_2, new ITs should gain more acceptance by the users in comparison to the old ITs. Mainly relevant to H_2 are the results of the questionnaire. The collected data offers a divisive perspective on the prototype. While most users disapprove of the implemented voice interaction, the participants could imagine using a similar system incorporating multimodal interaction on a regular basis. Although the mouse is being favored in terms of precision, both types of ITs are rated positively by the users. Furthermore, a slight majority of the users even considers pen and touch to be more intuitive in comparison to the mouse. These results show that the flexibility achieved through further interaction with new technologies needs to be considered as a very important aspect.

Based on the results, this thesis shows that multimodal interaction requires long-term research and evaluation whenever new interaction techniques are designed and implemented, especially when it comes to complex processes like the seismic interpretation or the interaction with scientific data. Suggestions made throughout this thesis, including altered goals for the future evaluation can be taken as valuable input for future research in the field of multimodal interaction for seismic interpretation.

Bibliography

7 Publication bibliography

Ambler, Scott W. (2014): User Stories: An Agile Introduction. Agile Modeling. Available online at http://www.agilemodeling.com/artifacts/userStory.htm#InitialFormal, updated on 10/9/2014, checked on 11/16/2016.

Bar-Zeev, Avi (2007): Scenegraphs: Past, Present, and Future. Reality Prime. Available online at http://www.realityprime.com/blog/2007/06/scenegraphs-past-present-and-future/, updated on 11/17/2016, checked on 11/17/2016.

Bolt, Richard A. (1980): "Put-that-there": Voice and Gesture at the Graphics Interface. In : Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques. New York, NY, USA: ACM (SIGGRAPH '80), pp. 262–270. Available online at http://doi.acm.org/10.1145/800250.807503.

Bowman, Doug A. (2005): 3D user interfaces. Theory and practice / Doug A. Bowman … [et al.]. Harlow: Addison-Wesley.

Chung Lee, Johnny (2008): Johnny Chung Lee - Projects - Wii. Available online at http://johnnylee.net/projects/wii/, updated on 11/16/2008, checked on 12/5/2016.

Ciccarello, Liborio; Berger, Dirk (2016): Tiefenwahrnehmung durch binokulare Stereopsis. Universität Mannheim. Available online at http://irtel.uni-mannheim.de/lehre/seminararbeiten/w96/Tiefe/binoc.html, checked on 12/6/2016.

Dubey, Madhuri R.; Chhabria, S. A. (2014): Eye and Speech Fusion in Human Computer Interaction. In *International Journal* 2 (3).

Dumas, Bruno; Lalanne, Denis; Oviatt, Sharon (2009): Multimodal Interfaces: A Survey of Principles, Models and Frameworks. In Denis Lalanne, Jürg Kohlas (Eds.): Human Machine Interaction: Research Results of the MMI Program. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 3–26. Available online at http://dx.doi.org/10.1007/978-3-642-00437-7_1.

Eric C. Brown (2010): Detecting Sleep Mode in SAPI. Microsoft. Available online at https://blogs.msdn.microsoft.com/tsfaware/2010/03/22/detecting-sleep-mode-in-sapi/, checked on 12/1/2016.

Fiedler, Jannik; Rilling, Stefan; Bogen, Manfred; Herder, Jens (Eds.) (2015): Multimodal Interaction Techniques in Scientific Data Visualization - An Analytical Survey. Proceedings of the 10th International Conference on Computer Graphics Theory and Applications. Berlin, Germany, March 11-14, 2015.

Jakob Nielsen (1995): 10 Heuristics for User Interface Design. Available online at https://www.nngroup.com/articles/ten-usability-heuristics/, checked on 11/30/2016.

Jakob Nielsen (2012): Mouse vs. Fingers as Input Device. Nielsen Norman Group. Available online at https://www.nngroup.com/articles/mouse-vs-fingers-input-device/, checked on 11/12/2016.

JSON (2016): JSON. Available online at http://www.json.org/index.html, updated on 11/30/2016, checked on 12/2/2016.

Kellnhofer, Petr; Didyk, Piotr; Ritschel, Tobias; Masia, Belen; Myszkowski, Karol; Seidel, Hans-Peter (2016): Motion parallax in stereo 3D: model and applications. In *ACM Trans. Graph.* 35 (6), pp. 1–12. DOI: 10.1145/2980179.2980230.

Khronos Group (2016): OpenGL - The Industry Standard for High Performance Graphics. Available online at https://www.opengl.org/, updated on 2016, checked on 11/15/2016.

Kobialka, Daniel (2016): Siri leads Google Now in usage, study shows | FierceWireless. FierceWireless. Available online at http://www.fiercewireless.com/developer/siri-leads-google-now-usage-study-shows, updated on 12/6/2016, checked on 12/6/2016.

