
MULTIMODAL INTERACTION TECHNIQUES FOR SCIENTIFIC DATA VISUALIZATION

BACHELOR OF ENGINEERING THESIS

BY JANNIK FIEDLER, 22.09.2014



SUPERVISION:

FH D: PROF. DR. ENG. / UNIV. OF TSUKUBA JENS HERDER

IAIS: DIPL.-INFORM. STEFAN RILLING

UNIVERSITY OF APPLIED SCIENCES DÜSSELDORF

DEPARTMENT OF MEDIA

STATEMENT OF ORIGINALITY

I hereby declare that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices.

Jannik Fiedler

ACKNOWLEDGMENTS

First, I would like to thank Prof. Jens Herder very much for agreeing to be the supervisor of this thesis, for assisting me in completing it, providing me with useful related information on it, and, for all the things I have learned from him during my days as a bachelor student at the University of Applied Sciences Düsseldorf.

Second, I would like to thank to my supervisor and our project manager at Fraunhofer IAIS, Stefan Rilling and Dr. Manfred Bogen, very much for their constant assistance, invaluable expertise, helping me in finding this incredibly interesting topic and all the things I have learned from them during my time at the Fraunhofer Institute IAIS. I would also like to thank the students with whom I worked alongside, Phillip Ladwig, Delger Lhamsuren, Ömer Genc and Dennis Dzendo for the great atmosphere during working hours and their ideas and input on my topics. As for the remaining and former people and employees in the "Adaptive Reflective Teams" (ART) department of the institute, I would like to state that I never enjoyed a working environment quite as much as I did while being a part of ART.

Furthermore, I would like to thank all delegates and members of the VRGeo consortium for sharing their expertise with me.

Many thanks to all participants of the user study as well, the input and feedback I received was very important and valuable for my thesis.

At last, I would like to thank my dear family, my beloved girlfriend and my irreplaceable friends for always being there for me.

ABSTRACT

The idea of an ideal interface in visions such as Sutherland’s “*The Ultimate Display*” [58] is always described similarly: The interface is able to understand every intend of a user, be it speech, pointing, head shaking or even a simple glance. It further is capable of giving visual, audible and haptic feedback to a user. The ideal interface is thus a multimodal user interface – an interface, which can use a variety of different human senses (modalities) for communication at once. The main advantages of these systems include the exploit of the full potential of a humans perceptual capabilities and the intuitiveness and ease of use that arise from using natural interaction – interaction that feels *natural*, that a user already knows from other areas and therefore does not have to newly master. Seeing as humans usually interact with each other in a multimodal fashion (gestures, speech, eye gaze etc.), it is evident that humans are used to simultaneous usage of multiple modalities and it therefore does not harm the cognitive load when used in an interface. The combination can further create more robust systems as the system only reacts to multiple modalities at once – false positive recognitions are reduced to a minimum in contrary to a unimodal interaction, where a single unintentional gesture could already be falsely interpreted as an intended. A possible field of application for such interfaces is scientific data visualization – visualization of mostly large datasets acquired by scientists through simulations, calculations or recordings. The data visualized in such applications is often massive in size and complexity, making proper visualization and interaction techniques a necessity. Since the data is often present in a three-dimensional form and thus require the use of complex interaction techniques for navigation around and handling of the data, the use of a multimodal interface seems like the perfect fit. In this work, the focus lies on multimodal interaction techniques for these scientific data visualization tasks. In particular, this work aims at evaluating if a synergistic fusion engine method is more effective and intuitive than other methods or none at all. After a complete overview of theoretical and practical approaches for the two topics, multimodal interaction and scientific data visualization, a fully expandable system which gives the user control over large three-dimensional datasets by the use of multimodal interaction techniques is described. The system is then evaluated with two different methods of combining multiple modalities modalities.

ZUSAMMENFASSUNG

Visionen wie zum Beispiel Sutherland's Vision in "*The Ultimate Display*" [58] beschreiben die Idee einer idealen Benutzerschnittstelle stets sehr ähnlich: Die Schnittstelle kann auf der einen Seite alles, was ein Nutzer macht, verstehen (Gesten, Sprache, Kopfbewegungen und sogar kleinste Augenbewegungen), während sie auf der anderen Seite ebenfalls mittels haptischen, visuellen und akustischen Feedback zum Benutzer zurück kommuniziert. Die Schnittstelle, die in diesem Szenario beschrieben ist, ist somit eine multimodale Benutzerschnittstelle – eine Schnittstelle, die viele verschiedenen Sinne für Kommunikation zum und vom Benutzer nutzt. Zu den Vorteilen einer solchen Schnittstelle gehören das volle Ausschöpfen der Wahrnehmungsfähigkeit des Benutzers und die intuitive und einfach zu lernende Art und Weise, für die natürlichen Benutzerschnittstellen bekannt sind (Benutzerschnittstellen, die eine Art von "*natürlicher*" Interaktion bieten, welche der Benutzer bereits kennt und sie deshalb nicht erneut erlernen muss). Wenn man den Fakt, dass Menschen normalerweise bereits multimodal miteinander kommunizieren (mit Gesten, Sprache, Augenbewegungen etc.), betrachtet, ist offenbar, dass das simultane Benutzen mehrerer Modalitäten in einer Schnittstelle keine Nachteile im Bezug auf die kognitive Last verursacht. Da durch die Kombination mehrerer Modalitäten die Schnittstelle nur auf eine Eingabe des Benutzers reagiert, wenn dieser mehrere Modalitäten einsetzt, wird das System automatisch fehlertoleranter im Vergleich zu einer Benutzerschnittstelle, die nur eine Modalität benutzt (hier könnte bereits eine simple, nicht beabsichtigte Geste eine falsche Reaktion des Systems hervorrufen). Ein Anwendungsgebiet für diese multimodalen Benutzerschnittstellen wären zum Beispiel Systeme, die wissenschaftliche Daten visualisieren. Diese Daten werden meist aus Kalkulationen, Simulationen oder Aufnahmen gewonnen und sind in Rohform meist sehr groß. Diese große, komplexe und oft dreidimensionale Natur der Daten macht anspruchsvolle Visualisierungs- und Interaktionstechniken notwendig. Dadurch werden die Interaktionstechniken, die für Navigation in und Umgang mit den Daten benötigt werden, allerdings auch sehr komplex, was wiederum sehr gut mit multimodalen Interaktionstechniken gelöst werden kann. Der Fokus dieser Arbeit liegt auf solchen multimodalen Interaktionstechniken für Visualisierungen wissenschaftlicher Daten. Insbesondere wird die Behauptung, dass eine synergistische Kombination von Modalitäten effektiver und intuitiver ist, als andere Kombinationen und gar keine. Nach einem umfangreichen Überblick über die Themen multimodale Interaktion und Visualisierung wissenschaftlicher Daten wird ein voll erweiterbares System dargestellt, welches solche Interaktionstechniken implementiert. Zum Abschluß wurde das System mit zwei verschiedenen Kombinationen von Modalitäten in Hinsicht auf intuitive und effektive Interaktion mittels einer Benutzerstudie evaluiert.

TABLE OF CONTENTS

1	INTRODUCTION	3
1.1	Terminology	4
1.2	Motivation	5
1.3	Thesis Goals	7
2	SCIENTIFIC DATA VISUALIZATION	11
2.1	Theory and Practice	11
2.2	Interaction Paradigms	13
2.3	Seismic Interpretation	13
2.4	Previous Work	15
3	MULTIMODAL INTERACTION: THEORY	19
3.1	Finite State Machines	20
3.2	Fusion Engine	21
3.3	Cognitive Load	23
3.4	Models and Frameworks	23
3.5	Previous Work	25
4	MULTIMODAL USER INTERFACE FOR SDV	29
4.1	Requirements	29
4.1.1	Scientific Data Visualization Part	29
4.1.2	Volume Renderer	30
4.1.3	Multimodal Interaction Part	31
4.2	General Approaches	31
4.2.1	Interaction Tasks	31
4.2.2	Interaction Techniques and Modalities	32
4.2.3	Finite State Machine	33
4.2.4	Handpointers	35
4.2.5	Feedback	36
4.3	System Overview	37
4.3.1	System Architecture	37
4.3.2	Remote Workstation	38
4.3.3	Tracking Machine	38
4.3.4	Rendering Machine	38
4.4	Software Architecture	39
4.4.1	Overview	39
4.4.2	ActiveMQ Connection	40
4.4.3	NIFramework	41
4.4.4	Main Application	43
5	USER STUDY	49
5.1	Objectives	49
5.2	Realization	51
5.3	Results	53
5.3.1	Subjective	53
5.3.2	Objective	55
6	CONCLUSION AND FUTURE DIRECTION	61
6.1	Conclusion	61
6.2	Future Direction	62
	BIBLIOGRAPHY	65
	APPENDIX	73

LIST OF FIGURES

1.1	Visualizing Large-Scale Atomistic Simulations in Ultra-Resolution Immersive Environments	5
1.2	The Evolution of User Interfaces	6
1.3	Volume and Slice based Rendering Techniques	8
2.1	SDV from different Fields	11
2.2	Tooth Visualization with 2D Transfer Functions	12
2.3	Seismic Interpretation	14
2.4	The Virtual Windtunnel	15
2.5	Pictures of the MSVT Application	16
3.1	Multimodal Input and Output	20
3.2	Finite State Machine Diagram	20
3.3	Evolution of Fusion Engine Methods	22
3.4	Learning Effort of NUIs	23
3.5	The NiMMiT Modeling Framework	24
3.6	The put-that-there Interface	25
3.7	Novel Pen & Touch based Interaction Technique for Seismic Interpretation	26
4.1	The Volume Renderer created at Fraunhofer IAIS	30
4.2	3D Navigation Figure	32
4.3	Finite State Machine Diagram (Unimodal)	34
4.4	Finite State Machine Diagram (Multimodal)	34
4.5	Handpointers in 3D-space	35
4.6	System Architecture	37
4.7	Software Overview	39
4.8	The NIFramework	42
4.9	The NIFramework (Architecture)	43
4.10	The Main Application (Architecture)	44
4.11	A Photo of the entire System	46
5.1	A Photo of the User Study	51
5.2	Screenshots used for the User Study	52
5.3	Chart with the Mean Results of Questions 3 and 4	53
5.4	Chart with the Mean Results of Questions 8 and 9	53
5.5	Chart with the Mean Results of Questions 5, 6 and 7	54
5.6	Chart that shows Mean Time Taken for each Task	55
5.7	Chart that shows the Mean of how many Errors were made	57
5.8	Chart that shows how many times Participants approached the tutor	58

01

INTRODUCTION

1. INTRODUCTION

It is the vision of a single interface which is able to address and understand all human senses for communication (modalities) as imaginable that drives the research field of multimodal user interfaces [62]. These interfaces are a subset of natural user interfaces, which are in turn meant to replace the well-known WIMP paradigm used in many graphical user interfaces by making the interaction between a user and a computer more natural and intuitive.

The recording and recognition of tracking data for a single modality is nowadays nothing special anymore – devices such as the *Microsoft Kinect* [38] or the *Wii Remote* [63] track a user’s movements, interfaces such as the *Microsoft Speech API* [39] can understand a user’s spoken commands and devices such as *The Eye Tribe* [60] even make it possible to track a persons eye movements.

Using these devices and technologies, developers of natural user interfaces aim to implement interaction techniques that are modeled similar to what a user does already know from other areas (such as human-to-human interaction). The main goal for every interaction technique is therefore to find the utmost natural way of designing it – for instance, the best approach for implementing an interaction technique for manipulation of 3D objects would certainly be a simple grabbing metaphor.

The combination of these devices, modalities and techniques in a single interface proofs to be a challenging task however [62]. More importantly, in what fashion the modalities are combined and to what degree the user is forced to use more than one. This combination techniques are commonly referred to as *Fusion Engine* or *Multimodal Integration* [43] (discussed in more detail in chapter 3).

Where the aim of multimodal interfaces is usually to design the interaction more natural and intuitive, using this type of interface together with the visualization of complex scientific datasets reveals another important advantage: Since the datasets are very often massive in size and complexity (see chapter 2 for more details), easing the cognitive load that weighs on a user during the interaction can make the interpretation process of such datasets more intuitive, efficient and natural. [14].

In this work, a system with a multimodal interface that is used to interact with and interpret on scientific data is described. In addition, an extensive overview of the current state-of-the-field in multimodal interaction and in scientific data visualization is provided beforehand. The described interface includes different types of combination (fusion engine methods) between multiple modalities for comparing purposes. At last, the system is evaluated by comparing two different fusion engine methods in a user study.

The remainder of this work is structured as follows: This chapter covers a brief overview of what is content and purpose of this thesis, it includes a section that contains a list of terms that are meant to improve the flow of reading within this work (section 1.1) in addition to a short motivation and a brief summary of the thesis goals. Chapter 2 then covers an overview of current and past applications that feature scientific data visualization as well as theory and common practices. Chapter 3 contains theoretical approaches in the research field of multimodal interaction. Both chapters (2 & 3) include previously in their specific area presented work that stand in correlation to both topics. Chapter 4 introduces the application part of this thesis with used general approaches, a system overview and software architecture. The realization and results of the user study are explained in chapter 5. In the end, this thesis is concluded with a view on possible future work in chapter 6.

1.1 Terminology

A few frequently in this thesis used terms are listed below.

Human–Computer Interaction (HCI)

The combination of both input and output, where a user is able to interact with a computer and simultaneously receive feedback from it, is commonly known as *Human–Computer Interaction* (HCI)[10].

Virtual Environment (VE)

As soon as an application gives the user real-time control over a rendered, spatial world through some kind of first-person point of view (or virtual camera), it is referred to as a *Virtual Environment* (VE)[10].

User Interface (UI)

An interface that allows communication between a user and a computer.

Interaction Technique (IT)

An *Interaction Technique* is the action a user has to perform to reach a desired goal (e.g. a grab gesture is used to manipulate objects).

Interaction Task

Tasks that represent what the developer intends the user to be able to do with the application (e.g. object selection and object manipulation).