Koons, David B.; Sparrell, Carlton J.; Thorisson, Kristinn Rr (1993): Integrating simultaneous input from speech, gaze, and hand gestures. In *MIT Press: Menlo Park, CA*, pp. 257–276.

Lalanne, Denis; Nigay, Laurence; Palanque, Philippe; Robinson, Peter; Vanderdonckt, Jean; Ladry, Jean-François (2009): Fusion Engines for Input Multimodal Interfaces: a Survey. In : ICMI'09: Proceedings of the 11th international conference on Multimodal interfaces, November 2-6, Cambridge, MA, USA. ACM. ACM New York, NY, USA, pp. 153–160.

leankit (2016): What is Kanban? - LeanKit. leankit. Available online at https://leankit.com/learn/kanban/what-is-kanban/, checked on 12/3/2016.

Ljamas, Ramon; Reith, Ryan; Nagamine, Kathy (2016): IDC: Smartphone OS Market Share. IDC. Available online at http://www.idc.com/prodserv/smartphone-os-market-share.jsp, checked on 12/6/2016.

Metulev, Nikola (2016): Meet the Speech Platform in Windows 10. Available online at http://metulev.com/meet-the-speech-platform-in-windows-10/, updated on 5/8/2016, checked on 11/18/2016.

Microsoft (2016a): ISpRecognizer3::GetCategory (SAPI 5.4). Available online at https://msdn.microsoft.com/en-us/library/ee413267(v=vs.85).aspx, checked on 12/1/2016.

Microsoft (2016b): Microsoft Speech API (SAPI) 5.4. Microsoft. Available online at https://msdn.microsoft.com/en-us/library/ee125663(v=vs.85).aspx, checked on 11/18/2016.

Microsoft (2016c): Speech API Overview (SAPI 5.4). Microsoft. Available online at https://msdn.microsoft.com/en-us/library/ee125077(v=vs.85).aspx, checked on 11/18/2016.

NDTV (2015): Siri Gets 1 Billion Requests a Week, Nearly 1 Million Locations to Accept Apple Pay: Apple. NDTV. Available online at http://gadgets.ndtv.com/mobiles/news/siri-gets-1-billion-requests-a-week-nearly-1-million-locations-to-accept-apple-pay-apple-701411, checked on 12/6/2016.

Neal, J. G.; Thielman, C. Y.; Dobes, Z.; Haller, S. M.; Shapiro, S. C. (1989): Natural language with integrated deictic and graphic gestures. In : Proceedings of the workshop on Speech and Natural Language. Cape Cod, Massachusetts: Association for Computational Linguistics, pp. 410–423.

Nielsen, Jakob (2008): Top 10 Application-Design Mistakes. Nielsen Norman Group. Available online at https://www.nngroup.com/articles/top-10-application-design-mistakes/, checked on 11/16/2016.

Nigay, Laurence; Joëlle; Coutaz, lle (1993): 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. Amsterdam, The Netherlands: ACM, pp. 172–178.

Nintendo (2016): Nintendo 3DS. Available online at http://www.nintendo.com/3ds, updated on 12/6/2016, checked on 12/6/2016.

NVIDIA (2016): Quadro Quad Buffered Professional Stereo Technology. Available online at http://www.nvidia.de/object/quadro_stereo_technology_uk.html, updated on 10/28/2016, checked on 11/15/2016.

Ömer Genç (2013): Novel Pen & Touch Based Interaction for Seismic Interpretation. Master's Thesis. University of Applied Sciences Düsseldorf, Düsseldorf, Germany. Department of Media. Available online at http://www.vrgeo.org/index.php?eID=tx_nawsecuredl&u=0&g=0&t=1479156796&hash=7585b60fe4e9ed2274b7c3cf60d9f0b268ee6c6c&file=fileadmin/VRGeo/Bilder/VRGeo_Papers/PenTouch_final.pdf, checked on 11/13/2016.

OpenSceneGraph: OpenSceneGraph. OpenSceneGraph. Available online at http://www.openscenegraph.org/, checked on 11/13/2016.

OpenSceneGraph (2007a): Support/Tutorials/Intersections – osg. OpenSceneGraph. Available online at http://trac.openscenegraph.org/projects/osg//wiki/Support/Tutorials/Intersections, checked on 11/17/2016.

OpenSceneGraph (2007b): Support/Tutorials/NodeMaskDemo – osg. OpenSceneGraph. Available online at http://trac.openscenegraph.org/projects/osg//wiki/Support/Tutorials/NodeMaskDemo, checked on 11/17/2016.

OpenSceneGraph (2009): OpenSceneGraph Forum :: View topic - osg::Node::NodeMask documentation. OpenSceneGraph. Available online at http://forum.openscenegraph.org/viewtopic.php?t=3236, checked on 11/17/2016.

OpenSceneGraph (2010): /OpenSceneGraph/trunk/examples/osgwidgetlabel/osgwidgetlabel.cpp – osg. OpenSceneGraph. Available online at http://trac.openscenegraph.org/projects/osg/browser/OpenSceneGraph/trunk/examples/osgwidgetlabel/osgwidgetlabel.cpp, checked on 11/17/2016.

OpenSceneGraph (2016): OpenSceneGraph: osg::Image Class Reference. Available online at http://trac.openscenegraph.org/documentation/OpenSceneGraphReferenceDocs/a00382.html, updated on 4/26/2016, checked on 12/7/2016.

Oviatt, Sharon; Lunsford, Rebecca; Coulston, Rachel (2005): Individual differences in multimodal integration patterns: what are they and why do they exist? In : Proceedings of

the SIGCHI Conference on Human Factors in Computing Systems. Portland, Oregon, USA: ACM, pp. 241–249.

Oxford (2016): Oxford Living Dictionaries. Available online at https://en.oxforddictionaries.com/, checked on 12/6/2016.

Qt (2016): The future is written with Qt: Cross-platform software development for embedded & desktop. Qt. Available online at https://www.qt.io/, updated on 11/17/2016, checked on 11/17/2016.

Sauter, Marc (2016): Eye Tracker 4C: Tobiis Eye-Tracking-Leiste erkennt auch Kopfbewegungen - Golem.de. Golem Media GmbH. Berlin, Germany. Available online at http://www.golem.de/news/eye-tracker-4c-tobiis-eye-tracking-leiste-erkennt-auch-kopfbewegungen-1610-123973.html, checked on 12/6/2016.

Schlumberger (1998a): Oilfield Glossary - acquisition. Available online at http://www.glossary.oilfield.slb.com/Terms/a/acquisition.aspx, checked on 11/16/2016.

Schlumberger (1998b): Oilfield Glossary - attribute. Available online at http://www.glossary.oilfield.slb.com/Terms/a/attribute.aspx, updated on 2016, checked on 3/11/2016.

Schlumberger (1998c): Oilfield Glossary - geophone. Available online at http://www.glossary.oilfield.slb.com/Terms/g/geophone.aspx, updated on 2016, checked on 3/11/2016.

Schlumberger (1998d): Oilfield Glossary - reflection. Available online at http://www.glossary.oilfield.slb.com/Terms/r/reflection.aspx, updated on 2016, checked on 3/11/2016.

Schlumberger (1998e): Oilfield Glossary -crossline. Available online at http://www.glossary.oilfield.slb.com/Terms/c/crossline.aspx, checked on 11/16/2016.

Schulz, Christian; Sonntag, Daniel; Weber, Markus; Toyama, Takumi (2013): Multimodal interaction strategies in a multi-device environment around natural speech. In : Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion. Santa Monica, California, USA: ACM, pp. 97–98.

SEG (2016): Society of Exploration Geophysicists. Available online at http://seg.org/, checked on 12/1/2016.

Sharon, Oviatt (1999): Ten myths of multimodal interaction. In *Commun. ACM* 42 (11), pp. 74–81. DOI: 10.1145/319382.319398.

Stingl, Peter (2009): Mathematik für Fachhochschulen. Technik und Informatik. 8., aktualisierte Aufl. München: Hanser.

Streeter, Tyler (2016): Vode: Open Source Speech Interaction. Available online at http://voce.sourceforge.net/, updated on 4/29/2015, checked on 12/6/2016.

The Eye Tribe (2016): The EyeTribe. Available online at https://theeyetribe.com/, checked on 11/15/2016.

Tobii (2016): Tobii.com - Tobii is the world leader in eye tracking. Available online at http://www.tobii.com/, updated on 4/27/2015, checked on 12/6/2016.

Tridelity (2016): TRIDELITY 3D WITHOUT GLASSES. Available online at http://www.tridelity.com/, checked on 12/6/2016.

Turk, Matthew (2014): Multimodal interaction: A review. In *Pattern Recognition Letters* 36, pp. 189–195.

UMLet (2016): UMLet - Free UML Tool for Fast UML Diagrams. UMLet. Available online at http://www.umlet.com/, updated on 4/14/2016, checked on 11/17/2016.

Unknown (2010): Sample Source Code for Speech Developers Part 2 and SAPI 5.4. Microsoft. Available online at https://blogs.msdn.microsoft.com/speech/2010/04/30/sample-source-code-for-speech-developers-part-2-and-sapi-5-4/, checked on 11/18/2016.