Interaction Metaphor

A metaphor from reality used to describe an IT (e.g. grabbing real objects – grabbing virtual objects).

Modality

Means of interaction between a user and a computer, always uses one corresponding communication channel (e.g. gestures, speech).

Multimodal

Combination of multiple modalities for input and/or output in a single interface (e.g. gesture and speech).

Scientific Data Visualization (SDV)

An application that renders and visualizes large sets of scientific data, often in immersive and collaborative environments (see for instance figure 1.1).



Figure 1.1: Example of an application including SDV in an immersive and collaborative environment: Visualizing Large-Scale Atomistic Simulations in Ultra-Resolution Immersive Environments [54]

1.2 Motivation

Since the very early days of the computer, engineers have been struggling with the idea of allowing every user access to a wide spectrum of functionality on a computer. In fact, the very first means of communication between a user and a computer could merely be called an *interface* since engineers were only able to use “punched-card input and line-printer output” [64]. Andries van Dam once stated that in this early phase (1950 - 1960) “...there were essentially no user interfaces because there were no interactive users” [64]. He further wrote that the second phase, during which engineers were able to type in commands and receive text feedback (a first *real* interface, the so-called *Command Line Interface* (CLI), see figure 1.2), lasted until the first *Graphical User Interfaces* (GUIs) were introduced by the Macintosh in 1984 (initially invented at Xerox PARC during the 1970s [64]).

GUIs are specifically designed to give every user easy access to a wide variety of functionality on a computer using simple input devices such as a mouse and a simple design using point-and-click metaphors – with success: Nowadays, the so called *WIMP GUIs* (GUIs that use *Windows, Icons, Menus and Pointers* together with a point-and-click metaphor [10]) are widely established and give any user access to such functionality with minimal learning effort. Thus, GUIs have fulfilled their purpose and the initial goal, granting every user access to the computer, has been reached – but the research in the fields of UIs is continuing to grow, why is that?

Imagine a person, who has never used a common input device such as a keyboard or mouse, is asked to write a simple text document on a computer. Even though WIMP metaphors and such common devices are very simplistic, it would still require the user some time to adapt to these devices and the WIMP GUIs. What if, on the other hand, a UI would be as intuitive as one can be – where a user does not have to learn anything before using it? What if a UI would feel as natural as if there were no interface at all? What if the UI would be able to learn and adapt to a user instead of the user having to adapt to the interface? The interfaces implied by these questions are commonly referred to as *Natural User Interfaces* (NUIs). These are the successors of the GUIs (see figure 1.2), with the focus on intuitive, natural interaction rather than solely on usability. They have been a major topic in the research fields of *Virtual Reality* (VR) and HCI for a long time.

"Ever since Sutherland's vision of the ultimate display, the notion of interacting with computers naturally and intuitively has been a driving force in the field of human-computer interaction and interactive computer graphics." [31]

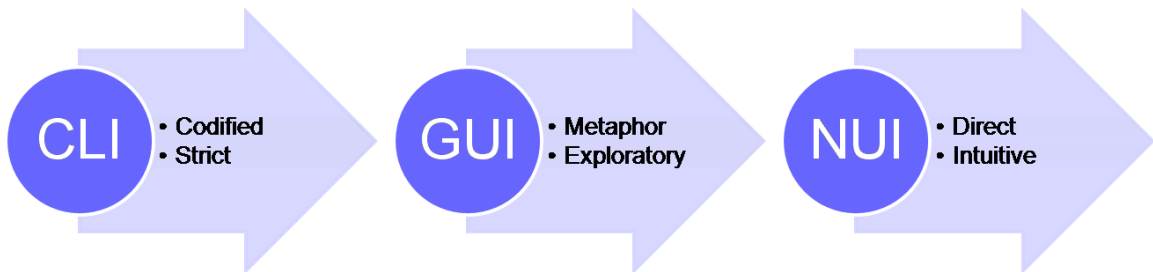


Figure 1.2: The Evolution of User Interfaces [66]:
CLI (Command-Line Interface), GUI (Graphical User Interface), NUI (Natural User Interface).

Indeed, Ivan Sutherland already predicted that the research field of UIs would one day turn into this direction in his vision of *The Ultimate Display* [58] in 1965. His version of an *ideal interface* could not only understand everything a user did and felt, but also alter the physical world – allowing a user to sit on a virtual chair.

Therefore, the next goals in the research field of UIs are clear – provide or even further improve the ease of use that common input devices nowadays already maintain while improving HCI with powerful and natural interfaces that are able to handle even the most complex situations and data types.

"While conventional human-computer interaction paradigms (e.g., Graphical User Interfaces) are useful in personal computing applications such as word processing, they do not adequately support tasks that require the manipulation of complex data types and constraints in the way intelligent, interactive systems have the potential to do." [20]

Above any future potential that might arise from NUIs, a view on application scenarios where NUIs have already been established to some degree raises the motivation even further. For instance, designers at car design facilities are able to look at new designs while being immersed in *Cave Automatic Virtual Environments* (CAVEs) [12]. The advantage is rather obvious: Without the immersive environment, the designer would have to look at pictures on a computer screen, sketch ideas on paper and imagine the spatial context all by himself. In addition, the VE would not only enhance efficiency, but could also increase creativity simply because the designer is able to try out different textures and shapes much faster.

A few more examples include:

Medicine

Novel interfaces dedicated to medicine aim to improve work flow in training and clinical applications [8] and even psychiatric treatment [12].

Scientific Data Visualization

Scientific data visualization can be improved by using powerful algorithms for rendering and NUIs for interaction [37, 30, 1, 54].

Home Environments

Aside from the increased efficiency that may be achieved in working environments, novel interfaces could also improve home environments through, for instance, televisions without remote control device.

Gaming

Probably the area with the largest impact of NUIs: The gaming industry and more specific, gaming consoles [36].

"Unless you've been living underground for the last couple of years, you know that the Nintendo Wii has taken the gaming world by storm." [11]

Other

Certain functionality of computers can be made available to physically disabled people with such interfaces [40, 27].

In conclusion, there is a lot of potential in NUIs in many different areas. This thesis will mainly focus on NUIs, or more specific, multimodal interfaces in SDV and what combination of modalities is best suited and has the best user acceptance.

1.3 Thesis Goals

The research field of *MultiModal Interaction* (MMI) has been a large portion of the research in NUIs since a long time [27, 62, 7, 30]. The field describes the (in most case simultaneous) usage of multiple modalities and human senses for communication between a user and a computer. Since SDV use large sets of data (often three-dimensional) which require complex ITs for users to be able to maneuver around the data, the argumentation that *MultiModal Interaction Techniques* (MMIT) are better suited for these complex tasks comes to mind. Indeed, researchers argue that since it is not necessary for the user to change the state of the system via a menu or some other kind of system control, it reduces cognitive load and therefore, allows for more complex ITs:

"Using an input channel that differs from the main input channel used for interaction with the environment, can decrease user cognitive load. If users do not have to switch between manipulation and system control actions, they can keep their attention focused on their main activity" [10]

"A third more significant advantage is the flexibility that multimodal systems permit users in selecting and alternating between input modes" [48]

On the contrary, the benefit of using multiple modalities highly depends on the manner in which the modalities are combined. The synergistic fusion engine method (or combination) of modalities, where a user must use multiple modalities simultaneously to perform a single task, is considered the most effective [62]. The fusion engine, cognitive load and MMI in general is discussed in chapter 3 in more detail.

The combination methods expressed through the synergistic model (parallel) and through the alternate model (sequential) have already been tested in the past in many different scenarios [67, 68, 50]. That is why the choice for the comparison of fusion engine methods evaluated in the present work fell on synergistic and concurrent – the concurrent describing an interaction that although it features multiple modalities that can be used in parallel, they are not integrated. Furthermore, if the evaluation takes place for each task separately, the ITs in the concurrent model can be considered a unimodal interaction since the user does only use a single modality for each specific task. The scope of this work can therefore further be narrowed down to a comparison between unimodal and multimodal interaction separated in single tasks, where the unimodal interaction can occur with the same modalities used in the multimodal interaction (yet not combined).

The main goals of this thesis can be expressed through the following hypotheses:

Hypothesis 1

The use of a synergistic fusion engine method, where a user is forced to use multiple modalities simultaneously, eases cognitive load and is therefore easier and faster to learn as other fusion engine methods or none at all.

Hypothesis 2

The use of a synergistic fusion engine method is more fun to use and gets more user acceptance than other combinations or none at all.

Using different approaches, measurements (e.g. error robustness, time needed for specific tasks) and questionnaires, these hypotheses will be evaluated using usability engineering techniques in a user study which is fully described in chapter 5.

Aside from the theoretical part, this thesis comes along with a fully functional application which features MMI and SDV. More specific, the application visualizes seismic data using volume- and section-based rendering techniques while giving the user control over the virtual camera, system control tasks and control over the data assets such as the sections with MMITs. The complete system is described in chapter 4.

The section-based rendering technique uses 2D sections throughout a 3D set of data, which give users a 2D view of the selected position within the data (described in more detail in chapter 2). Using this technique, interpreters of data in various fields can more easily access the data on desktop workstations with 2D displays. It is often used in SDV that include 3D sets of data since the data is difficult to navigate around and get a good look at due to its three-dimensional nature. This is demonstrated in figure 1.3: In the right frame, the center of the data is occluded by the data itself, whereas in the left frame, the center is visible on the sections.

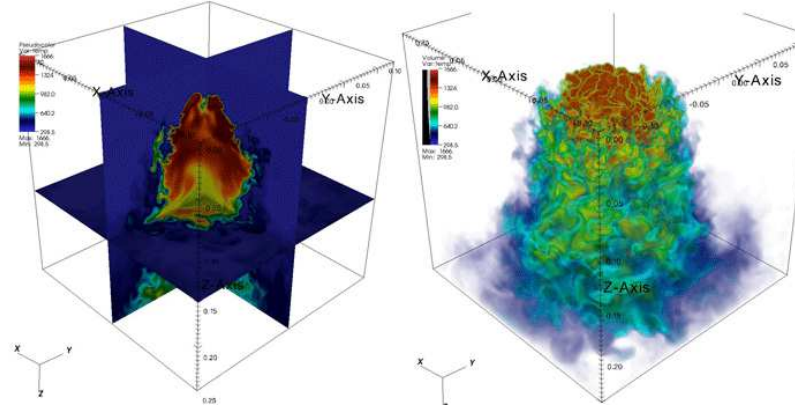


Figure 1.3: Volume (right) and slice (left) based rendering techniques for scientific data [5].

The description of the system in chapter 4 aims at giving the reader an idea or impression on how such an interface could be implemented. Regarding this aim, the system has been implemented in a fully expandable way.

Since the application part of this thesis uses seismic data visualization, the need to explain how this kind of datasets is usually used and interpreted is present. Thus, a brief explanation of seismic interpretation can be found in section 2.3.

02

SCIENTIFIC DATA VISUALIZATION

2. SCIENTIFIC DATA VISUALIZATION

This chapter covers commonly in SDV used theories and practices, related work and a description of seismic interpretation.

2.1 Theory and Practice

SDV describes the visualization of data that was the result of scientific measurement or similar and its aim is to allow scientists to interpret and understand the recorded data:

"Scientific visualization is the use of computer graphics to create visual images that aid in the understanding of complex (often massive) numerical representations of scientific concepts or results." [14]

Novel technologies nowadays make it possible for scientists and researchers from various fields to record data from many different sources [14]. For instance, seismic data that is usually invisible under the ground can be captured by sending sound waves into the ground and recording the reflections [23, 37]. Since expensive technologies are required for these tasks it is often wanted to record very large sets of data at once, so that starting the process of recording is worth it to begin with. When recording seismic reflections, for instance, scientists in the oil and gas industry usually record data of a few hundred kilometers at once and therefore, these recordings usually consist of hundreds of gigabytes of data stored in databases or files. Aside from seismic data visualization, which is used in the practical part of this thesis, SDV is used in many other areas. These include, for instance, atomic visualization [54] (figure 1.1, medicine (figure 2.2 and figure 2.1, right), fluid data visualization (figure 2.1, left), and many more [30, 14, 1]. In some of these fields, the data can also be acquired through other means like, for instance, numerical calculations or simulations [14].

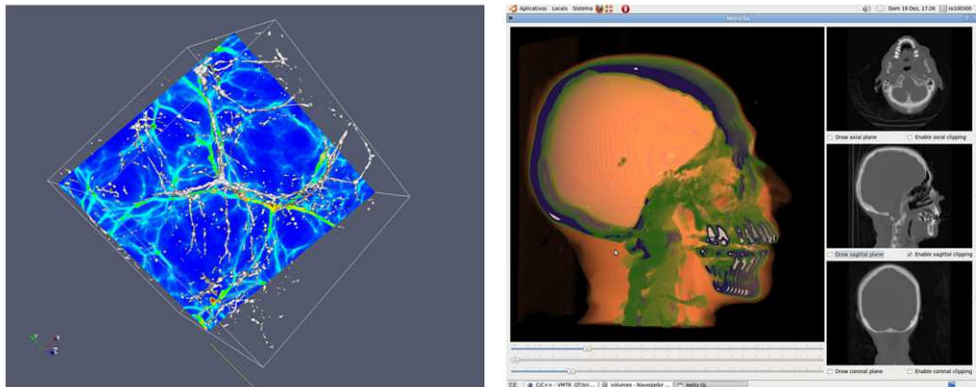


Figure 2.1: Fluid (left) [17] and medicine (right) [22] data visualization.

As soon as the recording and storage process is finished, experts of the specific fields have to get a good look at the data so they can start interpreting it. Thus, making effective techniques for visualization necessary. If the plain data is visualized without any aid of visualization or other tools, it would be a non-transparent three-dimensional shape where the center is completely occluded by the data itself. This would make it impossible for experts to even begin interpreting the data – thus, some kind of calculation has to take place where the center or region of interest of the data gets visible.

If seen as a 3D geometry, every pixel or point of the shape has some defined attributes which were recorded along with the data, e.g. classification by color or opacity which represent real attributes like density of a material. These attributes can then be mapped on a 1D or 2D diagram which scientists can use to define a range of values that they want to be visible while the rest gets transparent or even fully invisible (see figure 2.2). These mappings are commonly referred to as *transfer functions* [23].

In figure 2.2, three different transfer functions of a tooth visualization are compared visually: The frame at the bottom of each individual visualization can be understood as a 2D diagram, where all points of the data are mapped to their specific attributes (e.g. color and opacity are commonly used attributes [23]). The points and lines visible in the diagram in the left frame are the region, the scientist has chosen to be visible. The result is shown in the top part of the left frame – only inner parts of the tooth are visible, whereas the outer part is completely transparent. In the other two frames, other areas have been chosen and therefore, other transfer functions are being applied.

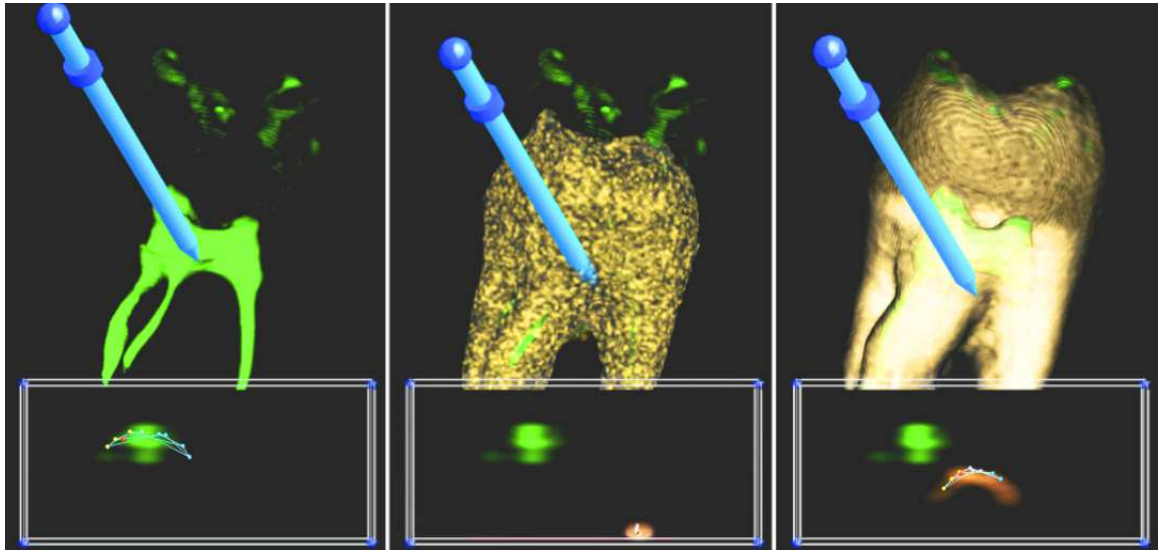


Figure 2.2: SDV of a tooth with drilling region and three different transfer functions. [28]

The purpose of the visualization in figure 2.2 is evident: A dentist wants to know where he has to drill to reach a desired region within the tooth. Here, the 3D visualization gives a great advantage over section-based rendering techniques which only grant a 2D section view of the data.

Since these large sets of data need to be rendered in a volumetric way to allow these transfer functions to be applied, rendering techniques for such volumes need to be added to the visualization. Aside from the well-known *raycasting* algorithm that simply renders a point cloud [55], one technique that is also often used is the slice-based rendering technique. An example for the slice-based technique is the so-called *octreemizer* [53].

2.2 Interaction Paradigms

Basically, there are two interaction paradigms to be distinguished: The VE and the desktop environment. The VE features large displays (often stereoscopic), intuitive interaction and large spatial capabilities, whereas the desktop environment is based on the classic WIMP interfaces using a two-dimensional display with mouse and keyboard as interaction devices. Where the VE is better suited for three-dimensional and large datasets, collaboration, immersion and intuitiveness, the lack of preciseness, robustness and familiarity still causes most scientists to use desktop environments for the daily work-flow of SDV. Although researchers involved in that field argue that the disadvantages, even though they are many, get outweighed by the advantages:

"However, all of these technologies suffer from various problems, including cost and lack of accuracy, although the advantages seem to significantly outweigh the disadvantages." [14]

Thus, the motivation to find a perfect solution for the VE paradigm in SDV application is very high. And, even though the drawbacks of the VE paradigm continue to make a fully stable and natural application impossible, there have been a lot of applications that feature VR and SDV already developed in the past that are being used on a regular basis [14]. Here, robustness is improved by, for instance, replacing single purely natural devices with more stable ones at the cost of naturalness. A good and well-known example for this is the fairly often in VEs used Wii Remote [63].

"Virtual reality and scientific visualization are well matched for several reasons, in addition to inherently three-dimensional display and control. Scientific visualization is oriented toward the informative display of abstract quantities and concepts, as opposed to an attempt to realistically represent objects in the real world." [14]

However, the disadvantages last to this day: Novel devices and technologies such as the well-known Microsoft Kinect [38], the Leap Motion [33] or speech recognition solutions (e.g. [39]) are still not perfectly accurate. This causes scientists in various fields to still rely on the workstation. But at what cost? Visualizing three-dimensional, large datasets on a small to average sized, two-dimensional display with two-dimensional input devices (mouse)? The solution to this issue is to reduce the data to two dimensions and visualize it on so-called *sections* [37] (see figure 2.1). Another commonly used solution is to allow scientists to adjust the view by using the mouse and, for instance, a *virtual trackball* metaphor [37].

In conclusion, while researchers are searching for an ideally working VE, scientists rely on the solid and robust desktop paradigm together with specifically designed techniques for handling the datasets on a workstation. One possibility to achieve the ideally VE with the nowadays devices and technologies could combine the two paradigms in one application: What if the tasks that greatly benefit from using the large display, collaboration and natural interaction can be performed while still being able to perform tasks that require high levels of precision on a 2D display within the same interface? This would certainly be a matter of interest for future work.

2.3 Seismic Interpretation

As mentioned in the beginning of this chapter, seismic data is acquired by sending sound waves into the ground and recording the reflections:

"Seismic reflection data, often referred to as seismic reflection volumes, is acquired by sending seismic waves, often sound waves from explosions, into the earth and recording their echoes. When a wave hits the boundary of two subsurface layers, a seismic horizon, it is partly transmitted to the lower layer and partly reflected back to the surface." [23]

These reflections are then mapped over time and visualized by giving every subsurface layer a different color (see figure 2.3, left). At this stage, it is not yet clear what material the subsurface layers consist of, which in turn means that the visualization alone does not give the scientists the information they want. Thus, the acquisition is only the first step of the job, whereas afterwards the data has to be properly interpreted by experts. This process is referred to as *Seismic Interpretation*.

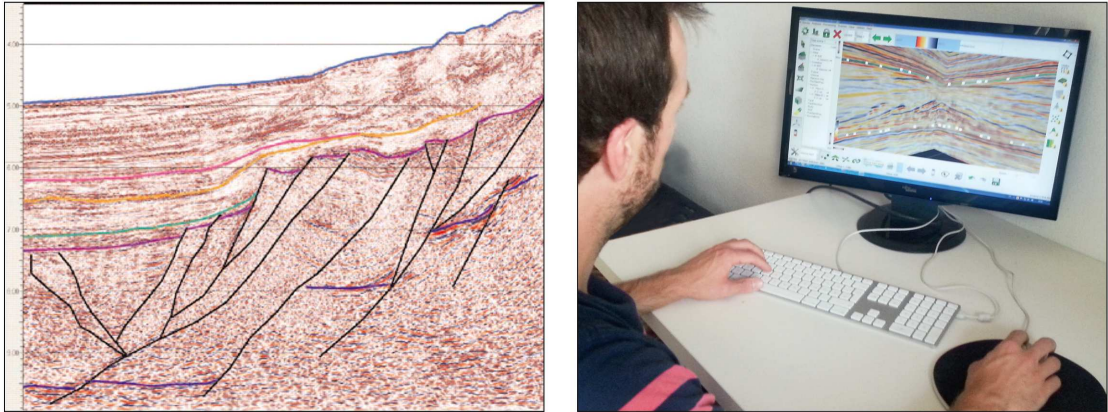


Figure 2.3: Seismic data interpreted on a seismic slice (left) [4] and a geoscientist performing seismic interpretation on a desktop workstation (right) [37].

One of the main aims of seismic interpretation is “...to trace continuous subsurface structures throughout a seismic volume and ultimately create a 3D model of them” [23]. Two of the most significant 3D models created in this way are so-called *horizons* and *faults* [23]. Where horizons are typically horizontally aligned (yellow and pink lines in figure 2.3, left), faults usually present in a more vertical shape (black lines in figure 2.3, left).

Aside from the seismic interpretation tasks that are being performed by single scientists such as *horizon picking* (shown in figure 2.3, right), scientists in the oil and gas company often rely on installations with large or multiple small displays for collaboration [51]. By collaborating with multiple scientists from different disciplines at once, false decisions can be reduced. Overall, the interpretation is a very crucial task since upon its results, decisions whether they want to start drilling at a specific location or not are being made:

“The petroleum geologist, who works on seismic interpretation, is an integral part of the complete hydrocarbon exploration workflow. It is in his responsibility to make an informed decision about whether or not oil and gas deposits are located in the geologic structures and based upon these considerations to advise where to drill.” [37]

The application part of this thesis includes seismic data being visualized on a large display using volume- and section-based rendering techniques. Here, it would be interesting to have an interface which features both three-dimensional interaction on a large display and two-dimensional interaction on a smaller display. Where the large display has advantages such as collaboration, immersion and lots of space for devices that need as much (e.g. *Microsoft Kinect* [38]), the 2D display has the advantage of familiarity, robustness and precision. Even though seismic interpretation is a much larger topic than this brief description make it seem [37, 23], giving a full overview goes beyond the scope of this work.

2.4 Previous Work

There have been many application in the past that aimed at visualizing scientific data in an immersive environment. Using an immersive environment or, more specific, a VE holds many advantages over the usual desktop workstation. For instance, the three-dimensional and largely scaled nature of scientific data is much more easy to handle on large and stereoscopic displays like those widely used in VR. Another reason is the collaborative aspect – it is believed that interpreting data whilst constantly communicating with other scientists greatly improves performances when handling complex data [51]. At last, interacting with the data in a natural way is argued to give a better impression of the data.

"We want to create the effect of interacting with things, not with pictures of things." [14]

One of the older surveys that largely promotes VEs with natural interaction in SDV was written by Steve Bryson in 1996 entitled *"Virtual reality in scientific visualization"* [14]. He sought to create a description of how to effectively implement SDV applications using VR tools and devices – and very detailed: Explanation on how to reduce computation time, how to subsample and compress data, how to effectively visualize it and, of course, how to interact with it intuitively. The article is very well correlated to the topic of the present work since the *"main point"* of it *"...is that virtual environment interfaces can enhance the role of scientific visualization in the scientific discovery process. Such a system clearly requires interactive capabilities that allow intuitive control of the data visualization displays, avoiding as much as possible difficulties due to the arbitrariness of the interface."* [14]

One year later he introduced another paper which main aim was to improve his past SDV application entitled *"The Virtual Windtunnel"* [15] by using the recently learned conventions. It is called *"An Extensible Interactive Visualization Framework for the Virtual Windtunnel"* [13] and it again promotes the use of VR in SDV applications.

The original application from 1992 visualizes *"...the results of computational fluid dynamics (CFD) simulations"* [13] in a so-called *"Windtunnel"* in 3D and gives the user control of it using gestures or buttons [15]. The developers only defined three gestures throughout the entire application: *grab*, *point* and *null* [13]. The gestures are, depending on the situation the user is in, mapped to different commands which has the immense advantage that the user only has to memorize three gestures before completely understanding the interface. Furthermore, head-tracking was included to maximize the levels of immersion for the user.

The *virtual windtunnel* has been a milestone in the field of SDV and many researchers followed his example of including VEs with natural interaction in their SDV applications since [30].

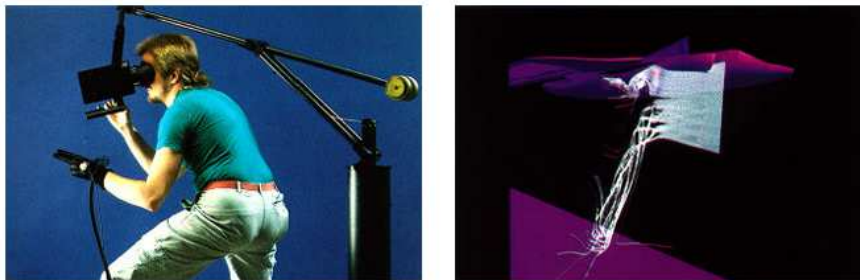


Figure 2.4: A photo (left) and a screenshot (right) of the virtual Windtunnel system from Bryson [15].

Another often referenced SDV application was introduced by Joseph Laviola Jr. as follow-up on his master thesis *"Whole-Hand and Speech Input In Virtual Environments"* [32]. The resulting application entitled *"MSVT: A virtual reality-based multimodal scientific visualization tool"* was introduced one year later [30]. Using the interface, users are sitting in a chair in front of a tilted, stereoscopic display and are able to interact with it using speech recognition and hand gestures recognized by worn gloves (see figure 2.5). The immersion levels are very high due to the fact that a user has the data in front of him like objects lying around on a usual desk.

The IT used by the interface is similar to those in the put-that-there [7] interface, which also featured simultaneous use of speech and hand gestures (explained in more detail in section 3.5). In fact, this combination of the two modalities is very often used in multimodal interfaces for a number reasons: The user does not have to wear any device (except when using gloves, but that is not the usual case), interaction with objects using the hands and fingers is one of the mostly evolved interaction paradigms of humans [58, 37] and *"...since human-to-human interaction often occurs with combinations of speech and hand movement."* [30]



Figure 2.5: A user currently using the MSVT application featuring SDV [30].