VRGeo (2016): VRGeo: Innovation for Oil and Gas Exploration. VRGeo. Available online at http://www.vrgeo.org, checked on 12/6/2016.

Wacom (2015): Wacom Cintiq 27 QHD Touch. Available online at http://www.wacom.com/en-us/products/pen-displays/cintiq-27-qhd-touch, checked on 11/12/2016.

Wacom (2016a): Intuos Art. Wacom. Available online at http://www.wacom.com/de-ch/products/intuos-art, checked on 11/17/2016.

Wacom (2016b): Wacom Developer Program. Available online at https://developer.wacom.com/, checked on 11/13/2016.

Zaiontz, Charles (2016): Two Sample t Test: unequal variances | Real Statistics Using Excel. Real Statistics Using Excel. Available online at http://www.real-statistics.com/students-t-distribution/two-sample-t-test-uequal-variances/, checked on 11/20/2016.

Appendix

A.1. Appendix

User stories

Nr.	User Story
1	As a user I want to rotate the painting area view so that i can paint more comfortable.
2	As a user I want to view an overview on the overview display so that i can quickly figure where to navigate.
4	As a user I want to use a voice command to switch the current view direction (Crossline, Inline, Timeline), so that i don't need to use a different interaction technology.
5	As a user I want to have my paintings linked to the position of the slice, so that I can have multiple paintings on a slice.
6	As a user I want to have multiple slices visualized for interaction purposes.
7	As a user I want to use speech to change the attribute I am painting with.
8	As a user I want to use speech to change the attribute of the currently as active selected slice.
9	As a user I want to select a slice as active.
10	As a user I want to be able to activate an eraser function to erase my painting.
11	As a user I want my paintings to be saved when the attribute of the slice is changed.
12	As a user I want to save my painting.
13	As a user I want to remove a slice from the scene.
14	As a user I want to create an unedited slice in a direction of my choice (Crossline, Inline, Timeline).
15	As a user I want to save my painting using speech.
16	As a user I want to remove a slice from the scene using speech.
17	As a user I want to move the slice selected as active in it's direction (Crossline, Inline, Timeline).
18	As a user I want to pan the painting area view, so that I can paint more comfortable.
19	As a user I want to be notified about currently executed actions.
20	As a user I want to view my paintings listed in the overview.
21	As a user I want to use my gaze on the overview combined with speech so that i can move my selection to different area without the need of using mouse or keyboard.
22	As a user I want to use my gaze to transfer touch interactions to the overview display.

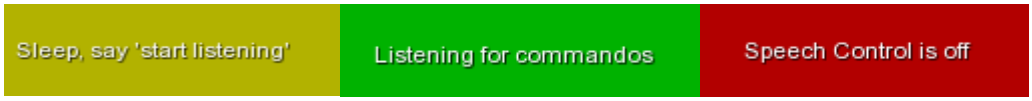
Tasks Multimodal / new Interaction Technology

General Instructions:

When you are asked to use the Graphical User Interface: Either touch or click with the pen on the shown icon to carry out the requested task.

When you are asked to use Speech Control:

Speech Control has three modes, “Sleep”, “Listen” and “Off”.



When Speech Control is off, use the microphone Icon to activate Speech Control.

When Speech Control is in sleep mode, say “start listening” or push the microphone icon the activate speech control.

When Speech Control is listening mode, say sleep/pause/hold to put speech to sleep mode or stop listening to turn speech control off.

FAQ

When is a Slice selected as “active”?

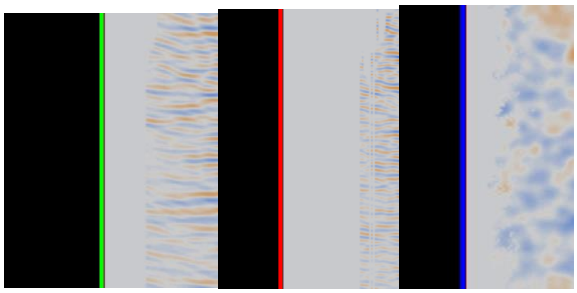
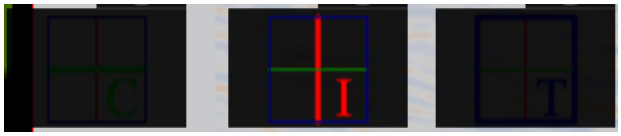
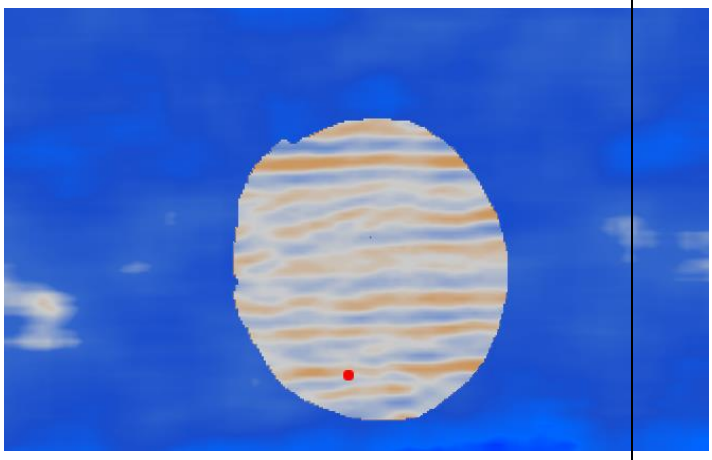
- When you click on a Slice with the Interactive Pen, this Slice becomes highlighted, which indicates, that the slice has been set as active. When you start painting on a Slice, the Slice becomes “active” automatically.