On the contrary, the disadvantages of the interface are evidently visible in the pictures in figure 2.5: A user has to hold his hands and arms up the entire time of the interaction which would cause fatigue. Furthermore, researchers argue that the use of glove reduces the naturalness of the interaction since a user is normally not wearing gloves when sitting at a desk [14]. This is nowadays not an issue anymore due to fairly accurate optical sensors such as the *Microsoft Kinect* [38]. At last, the user is attached to many cables to the worn devices such as earplugs and gloves, which could impair the interaction.

One more interface that features volume rendering of scientific data and allows interaction using the *Microsoft Kinect* [38] was introduced last year and is entitled *"Natural User Interfaces in Volume Visualisation Using Microsoft Kinect"* [1]. An example field of application for the interface would be providing a virtual training application for medical sciences.

Especially interesting about the interface is that it uses no input devices worn by the user and that it uses the *pointcloud library* [55] to render a 3D representation of the users hands [1].

03

MULTIMODAL INTERACTION: THEORY

3. MULTIMODAL INTERACTION: THEORY

As already described in the motivation section of this work, NUIs aim at giving the user a natural and intuitive way of interacting with a computer. But what does that mean precisely? *Natural* or *intuitive* can be explained as *having a familiar feeling while trying something new*, which in turn means that it is not required of users to intentionally memorize any ITs used by an interface, but instead users are already used to the ITs because they are well known from other areas. For instance, while speaking to one another, a human unintentionally and intentionally communicates through body language such as postures, gestures, eye gaze or facial expressions as well [26, 30]. Therefore, interacting with an interface using speech while maintaining a push-to-talk posture, for instance, would not feel natural at all simply because the user would be unable to do unintentional gestures that naturally come along with speaking. Considering this, the interaction in multimodal interfaces feels more natural for users than in unimodal interfaces (interfaces, that just uses a single modality such as gestural interfaces).

Aside from the increased ease of use that is achieved because of the similarities to HHI, there is one more big advantage to using MMI (psychological aspect):

"...there is growing evidence that simultaneous stimulation of multiple modalities can influence the activity in unimodal sensory areas and improve or impair performance in unimodal tasks." [2]

This means that without users noticing it, they can get more effective at unimodal tasks while interacting in a multimodal way, if utilized correctly. These two aspects are considered to be

the main advantages of interfaces that use MMI: First, ease of use, intuitiveness and naturalness, and second, exploiting the full potential of a users perceptual capabilities [24]. To further exploit that potential, a designer of a multimodal interface has to design both input and output in a multimodal way. If users interact with an interface using speech, they will automatically expect audio as feedback as well. This is, again, due to the nature of HHI: While talking on the phone, the feedback to speech input is audio output, when writing an email, both input and feedback consist of text. Receiving a text message as feedback when talking on the phone does not happen and would therefore feel utmost unnatural. This correlation between different modalities is also known as *"...the binding problem, and the traditional assumption has been that only at the highest levels of brain functioning in the cortex are sensory streams integrated, and they interrelate only through experience."* [62].

A NUI using both input and output in a multimodal way is figuratively described in figure 3.1. The figure further shows, that multiple actions can be mapped to a single sense: For instance, if interacting with the computer using speech recognition, the audio feedback is not necessarily speech output but could also be simple audio cues without penetrating the barrier to another sense or modality.

So far this chapter has been about aims and basics of MMI. The next sections will cover common theoretical approaches and practices, challenges at an implementation stage and previous work.

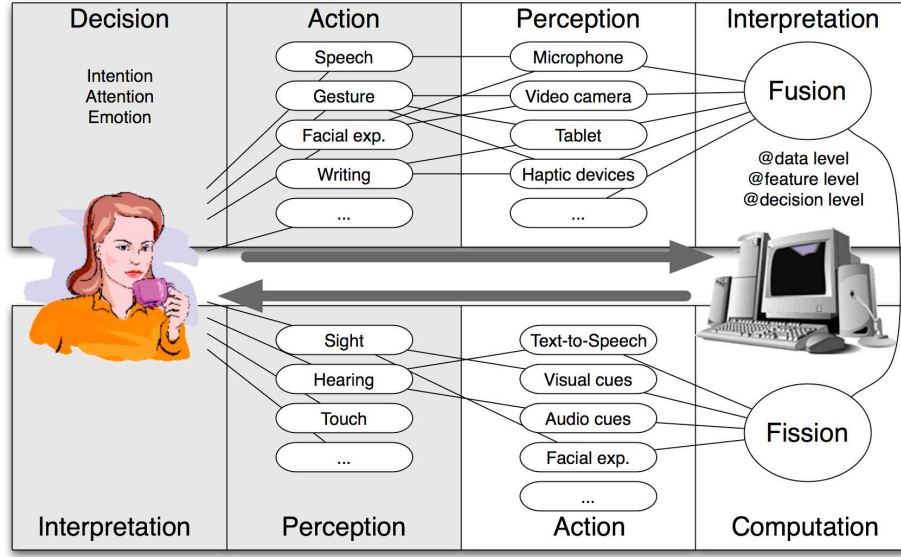


Figure 3.1: Interface with both input and output multimodal [24].

3.1 Finite State Machines

Modeling frameworks or languages for NUIs such as the NiMMiT-framework [6] described in section 3.4 often rely on so-called *Finite State Machines* (FSMs). These FSMs mainly aim at simplifying the development process of NUIs: A developer is able to describe the entire interface with a wide variety of complex ITs with the use of only a few diagrams (see for instance figure 3.2).

"As FSMs can represent multiple states and also identify multiple actions which provoke the change of states in a certain system with only a single or a few diagrams, we can easily describe and understand the structure of multimodal interactions using FSMs." [19]

This is not only advantageous for the initial implementation of the interface due to a good overview and thus, error prevention, but also for future changes after completion or a redesign process.

In figure 3.2, a simplistic example of an application described using a FSM is shown. The application pictured consists of a total of 6 states which are represented by circles (e.g. s_1s_1 , s_1s_2) and which have state transitions between them represented by single characters (a, b).

It is fairly easy to see that special characteristics can be very comfortable pictured: For instance, see the state transitions from state s_1s_3 to s_1s_2 . Here, the user is offered two possibilities to reach state s_1s_2 from s_1s_3 .

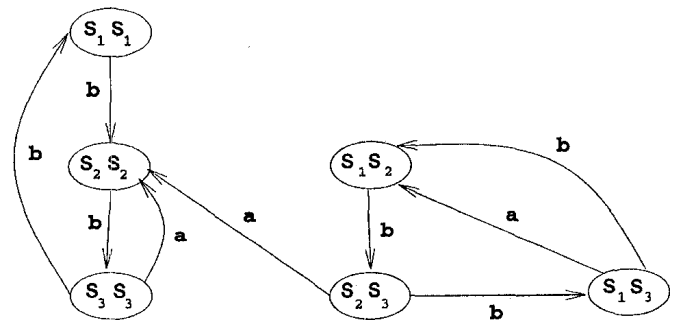


Figure 3.2: A simple FSM diagram example. [34]

3.2 Fusion Engine

One of the myths in Sharon Oviatt’s article *Ten Myths of Multimodal Interaction* [48] states that when users are facing a multimodal interface, they will not necessarily interact with it multimodally:

“...just because users prefer to interact multimodally is no guarantee that they will issue every command to a system multimodally. Instead, they typically intermix unimodal and multimodal expressions.” [48]

The article further references to a study which has shown that users interact only about 20% of the time multimodally if they have the choice [49]. The rest of the time, users tend to use a mix of unimodal and multimodal interaction or even just stick with the unimodal interaction. That is one of the reasons why the so-called *Fusion Engine* or *Multimodal Integration*, which defines to what degree modalities are combined with each other, is considered to utmost key technical challenge for developers of multimodal interfaces [62]. Because if the modalities are not combined properly, users might just ignore the possibility of using them and stick with the unimodal interaction, which would cause all advantages of the MMI to be in vain (not to mention the hard work that it is to develop a multimodal interface).

Thus, the fusion engine method should be one of the major concerns when designing an interface for MMI.

A well-known and often referenced [24, 62, 29] classification of fusion engine methods is derived from Nigay’s and Coutaz’s article in 1993 “*A Design Space For Multimodal Systems: Concurrent Processing and Data Fusion.*” [43]. It defines four different methods in a 2 by 2 table (see table 3.1).

		Time correlation	
		Sequential	Parallel
Combination	Independent	Exclusive	Concurrent
	Combined	Alternate	Synergistic

Table 3.1: Four fusion engine methods for multimodal interfaces [43].

In table 3.1, the rows distinguish by combination, where *Independent* means that the user has the choice between using unimodal interaction or multimodal interaction – the modalities are therefore not integrated. *Combined* means that a user has to use multiple modalities to reach a single goal or to perform a single task. The columns distinguish by time correlation. Here, *Parallel* means that a user can use multiple modalities simultaneously, whereas *Sequential* allows a user only to use them after one another.

Where some might even consider the *Exclusive* method to not be fully multimodal, most researchers agree that the *Synergistic* method is ahead of the other three methods [62], or even that it subsumes the others:

"Technically, synergistic systems subsume the other three classes of multimodal systems." [43]

Since the fusion engine methods are considered very important, there were a lot more models and concepts introduced in the past (see figure 3.3). Furthermore, some researchers argue that the establishment and success of multimodal interfaces relies very much on good methods and even though the market is ready for such interfaces, the fusion engine methods need to be further improved until there will be *"...reliable and usable systems."* [29].

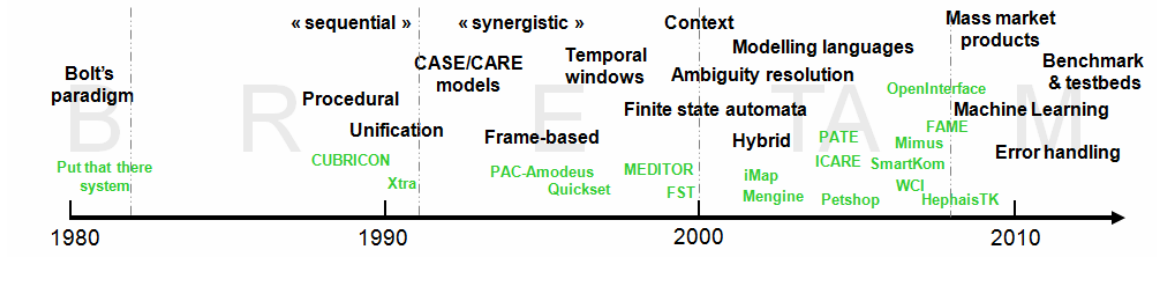


Figure 3.3: The evolution of fusion engine methods since 1980. [29]

As follow-up on their fusion engine methods, Courtaz and Nigay further introduced the CARE-properties (*Complementary, Assignment, Redundancy, Equivalence*) [21] which describe global relations between multiple modalities (see table 3.2). These are also well-known and play an important role in the planning phase of multimodal interfaces.

Complementary	Modalities are <i>complementary</i> if they are to be used together to reach a target state. This includes that the particular state cannot be reached with only a single or a subset of the available modalities.
Assignment	A single modality is <i>assigned</i> to a single target state if no other modality can be used to reach that particular state.
Redundancy	Multiple modalities are defined as <i>redundant</i> if each single one of them can be used to reach a target state with the same time effort (here, <i>equivalence</i> has to be applicable as well).
Equivalence	Multiple modalities are considered <i>equivalent</i> if one of them has to be used to reach the same goal (but the modalities differ in meanings of effort).

Table 3.2: The CARE-properties [21].

3.3 Cognitive Load

Reducing cognitive load on the user is one more important aim of multimodal interfaces [10]. Researchers involved in NUIs argue that even though the computer sciences of the last years have made large portions of functionality available to users, the capabilities of the human brain remain unchanged and thus, giving clear limitations to the load that can be used in novel interfaces, devices or applications [20]. Cognitive load is therefore another very important aspect to keep in mind while developing multimodal interfaces.

"Given the complex nature of users' multimodal interaction, cognitive science will play an essential role in guiding the design of robust multimodal systems." [49]

Too high levels of cognitive load can cause a user to get frustrated, stressed or even lose control over the situation [20]. A major aspect that plays a role in cognitive load is the learning effort: If it takes a user too long to learn how to use an interface, he or she is more likely to get frustrated and decide that the interface is not something he or she would like to continue using.

In the end, users should spend more time playing with the interface than learning how to use it (figure 3.4).

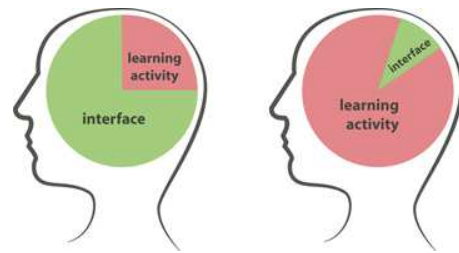


Figure 3.4: How long do users require for the "learning activity"? The less, better. [52]

Thus, developers have to think of strategies how to achieve minimal levels of cognitive load in NUIs. One possibility is to reduce the total amount of commands a user has to memorize. This can be achieved by using the same commands for similar tasks in multiple situations. Another possibility that is more to the point of NUIs is to use gestures or commands the user already knows from other areas such as HHI. Both ideas can be achieved at the same time with MMITs since *"...if we combine the modes in a complementary fashion, the set of interactions remains the same as either single modality, yet their respective vocabularies are simplified, easing cognitive load."* [30].

Here, the synergistic fusion method is argued to be the most difficult in terms of reducing the cognitive load:

"The design of multimodal systems that blend input modes synergistically depends on intimate knowledge of the properties of different modes and the information content they carry..." [49]

3.4 Models and Frameworks

Other frequently referenced and discussed topics around MMI include frameworks, descriptions, notations and modeling [9, 8, 41, 19, 47, 24]. Similar to the *Unified Modeling Language* (UML) [46] that is widely used in computer science, these notations are used to improve the development process of multimodal interfaces. These frameworks could, for instance, help the development of an interface by providing an easy tool to create an effective FSM. This would grant the developer the advantages described in section 3.1.

"Analyzing highly abstract and incomplete models early in the development cycle is critical because software designers make most fundamental design decisions during this stage." [47]

Figure 3.5 shows an example of such a notation. The in this case used framework is called *Notation for Multimodal Interaction Techniques* (NiMMiT) and it was introduced by Joan De Boeck [6].

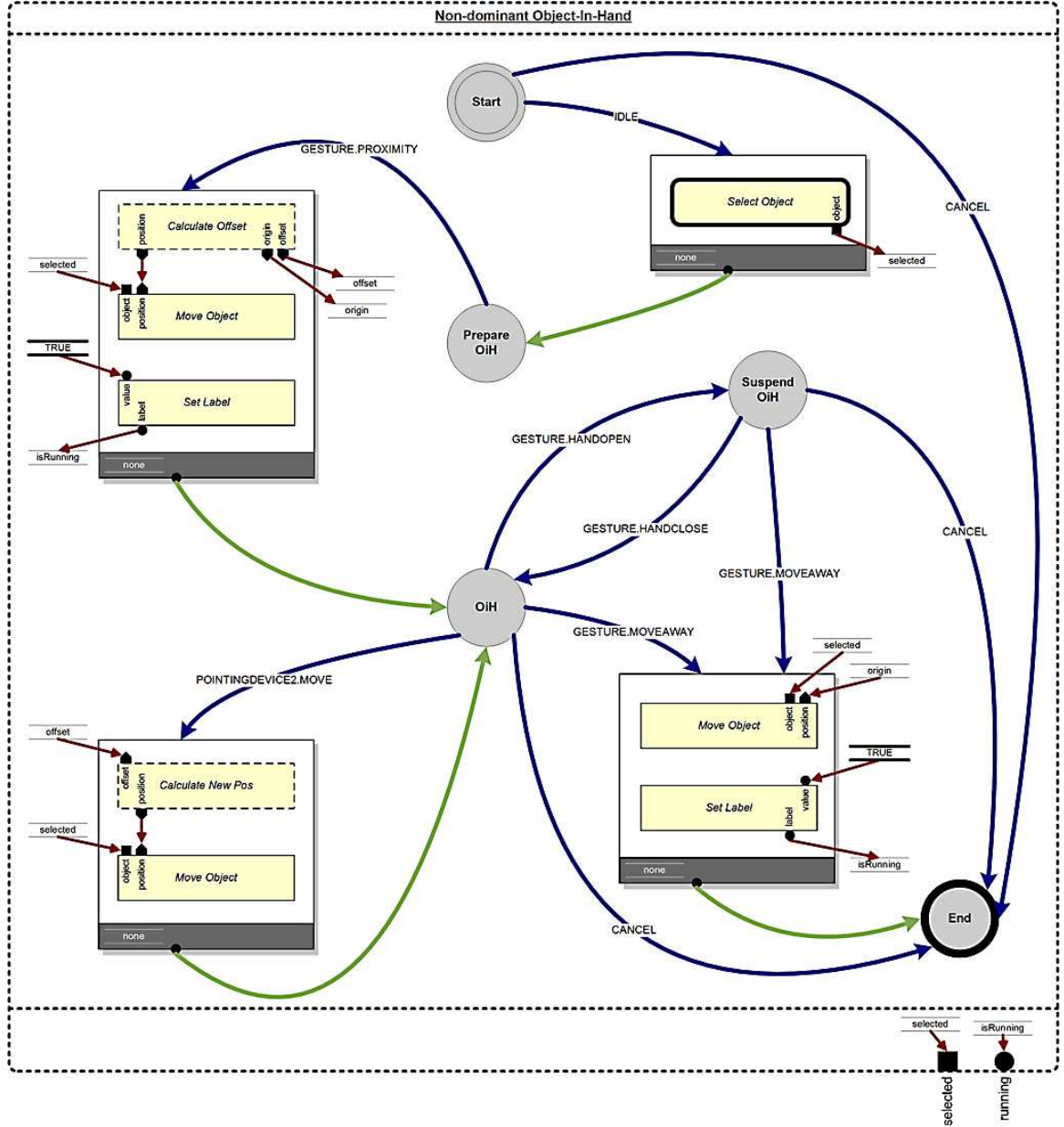


Figure 3.5: An example of the NiMMiT Framework featuring selection and manipulation of an object [6].

The application figuratively described in figure 3.5 features object selection and manipulation interaction tasks. Here, the circular shapes are states where the application waits for input. The blue arrows are triggered when a user starts interacting and are followed by in squares described calculations the application has to do step-by-step in reaction to the users input. The green arrows are pointing at those states, the application reaches after the calculation is done. At last, the red arrows indicate set or get methods of variables, that are shown at the bottom of the diagram. The term OIH means *Object in Hand* and indicates if a user has successfully selected an object.

3.5 Previous Work

The research field of multimodal interfaces is not a new one, it has been there for quite a few years now. Probably one of the most often referenced and discussed interfaces was introduced by Richard Bolt in 1980 [7]:

"Probably the best-known multimodal technique is the famous "put-that-there" technique [...] users can perform actions by combining pointing with speech. Many others have used the same combination of gesture and speech." [10]

It is referenced in many surveys or state of the art reports that have MMI as one of their topics [62, 24, 48, 29, 30] – and not without a reason: The *put-that-there* interface was one of the first interfaces that used exclusively MMITs and that used a synergistic fusion engine. The interface featured speech and pointing gestures as input and a map with drawn shapes on it on a large display as output (see figure 3.6). Users were sitting in front of the large display and were able to move, alter, delete or create shapes by pointing to a location and specifying the type by speech commands. The pointing gestures were recognized by a device worn by the user around the wrist (not visible in the picture).

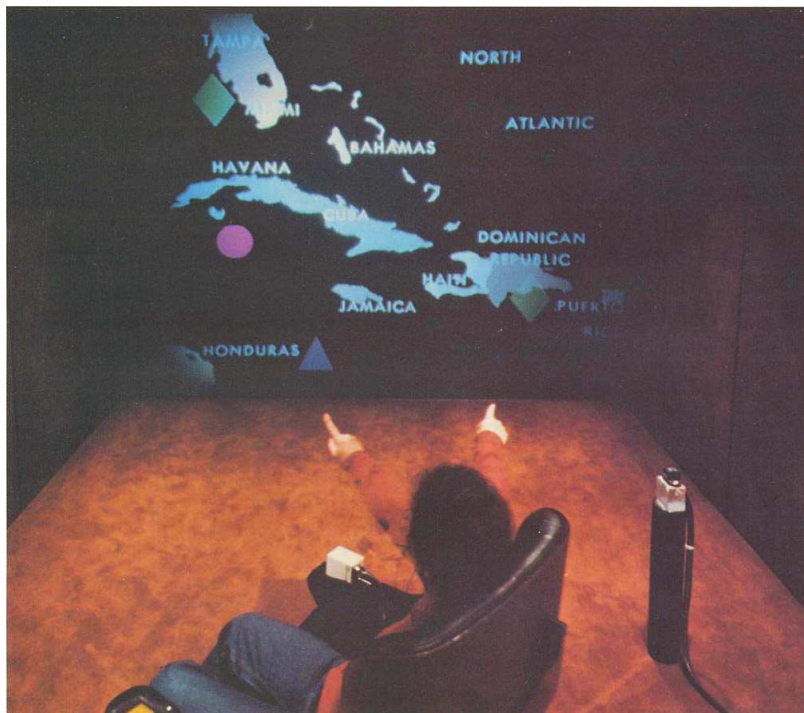


Figure 3.6: A photo of a user who is just using the famous *put-that-there* interface by Richard Bolt [7].

A more recent (2013) interface that uses MMI with seismic interpretation was introduced in a master thesis by Ömer Genc with the topic *"Novel Pen & Touch based Interaction Techniques for Seismic Interpretation"* [37]. The interface is meant to replace the usual mouse and keyboard interaction used by seismic interpreters in the oil and gas industry by a novel combination of pen and touch based input on a desktop-sized display (see figure 3.7). It uses a synergistic fusion concept as well since the user is simultaneously using pen and touch input to perform seismic interpretation tasks.

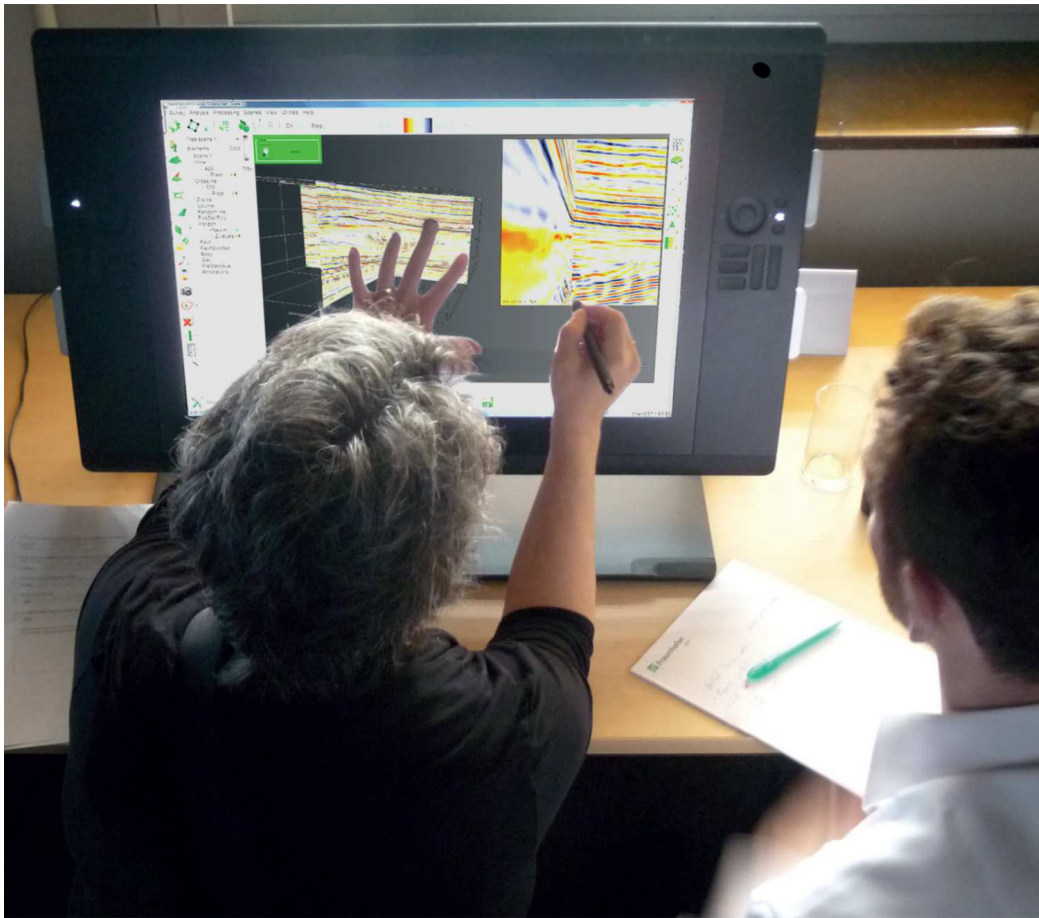


Figure 3.7: A photo of the interface with pen and touch based interaction introduced by Ömer Genc in 2013 [37].

Since the seismic data is often reduced to 2D (see section 2.3), this novel IT is very well suited for the circumstances:

"Although the 3D visualization tools of seismic software packages offer great potential for manipulation and interpretation of the data, the common workflow for interpretation often relies on using 2D seismic sections..." [37]

Along with the interface, a user study was performed to determine if seismic interpreters would prefer the novel interface over the usual desktop workstation. The results were very much in favor of the novel interface since *"...all participants preferred the pen and touch interaction over the conventional desktop interaction"* [37], proving that this novel interface could in fact replace the desktop workstation of seismic interpreters. Yet the interface has one final drawback: Due to the nature of the display, which is capable of recognizing pen and touch input simultaneously, a user is unable to place his or her hands on the display while interacting with the interface. Instead, the user has to hold his arms up the entire time of the interaction (see figure 3.7). Thus, some kind of palm rejection is required which is able to distinguish between intentional and unintentional touches by the user.

"From a technical point of view, the palm rejection is still a challenging and unresolved task. When using both hands and distinguishing between touch and pen input, the resting palm of the pen hand may be interpreted as intended touch and cause interference with the interaction." [37]

04

MULTIMODAL USER INTERFACE FOR SCIENTIFIC DATA VISUALIZATION

4. MULTIMODAL USER INTERFACE FOR SDV

This chapter covers a complete overview of the system that includes the interface and represents the practical part of this thesis. It should provide an explanation on how such a system could be implemented and will be evaluated with a user study in chapter 5.

4.1 Requirements

The requirements of the system can be divided into two subparts: The SDV part and the MMI part.

4.1.1 Scientific Data Visualization Part

Most importantly, the application needs to visualize some kind of scientific data in some form. Since it is not important what kind of data or from what field the data comes from, the in this thesis used example dataset of seismic data fully suffices. In what form the data is visualized, on the other hand, is much more important: Seeing as the data is being reduced to two-dimensions (as explained in chapter 2), it is logical that scientists have become somewhat familiar with working with the data in form of sections. That, and the fact that no natural interaction device is fully stable nor provides a perfect interaction paradigm nowadays, causes the visualization to require both three-dimensional and two-dimensional representation of the data. Thus, declaring volume- and section-based rendering techniques a necessity.

The choice for the used visualization tool in this work fell on a volume renderer that was developed at Fraunhofer IAIS. Its configuration is completely aimed at giving a user a good visual feedback on what he is currently seeing or doing and it is explained in section 4.1.2.

In terms of display size and type, applications featuring SDV can vary a lot. From desktop-sized solutions without stereoscopic capabilities [37] over large projectors [1] to fully immersive stereoscopic setups that surround the user [54], everything was represented in the past. In the end, it depends on the type of interaction and the type of data that is being visualized. Seeing as it is very attractive and natural [14] to have an interface that does not require the user to be attached to any device, the choice for gesture recognition fell on an optical sensor which in turn needs large spatial capabilities (the Microsoft Kinect [38]). Two more reasons to use a large display with stereoscopic capabilities include the higher levels of immersion and the assumption that users feel more comfortable having more space to move when interacting using gestures. Thus, the system requires a large display with high levels of immersion (e.g. through stereoscopic rendering). Some interpretation tasks (e.g. horizon picking [23]) require high levels of precision that neither gesture recognition nor speech recognition (or any other ITs that could be used in the spatial context of the large display) could grant the user, however. To compensate for this fact, the system should be expandable in a way that allows smaller displays with ITs that grant more precision to be added to the system with little effort.