What is the SplitView, the Painting Area, the MapView etc.?

- These are names for the different sections of the application. There are also small labels in the areas.

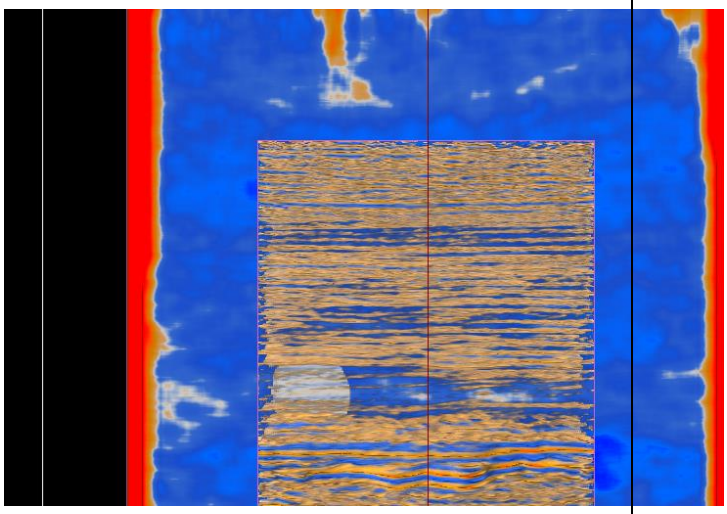
	Task (Type)	Interaction technology which should be used when more than one option is available	Hints/Description/ How to
1.	*“Adjust the displayed scene by dragging and*		Use three fingers to pan the view. Pinch-Zoom with two fingers to zoom using touch. Rotate

	zooming the scene to achieve a comfortable position." (navigation)		with two fingers circling one finger around the other to rotate the view.
2.	*"Change the painter attribute to "Entropy" and paint a square" (selection, manipulation)*	Please use Pen&Touch	Use the Graphical User Interface. 
3.	*"Save the painted Slice." (manipulation)*	Please use Pen&Touch	Use the Graphical User Interface. 
4.	*"Select a Slice as active and change the selected Slice's*	Please use Pen&Touch	Select a Slice as active using the pen. The edges of a slice are highlighted when it is selected as

	position." (selection, manipulation)		active! To move a Slice, drag the circle on the green slider, which is on the left side. 
5.	*"Change the view direction to Inline." (navigation)*	Please use Pen&Touch	Use the Graphical User Interface. It's the Button with the I (the middle one), look at the left-bottom of the Display. 
6.	*"Change the displayed attribute of the selected Slice to Attribute "Entropy" and the drawing attribute to Amplitude. Paint a circle" (selection, manipulation)*	Please use Speech Control	Just say "Attribute 2" or "Attribute Entropy" after activating speech control. To change the drawing attribute, use the speech commando "Paint 1" or "Paint Amplitude". 
7.	*"Save the painted Slice." (manipulation)*	Please use Speech Control	Use the speech commando "Slice save" while a Slice is selected as active.

8.	*"Remove the selected Slice." (manipulation)*	Please use Speech Control	Use the speech commando "Slice remove" while a Slice is selected as active.
9.	*"Select your painting in the MapView and preview it on the right side of the Split View." (selection, manipulation)*	Please use Speech Control in combination with the Eye Tracker.	Just say "Select #" (# is the number of the row) to select a row while looking on the left side of the SplitView and use the speech commando "push right" to view your selection on the right side of the SplitView. 
10	*"Pan, zoom and/or rotate the previewed Slice in the SplitView".(Navigation, Manipulation)*	Please use the Eye Tracker in combination with touch interaction in the Painting Area.	*Look on the right side of the SplitView to activate touch control for this area and try to manipulate the view with touch interaction in the painting area. You have to keep looking on the right side while touching.* 

11	*"Change the view direction to In-line." (navigation)*	Please use Speech Control.	Use the speech commando "View In(line)" after activating Speech Control. (If Inline is already selected, change to Cross or Timeline) 
12	*"Return to your last painting by pushing the pre-view on the right to the painting area." (navigation, ma-nipulation)*	Please use Speech Control in combi-nation with the Eye Tracker.	*Look on the right side of the SplitView and say "push down".*
13	*"Select and remove your first painting in the MapView." (selection, manipu-lation)*		Just say "Select #" to select a row while looking on the left side of the SplitView and use the speech commando "remove selection" to re-move the saved slice. 

14	*Change the mode to Direct Volume Rendering*	Please use Speech Control.	*Use the commando "Direct Volume" or "Split View"* 

Tasks Unimodal / common Interaction Technology

General Instructions:

Use the Graphical User Interface: Click with the left mouse button on the shown icon to carry out the requested task.

FAQ

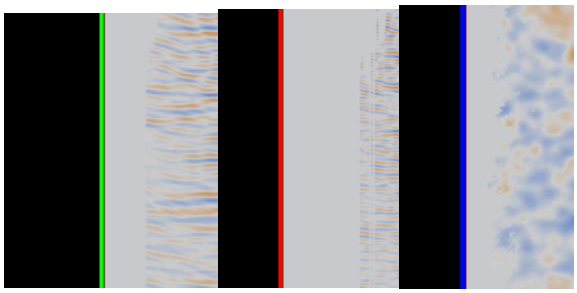
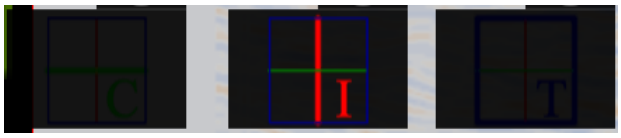
When is, a Slice selected as “active”?

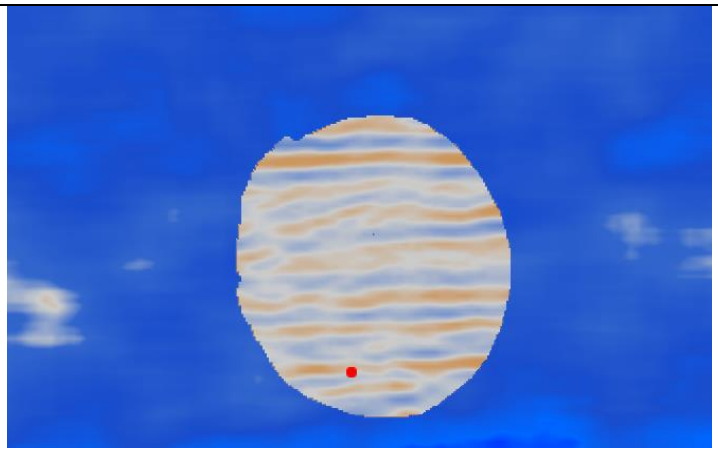
- When you click on a Slice with the Interactive Pen, this Slice becomes highlighted, which indicates, that the slice has been set as active. When you start painting on a Slice, the Slice becomes “active” automatically.

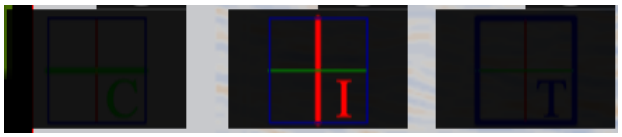
What is the SplitView, the Painting Area, the MapView etc.?

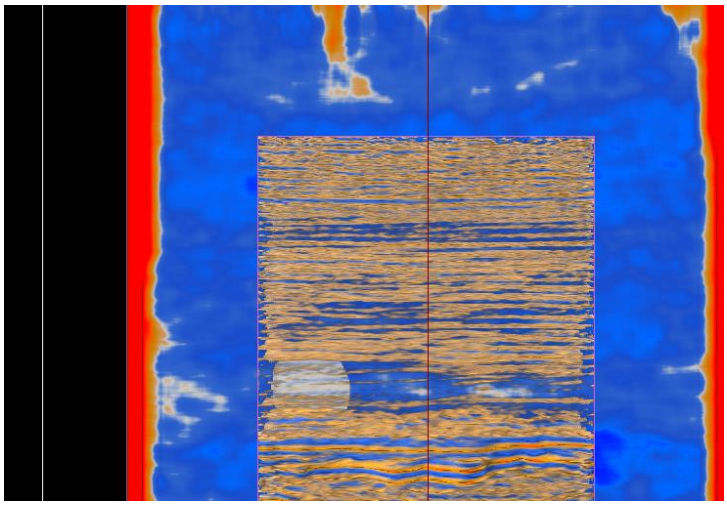
- These are names for the different sections of the application. There are also small labels in the areas. Here is a Picture of these areas.