4.1.2 Volume Renderer

In the volume renderer, the complete dataset is represented by a rectangle, where the extreme values of the data in all three dimensions define the size and scale of the rectangle. Within the rectangle, a scientist then has the ability to create either a section, which just is a two-dimensional plane limited to the bounds of the rectangle, or a smaller rectangle which can be understood as a kind of lens, making data inside of the smaller rectangle visible. Sections or the *"volume lens"* will from now on be classified as *"data assets"*. The entire rectangle will from now on be referenced to as *"Volume"*. If no data assets are created, all of the data is invisible and the rectangle is empty.

The outer bounds of the rectangle will always be represented by white lines, whereas the outer bounds of the volume lens will be represented by purple lines (see figure 4.1). The sections can be further classified into three groups – one for each axis. Each section is surrounded by lines in the particular color of their corresponding axis, where the x-axis is represented in red (*"Inline"*-section), the y-axis in green (*"Crossline"*-section) and the z-axis in blue (*"Timeline"*-section) (see figure 4.1). The sections are referred to as inline, crossline and timeline respectively.

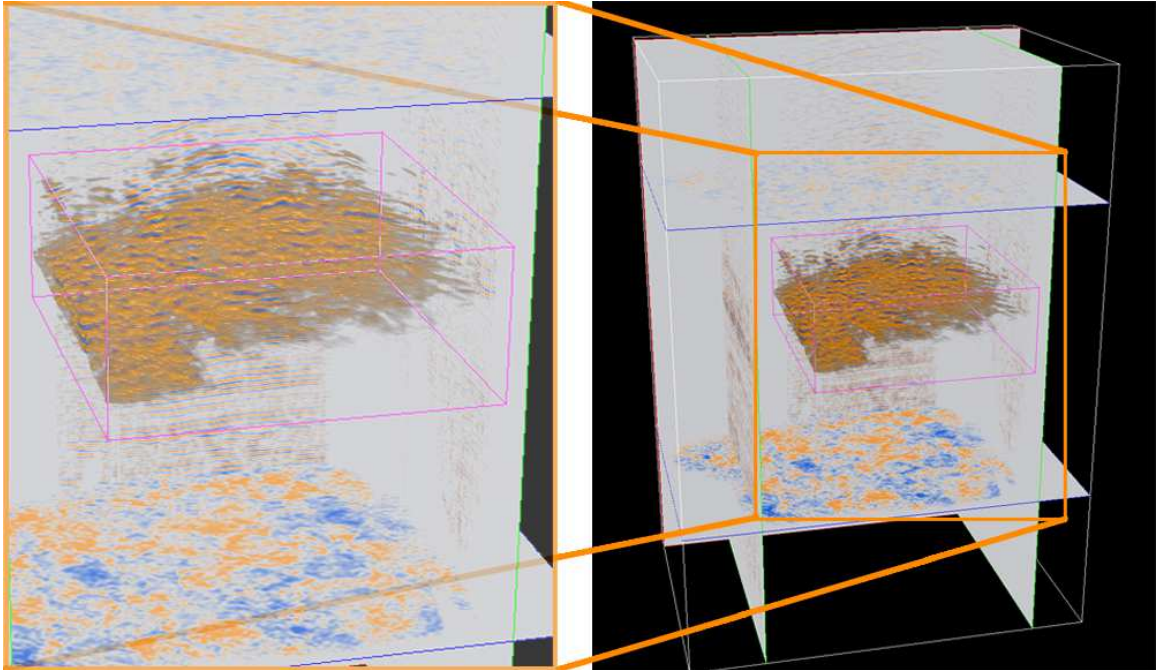


Figure 4.1: A screenshot of the volume renderer developed at Fraunhofer IAIS.

Even though the user should have a fully natural experience which would in turn mean to have as many degrees of freedom as possible, the best practice is sometimes to *"Reduce degrees of freedom when possible."* [10] Thus, both the lens and the sections can neither be moved outside of the bounds nor be rotated, so the user does not get lost or confused. The origin of the coordinate system is located at the front bottom left, so all calculations take place in positive values as the coordinate system is right-handed. At last, the volume-based rendering method is a ray-casting algorithm combined with depth buffer to enable occlusion of the geometry.

4.1.3 Multimodal Interaction Part

As explained in chapter 2, scientific data is most often very large and thus, requires powerful graphics hardware to still be able to render it in real-time. But aside from the rendering effort that arises from the nature of the data, the interaction devices used by the system require computation time as well. A single device would probably not cause any issues, but depending on how many devices are combined and how many devices or modalities the user can use simultaneously, the computation time of those can become an issue by, for instance, causing delays. Delays in a real-time environment can create immense problems as the user may try to redo a command because it is not recognized immediately, causing the system to receive the same command twice in a row after the delay has passed. It would also cause the interface to not seem natural because a user would always have to wait a moment until the system reacts. A solution is to attach the interaction devices to an extra machine and connect it via a network, making a network and the communication between multiple machines a necessity.

This would also cause the system to get easily expendable since a developer can attach as many machines as wanted. However, it also reveals another two requirements: First, if multiple machines send message over the network, the best practice is probably to create a server where all messages land classified in some way instead of simply sending all messages to the main application. Second, seeing as the interface may temporarily disable modalities or devices due to the user being in a state where they are not needed (since it is a FSM, as described in section 3.1), all machines would need to be able to send and receive messages from and to the server.

Therefore, the system is divided into multiple applications running on different machines connected via a network.

At last, as the fusion engine needs all interactions from all modalities at one point, there has to be some kind of management, where all interactions land and the combination techniques can be implemented before the system reacts to the interaction.

4.2 General Approaches

This section generally describes the idea behind the interface, how it was developed and what theoretical approaches have been used.

4.2.1 Interaction Tasks

Interaction task need to be specifically defined: Why does the user use the system? What can the user do with the system? What are the systems aims? The in the in this work used interaction tasks can be classified into three groups:

Navigation

Seeing as the data is visualized in a three-dimensional way, the user needs to be able to navigate around the data. Specifically, users must be able to reach every 3D position they desire. Here, it is especially important that the user does not get lost in the 3D space while trying to navigate through the scene. Thus, a reduction of the degrees-of-freedom is appropriate. This is achieved by only allowing the user to zoom and rotate, where the rotation is limited to two angles – pitch and yaw (in figure 4.2 they are represented by "*Azimuth*" and "*Elevation*"). The pitch (vertical) angle is further limited to 90 degrees, (positive or negative), so the user does not end up upside down. Using the radius and two angles, every 3D position around the center of interest can be reached (see figure 4.2).

Object Manipulation

The user requires a possibility to control the data assets. Object manipulation tasks include selection of data assets, translation of sections, translation and scaling of the lens and hiding or deleting of data assets. Since the deletion of data objects is object specific (first choose the object and then delete it), it is classified as object manipulation whereas object creation is classified as system control task.

System Control

The user further needs to be able to create objects via system control. For instance, the user must be able to create the different types of sections and show or hide the lens.

To provide the user with a fully natural experience, the system should generally be designed in a fashion where the user does not have to rely on any kind of GUI at all. The use of a GUI element such as a menu or button could negatively influence the evaluation results as the interaction with three-dimensional data requires up to six degrees-of-freedom whereas the usual GUI is designed only for two [10].

Where system control tasks are usually performed by the use of a menu, in the present work those tasks should be accessible by natural ITs as well. Commonly, speech recognition is considered to be best suited for system control tasks [10]. However, using speech commands can only be as effective if the user does not have to memorize them, but instead they are obviously designed (e.g. "delete this object" or "show the volume lens").

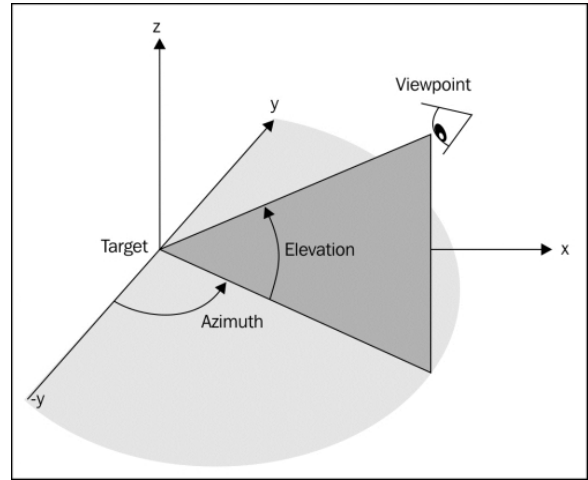


Figure 4.2: Only two angles and the radius are needed to reach every position in 3D-space. [56].

4.2.2 Interaction Techniques and Modalities

The combination of speech and hand gestures has been used in a lot of interfaces [30, 10, 7]. The reason for that is probably that speech recognition is best suited for system control tasks, whereas hand gestures are one of the most precisely evolved interaction methods of humans [58] and because they both are fairly easily implemented. The choice for the used modalities in the system therefore fell on speech and hand gestures, although it should at all time be possible to add new modalities with little effort.

As for the ITs, they need to be distinguished in regards to which fusion engine method is used. For instance, object movement could be implemented as follows: If no combination of modalities should be used, it would certainly be the best practice to implement a simple grabbing metaphor, whereas if a synergistic fusion engine method is used, it could be realized by a similar paradigm to the put-that-there [7] interface. At this stage, it is impossible to know which combination of modalities for which task has the best user acceptance or is easiest to learn. Therefore, an evaluation with multiple users has to take place to determine such characteristics (user study). This user study is described in more detail in chapter 5. For the evaluation of the system, two different fusion engine methods were implemented for different tasks. Since the evaluation targets the combination of modalities and should not be influenced by the complexity of a task, the navigation task will not be evaluated using different fusion engine methods.

Unimodal/Concurrent

The evaluation if a user would prefer to interact in a unimodal way is of course important for the evaluation of the entire system. Tasks, that are object specific such as object selection, object manipulation and object deletion are tasks that can be performed unimodally in the system by just using hand gestures. Object creation, on the other hand, is a system control task as the user needs to specify what type of object he or she wants to create. If implemented using only hand gestures, the user would need a menu or atleast buttons. It is therefore the best practice to implement it only using speech. This means that the interaction could be considered multimodal since the user is able to use both gestures and speech. However, since the user is not forced to use multiple modalities to reach one desired goal, the modalities are not combined or integrated. The interaction could therefore still be considered a concurrent fusion engine method: A user can use gestures and speech in parallel, yet the modalities are not combined.

Synergistic

Using the synergistic fusion engine method, users have to be forced to use multiple modalities simultaneously, otherwise they might not interact multimodally at all [48, 62]. Thus, ITs for simple tasks such as object manipulation must be implemented in a fully synergistic way as well. Thus, with the synergistic model, the user can perform all tasks of the system: Object creation, object selection, object manipulation and object deletion.

Looking at the ITs listed above, there is already an advantage in using the synergistic model visible: The unimodal model seems incapable of realizing system control tasks, whereas the synergistic model is capable of realizing every task.

4.2.3 Finite State Machine

As described in section 3.1 of this work, NUIs are often designed as a FSM, where the number of states should be as little as possible. The used interaction paradigms in this system are described using FSMs below.

If just the unimodal capabilities are available (speech and gesture are not combined), the system consists of a total of three states (see figure 4.3):

First, in the *Start*-state, the system is idle until a user steps into the tracking area of the optical sensor – which triggers a state transition to the *Idle*-state. In the *Idle*-state, the user can either create new objects using speech, which would cause the system to return to the *Idle*-state after creation, or to select an object using a grab gesture, which would cause a transition to the *Select*-state. In this state, the user can perform three tasks: Deleting, scaling or translating the selected object. Scaling and translating would cause the system to remain in the *Select*-state as long as the user keeps his hand grabbed, whereas deleting an object causes the object to be discarded and the system to return to the *Idle*-state. One disadvantage is immediately evident: The entire time the system remains in the *Select*-state, the user has to hold his hand both grabbed and up in the air.

In the FSM diagram for the synergistic model, on the other hand, the *Select*-state is completely obsolete since the user performs object selection and manipulating with a single command (see figure 4.4). For instance, an object is deleted by saying the phrase "*delete this*" while specifying which object using the 3D-location of the handpointer described in section 4.2.4. The object is therefore selected and afterwards deleted with the use of a single state transition, where the system ends up at the same state as before and the user can take his hand down again right after finishing the phrase.