	Task (Type)	Hints/Description/ How to
15.	*“Adjust the displayed scene by dragging and zooming the scene to achieve a comfortable position.” (navigation)*	You can zoom in by clicking the right mouse button and moving the mouse around. Use the left mouse button to pan the view and the middle mouse button to paint.
16.	*“Change the painter attribute to “Entropy” and paint a square” (selection, manipulation)*	Use the Graphical User Interface. 

17.	*"Save the painted Slice." (manipulation)*	Use the Graphical User Interface. 
18.	*"Select a Slice as active and change the selected Slice's position." (selection, manipulation)*	Click on a Slice with left mouse button. The edges of a slice are highlighted when it is selected as active! To move a Slice, drag the circle on the green slider, which is on the left side. 
19.	*"Change the view direction to Inline." (navigation)*	Use the Graphical User Interface. It's the Button with the I (the middle one), look at the left-bottom of the Display. 
20.	*"Change the displayed attribute of the selected Slice to Attribute "Entropy" and the drawing attribute to Amplitude. Paint a circle" (selection, manipulation)*	Use the menu "Attributes" to select a different attribute for the active Slice. Use the menu "Painter" to select a different attribute to paint with.

		
21.	*"Save the painted Slice." (manipulation)*	Use the Graphical User Interface. 
22.	*"Remove the selected Slice." (manipulation)*	Use the trash bin icon in the painting area.
23.	*"Select your painting in the MapView and preview it on the right side of the Split View." (selection, manipulation)*	*Select a row by clicking on it with the left mouse button. Use the designated arrow icon aiming to the right side of the SplitView to push your selection right.* 
24.	*"Pan, zoom and/or rotate the previewed Slice in the SplitView".(Navigation, Manipulation)*	You can zoom in on the right side of the SplitView by clicking the right mouse button and moving the mouse around. Use the left mouse button to pan the view.

		
25.	*"Change the view direction to Inline." (navigation)*	Use the Graphical User Interface. It's the Button with the I (the middle one), look at the left-bottom of the Display. 
26.	*"Return to your last painting by pushing the preview on the right to the painting area." (navigation, manipulation)*	*Use the designated arrow icon on the right side of the SplitView.*
27.	*"Select and remove your first painting in the MapView." (selection, manipulation)*	*Select a row by clicking on it with the left mouse button. Use the designated "Trash" icon to remove your selection.*

		
28.	*Change the mode to Direct Volume Rendering*	*Use the designated Icon in the Painting Area.* 

Questionnaire User Study – Multimodal Interaction with the Multimodal Seismic Interpretation Workspace

This user study aims to determine if natural interaction technologies may improve seismic interpretation in terms of intuitiveness and efficiency. As a participant, a video will be shown to you, briefly explaining the interaction paradigms available on the system. After that, please carry out the listed tasks. After you finished every task with new interaction paradigms and keyboard and mouse, please complete the survey by answering the questions at the bottom.

You will be supervised during the study and may ask for help. The time it takes for you to complete a task, the amount of errors and the how often you ask for help will be the metrics to determine the efficiency of the different interaction paradigms. Thank you for participation!

Participant started with which technology? _____

Participant Information

Gender:

Age:

How often do you use a computer?

☐ ☐ ☐ ☐ ☐

Never/ rarely Almost every minute

How experienced are you with pen interaction?

☐ ☐ ☐ ☐ ☐

I'm a rookie I'm an expert

How experienced are you with touch interaction?

☐ ☐ ☐ ☐ ☐

I'm a rookie I'm an expert

How experienced are you with speech interaction?

☐ ☐ ☐ ☐ ☐

I'm a rookie I'm an expert

How experienced are you with eye tracking interaction?

☐ ☐ ☐ ☐ ☐

I'm a rookie I'm an expert

How experienced are you in seismic Interpretation?

☐ ☐ ☐ ☐ ☐

I'm a rookie I'm an expert

How much fun was it to use the system?

With keyboard and mouse:

☐ ☐ ☐ ☐ ☐

very little fun very much fun

With pen&touch, speech control and eye tracking:

☐ ☐ ☐ ☐ ☐

Very little fun very much fun

How precise would you evaluate the navigation tasks (panning, zooming, moving a slice etc.)?

With keyboard and mouse:

☐ ☐ ☐ ☐ ☐

Not very precise very precise

With pen&touch, speech control and eye tracking:

☐ ☐ ☐ ☐ ☐

Not very precise very precise

How precise would you evaluate the manipulation tasks (painting, erasing, pushing slices from on area to another etc)?

With Keyboard and Mouse:

☐ ☐ ☐ ☐ ☐

Not very precise very precise

With pen&touch, speech control and eye tracking:

☐ ☐ ☐ ☐ ☐

Not very precise very precise

How precise would you evaluate the selection tasks (selecting a slice as active, switching directions, switching attributes)?

With keyboard and mouse:

☐ ☐ ☐ ☐ ☐

Not very precise very precise

With pen&touch, speech control and eye tracking:

☐ ☐ ☐ ☐ ☐

Not very precise very precise

“I prefer speech control to change the attribute/drawing attribute/pen options/directions”. Do you agree?

☐ ☐

Yes No

“I prefer pen&touch to change the attribute/drawing attribute/pen options/directions”. Do you agree?

☐ ☐

Yes No

“I prefer mouse&keyboard to change the attribute/drawing attribute/pen options/directions”. Do you agree?

☐ ☐

Yes No

“I think pen&touch interaction is more intuitive than keyboard&mouse interaction”. Do you agree?

☐ ☐

Yes No

How would you rate the speech recognition?

☐ ☐ ☐ ☐ ☐

Not very good very good

How would you rate the eye tracking?

☐ ☐ ☐ ☐ ☐

Not very good very good

How would you rate the robustness of the system?

☐ ☐ ☐ ☐ ☐

Not very good very good

“Could you imagine using such a system with pen&touch, eye tracking and speech control on a regular basis?

☐ ☐

Yes No

Feedback Box:

Available Speech Commandos (the input in the clamps are optional commands)

	Task	Commando
1	Set Painter to Amplitude or 1	Say “[Please] Paint Amplitude” or “Paint 1”
2	Set Painter to Entropy or 2	Say “[Please] Paint Entropy” or “Paint 2”
3	Set Painter to Filtered or 3	Say “[Please] Paint Filtered” or “Paint 3”
4	Set Painter to Frequency or 4	Say “[Please] Paint Frequency” or “Paint 4”
5	Set Painter to GLCM Entropy or 5	Say “[Please] Paint grey” or “Paint 5”
6	Set Painter to Instantaneous or 6	Say “[Please] Paint Instantaneous” or “Paint 6”
7	Set Painter to Variance or 7	Say “[Please] Paint Variance” or “Paint 7”
8	Set Painter to Velocity or 8	Say “[Please] Paint Velocity” or “Paint 8”
9	Set Attribute to Amplitude or 1	Say “[Switch/Select] Attribute Amplitude” or “Paint 1”
10	Set Attribute to Entropy or 2	Say “[Switch/Select] Attribute Entropy” or “Paint 2”
11	Set Attribute to Filtered or 3	Say “[Switch/Select] Attribute Filtered” or “Paint 3”
12	Set Attribute to Frequency or 4	Say “[Switch/Select] Attribute Frequency” or “Paint 4”
13	Set Attribute to GLCM Entropy or 5	Say “[Switch/Select] Attribute grey” or “Paint 5”
14	Set Attribute to Instantaneous or 6	Say “[Switch/Select] Attribute Instantaneous” or “Paint 6”
15	Set Attribute to Variance or 7	Say “[Switch/Select] Attribute Variance” or “Paint 7”
16	Set Attribute to Velocity or 8	Say “[Switch/Select] Attribute Velocity” or “Paint 8”
17	Increase the thickness of the drawing line	Say “Pen thicker”
18	Decrease the thickness of the drawing line	Say “Pen thinner”
19	Activate the eraser	Say “Eraser on”
20	Deactivate the eraser	Say “Eraser off”
21	Change current direction to Crossline	Say “View Cross[Line]” or “Direction Cross[Line]”

22	Change current direction to Inline	Say “View In[Line]” or “Direction In[Line]”
23	Change current direction to Timeline	Say “View Time[Line]” or “Direction Time[Line]”
24	Create a Crossline Slice	Say “Create Cross[Line]” or “Set Cross[Line]”
25	Create a Inline Slice	Say “Create In[Line]” or “Set In[Line]”
26	Create a Timeline Slice	Say “Create Time[Line]” or “Set Time[Line]”
27	Remove the selected Slice	Say “Slice remove”
28	Save the selected Slice	Say “Slice save”
29	Select a row in the MapView	Say “Select #” # needs to be number between zero and fifteen.
30	Remove MapView selection	Say “Remove (selected/choosen/marked)”
31	View the MapView selection or the preview in the right side of the SplitView in the Painter Area	Say “Push down”
32	View the MapView selection in the right side of the SplitView.	Say “Push right”
33	Change the mode of the OverView from SplitView to Direct Volume or vice versa	Say “(Mode/Direct/Split) (change/direct/volume/view)”
34	Put Speech Control to sleep	Say “Sleep/pause/hold”
35	Turn Speech Control on	Say “start listening”
36	Turn Speech Control off	Say “stop listening”