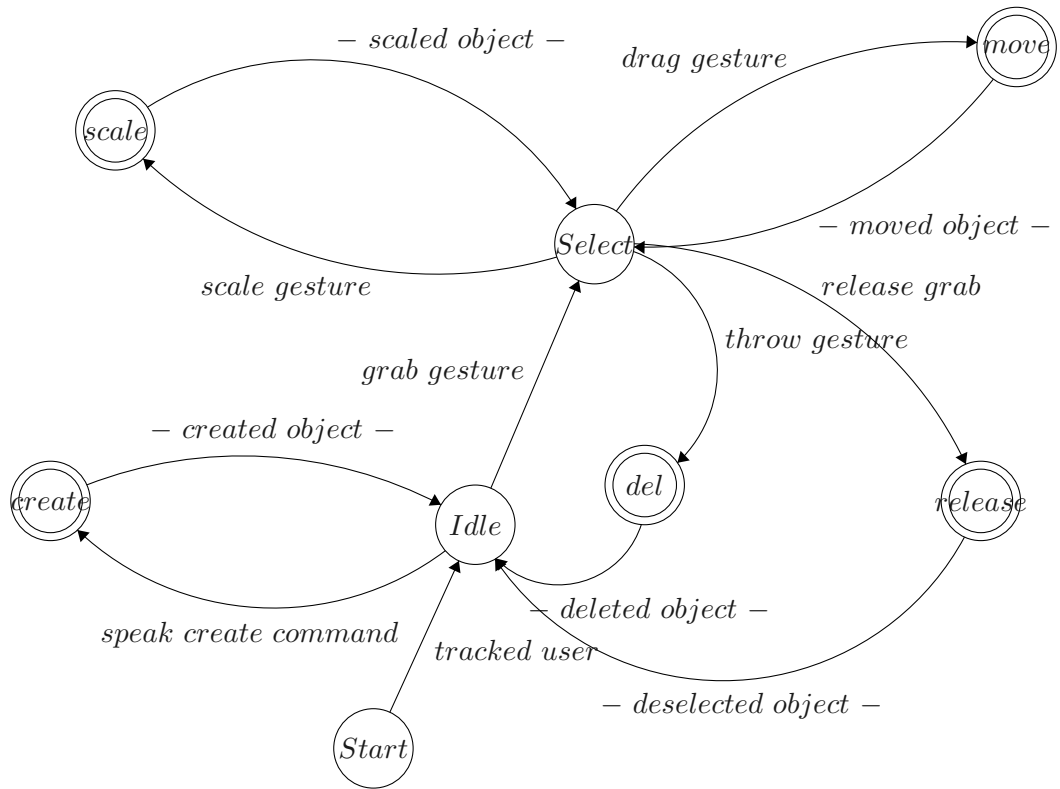


Figure 4.3: A FSM diagram of the unimodal ITs. Designed with the FSM Designer by Evan Wallace [65].

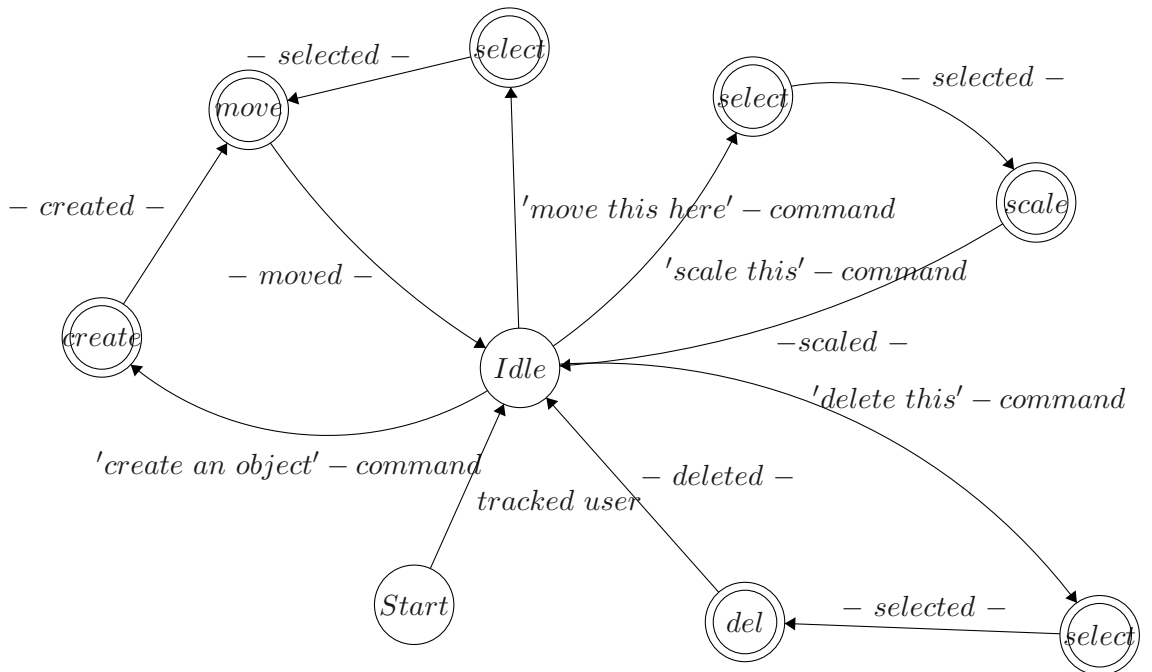


Figure 4.4: A FSM diagram of the multimodal ITs. Designed with the FSM Designer by Evan Wallace [65].

4.2.4 Handpointers

When interacting with 3D content, users always need to know where they are interacting at the moment [10]. In addition, it could be a great advantage to let the user know what he or she is currently doing and with which object he or she is currently interacting. This way, errors are minimized as the user immediately notices false recognitions (or not recognized commands).

For this purpose, the so-called *"handpointers"* were added, these are simple 3D geometries (spheres in figure 4.5) which represent both of the users hands and show their current position and state. The advantage of being able to see where in 3D space the user is currently interacting may be evident because the user has to know what object he would select when he or she executes the selection command at any point, however another big advantage is to show the state of the hand: If the gesture recognition fails to recognize something, the user immediately notices it and can react to the circumstances (e.g. by redoing the gesture, if not recognized).

Thus, the handpointers are able to show whether a users hand is currently recognized as grabbed or not by changing their color. In figure 4.5, the spheres have changed their color dependent on what the user is currently doing.

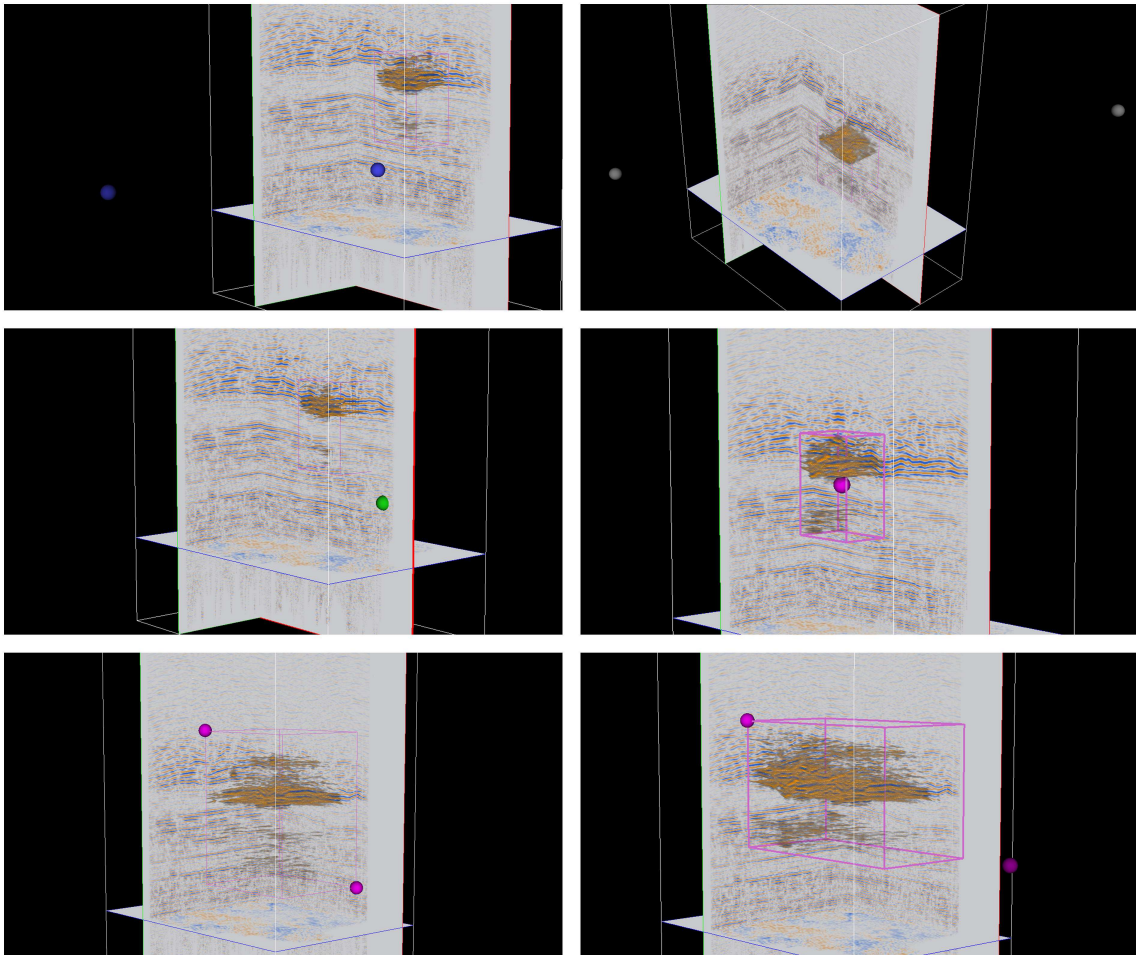


Figure 4.5: The spheres that represent a users hand position and state.

Adding not only the information if a users hand is grabbed but also what the user has currently grabbed is fairly easy added since the data assets are already color-coded.

The handpointers therefore change their color in correlation to the object the user is currently holding in that particular hand – **blue** means the hand is not grabbed (the user can start the interaction but is currently not interacting with any data asset, top left frame in figure 4.5), **white** means the entire volume in hand (the user is able to rotate or zoom the view, top right frame in figure 4.5), **green** means a section in hand (the user is able to translate or delete the section, center left frame in figure 4.5) and **purple** means the lens in hand (the user is able to translate or hide the lens, center right in figure 4.5). The user is also able to interact using both hands. For instance, in the bottom left frame of figure 4.5, the user is scaling the lens using a two-handed IT.

At last, adding the information if a user’s hand is currently inside of the bounds of the volume could be a great advantage because interacting with three-dimensional content is complicated enough as it is (even though the system has stereoscopic capabilities). Thus, the brightness of a handpointer is reduced as long as it is outside of the bounds of the volume (in the bottom right frame in figure 4.5, a user is scaling the lens using two hands, where one of the hands not inside the volume).

This way, users always know what actions they are currently able to perform – for instance, what object they would delete if the deletion command is entered.

4.2.5 Feedback

The usage of proper feedback in the proper format (e.g. audio feedback for audio input) has already been discussed in chapter 3.

Seeing as the system features two modalities as input (gesture and speech), the ideal output formats would be sound and visual respectively: If a user interacts using his or her hands (gestures), the system responds with something visual, whereas when the user interacts using a speech command, the system responds with an audio-clue to acknowledge understanding.

Visual feedback includes the highlighting of objects that can be selected as soon as a handpointer gets near enough and the change of color and brightness of the handpointers as described in section 4.2.4.

Furthermore, if users signal that they do not want to interact at the moment by lowering their hands or leaving the tracking are the handpointers get invisible, causing a user to immediately notice when the tracking hardware fails at any time.

Sound feedback is given by the rendering machine when a command has an effect on the system. This means that even if a command is recognized, the feedback is only triggered if the command is even possible at that time. For instance, if the user does not point at an object at the time of saying the phrase *“delete this”*, the system does not react to it since the user did not specify what object to delete and thus, not triggering any audio feedback.

4.3 System Overview

This section covers an overview of the used devices and other hardware as well as which machine is used for what and how they are connected. The applications that runs on the specific machines will be described in section 4.4.

4.3.1 System Architecture

The entire system is divided into multiple machines (as explained in section 4.1). In the current state, there are three machines used: The *"Rendering Machine"*, the *"Tracking Machine"* and the *"Remote Workstation"* (see figure 4.6). Devices such as the Microsoft Kinect [38] or the bluetooth-headset used for speech recognition are connected to the tracking machine, whereas the large display is connected to the rendering machine. The main application including the volume renderer is running on the rendering machine, the message broker and the tracking framework are running on the tracking machine and the remote workstation was just installed for developing purposes.

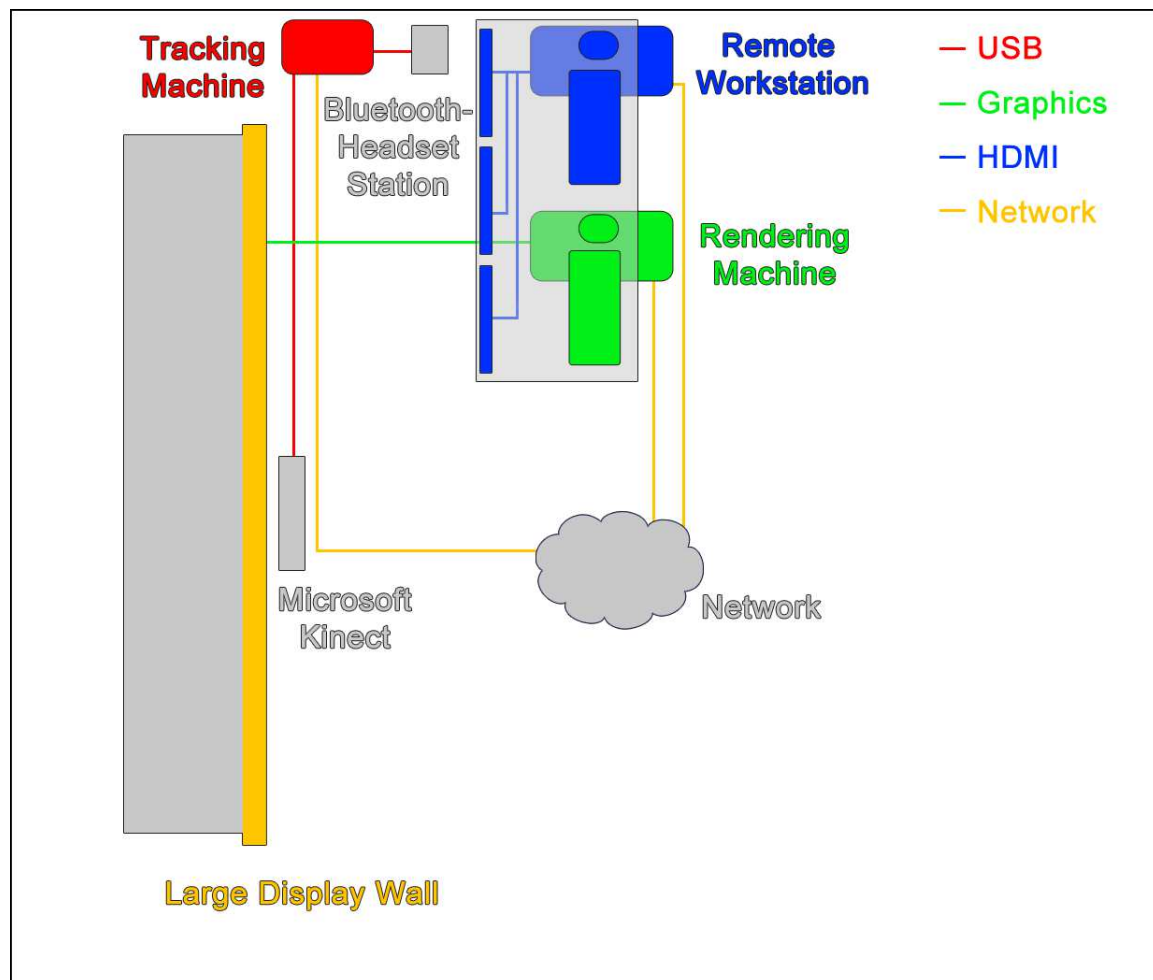


Figure 4.6: The system architecture.

All three machines are connected in a LAN, although the tracking machine does also have wireless capabilities so that more machines could be added with or without cable.

4.3.2 Remote Workstation

It is a good practice in a VR application to have an environment where a developer can develop comfortably using the mouse and keyboard and simultaneously test the system on a large, immersive display.

The remote workstation fulfills that purpose: Using a VNC software (TightVNC [25]), a developer is able to see what is currently being shown on the rendering machine using three HD-monitors (for the same resolution as the large display). Interacting with the rendering machine is thus possible with keyboard, mouse and monitors on a usual desk.

4.3.3 Tracking Machine

All interaction devices used by the system are connected to the tracking machine, its purpose is to process the tracking data and then send it to the main application via the network. The tracking machine is also the message broker of the network, so any additional machines would send their message to the tracking machine first, where they are then send along to the main application on the rendering machine if wanted. The advantage of using the message broker is that specific data can be deactivated on the tracking machine, if the data is irrelevant at the time. And, the tracking data could of course be reused in another application.

Currently attached devices include the Microsoft Kinect [38], a bluetooth headset, a display and speakers for audio feedback. During runtime, the user is able to see a debug window on the display of the tracking machine which further gives the user visual feedback about the current situation – for instance, a user could see if he or she is outside of the tracking area of the optical sensor. Overall, the tracking machine does not require much graphics performance but instead a good processor for processing tracking data of interaction devices.

4.3.4 Rendering Machine

The currently used large and immersive display is a display wall that consists of a total of three Barco LED panels of type *OLS-721* [3] with HD resolution each (5760 x 1080 in total). The used stereoscopic rendering method used by it is the so-called "*quad buffering*"-technology from NVIDIA [44]. Furthermore, an NVIDIA Quadro Plex 7000 [45] was added to the rendering machine since the resolution together with the doubled framerate (stereo) and the ray-casting rendering algorithm require a lot of graphical performance to perform smoothly in real-time.

4.4 Software Architecture

This section gives a complete overview of the software that was developed alongside with this thesis including an overview, used frameworks and libraries, both applications and the message broker.

4.4.1 Overview

The application including the volume renderer and the fusion engine methods (running on the rendering machine) will from here on be referenced as *"main application"* (section 4.4.4), whereas the application on the tracking machine including the gesture and speech recognition will be called *"NIFramework"* (*Natural Interaction Framework*) (section 4.4.3). Both applications are written in C++ and start two extra threads at start-up for the network communication – a consumer and a producer. The library used for the message broker is ActiveMQ [59], which will be explained in more detail in section 4.4.2.

In figure 4.7, the connections between the applications is figuratively described. The ActiveMQProducer and the ActiveMQConsumer are the extra threads all applications start at start-up for the network communication, the consumer receives and the producer sends messages. There is no limit on how many producers and consumers an application may start. That way it is possible to create multiple consumers which receive messages from different brokers or multiple producer which send messages to different brokers. The message broker (*"ActiveMQ server"*) is a separate program running on the tracking machine. Additional machines providing additional functionality could be easily added to the system as they would just require to start sending properly formatted messages to the broker that the main application can understand (figure 4.7).

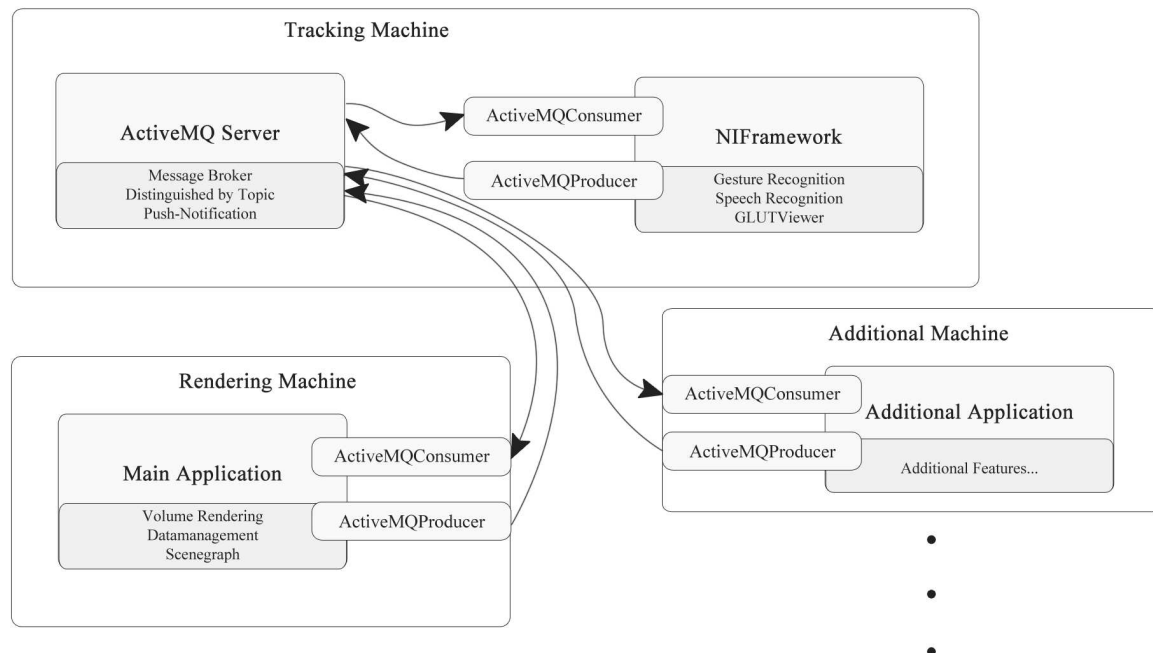


Figure 4.7: The software overview. ActiveMQConsumer and ActiveMQProducer are extra threads started for the network communication.

4.4.2 ActiveMQ Connection

ActiveMQ is a java-based "...popular and powerful open source messaging and Integration Patterns Server." [59] The authors provide a C++ binding called "*ActiveMQ-CPP*" which is used in the system to send and receive messages to an on the tracking machine separately running java application ("*ActiveMQ Server*") which serves as message broker.

The server is configured to distinguish messages by topics. Here, the "*producer*" can send messages on specific topics and the "*consumer*" can register itself on specific topics and then gets notified when new messages under those particular topics arrive. Both producer and consumer are running in their own threads in the system so that they do not interfere with the applications if something goes wrong (e.g. slow network connection). Furthermore, both are configured to use or register themselves on only a single topic at a time since every topic should be handled by an additional thread to create a best possible result in computational terms.

All applications including possible ones in the future use the same so-called "*ActiveMQConnection*" library, which was specifically developed by the author of the present work. It is therefore not necessary to explain the module explicitly for every application.

The main class of the module entitled "*ActiveMQConnection*" is meant as a "*connection-factory*", where an infinite number of producers and consumers can be created. They are then saved in a proper form and to ensure a smooth shutdown of all threads, they are all terminated when the *ActiveMQConnection* is disposed. Furthermore, to be able to reference to a particular consumer or producer during runtime again, consumers and producers are to be assigned a unique URI at creation.

The two classes for the producer and the consumer are called "*ActiveMQConsumer*" and "*ActiveMQProducer*", however they will just be referenced as "*consumer*" and "*producer*" respectively throughout this work.

The consumer is defined as

```
class ActiveMQConsumer : public decaf::lang::Runnable
```

and is derived from the *Runnable*-class of the *ActiveMQ-CPP* library since it should be able to start its own thread and then apply itself as runnable to it. At creation, the consumer needs the URI of a broker including protocol and hostname, a topic and an instance of the "*MessageListener*"-class. The *MessageListener* is derived from a class with the same name out of the *ActiveMQ-CPP* library and is the base class from which all specific *MessageListener* must derive in order to be assigned to a consumer.

```
class MessageListener : public cms::MessageListener
```

Depending on what format the messages that arrive are in, a different *MessageListener* can be applied. At the current state the system provides a "*TextMessageListener*" which understands the *ActiveMQ-CPP* internal format "*TextMessage*" and an "*XmlMessageListener*" which can understand xml-formatted messages. The *MessageListeners* are meant to translate the messages of various formats and pass on the important information contained in the message to observers in the application. In the main application, for instance, an *XmlMessageListener* is defined which is registered under the topic "*gesture*" and can pass on the information to various observers for various gestures, depending on what type of gesture was recognized. Here, the *MessageListener* uses objects derived from the class "*InteractionInfo*" to store the tracking information and pass it on to the particular observers – for example, the "*ThrowGestureObserver*"-class receives its information in an object of the type "*ThrowGestureInfo*" (more details in section 4.4.4).

The thread of the consumer is not immediately started with creation of the object, instead the consumer provides a method for this purpose:

```
void ActiveMQConsumer::start()
```

The call of the `start()`-method causes the consumer to create a new thread with itself as *Runnable*, start it and then wait for the connection to the ActiveMQ server to be established within the thread (or, if no connection could be established, abort after ten seconds with an error message).

The opposite side of the communication (producer) is designed in a similar fashion. The producer is defined as

```
class ActiveMQProducer : public decaf::lang::Runnable
```

and is derived from the *Runnable*-class as well. At creation, the producer needs the URI of a broker, a topic and an instance of the *"MessageSerializer"*-class. The *MessageSerializer* is the complement to the *MessageListener*: For every message-format that should be send via the ActiveMQ server, a class that derives from the *MessageSerializer* must be created that is then assigned to the producer.

```
class MessageSerializer
```

The system provides *MessageSerializer* for xml-formatted and for *TextMessage*-formatted messages in the same way the consumer does for the *MessageListener*-class. At last, the thread is started by the use of the same method as the consumer provides:

```
void ActiveMQProducer::start()
```

4.4.3 NIFramework

The application running on the tracking machine is the so-called *"Natural Interaction Framework (NIFramework)"*. It was already developed at the Fraunhofer IAIS previously, yet it has been re-designed and improved by a major portion by the author for the purposes of this work. The general idea is to provide a framework where all SDKs from devices for natural interaction used by the system get combined and converted to a device and platform independent format, which is afterwards send via a network. Furthermore, the NIFramework creates a window using the GLUT-library [61], where tracking data is visualized for debugging purposes (figure 4.8).

All users that want to interact with the system first have to enter the tracking area of the optical sensor. Upon entering, a new user gets a unique ID assigned and the application starts to track him or her throughout the system. The tracking information for the users body provided by the optical sensor is then saved in an instance of the so-called *"skeleton"*-class.

```
class NI::Skeleton
```

The optical sensor that is currently used in the system is capable of tracking a total of 21 joints of a user (Microsoft Kinect [38]). The NIFramework further distinguishes between two types of skeleton (or users): Primary and secondary. The only command that has an effect when a secondary user enters it is a *"sign in"*-gesture which causes the performing user to become the new primary, whereas the primary user is in control of the system and can therefore enter all commands. There is only one primary skeleton existing in the system at any time. If new commands are defined (e.g. new gestures), the developer can define what type of skeleton can perform them by the call of a single method:

```
setValidSkeletonControlMode(NI::SkeletonControlMode::PRIMARY_SKELETON);
```

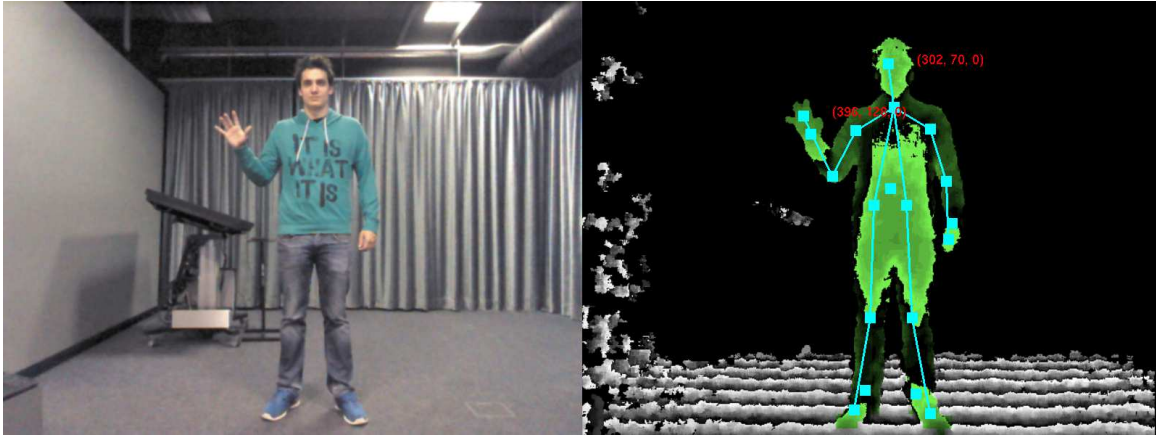


Figure 4.8: A screenshot of the *NIFramework*, where a user is currently interacting with gestures. The RGB frame (left) and the depth frame (right, with highlighted primary skeleton/user) provided by the Kinect for Windows SDK [38] together with a representation of the users tracked body (skeleton) rendered with GLUT [61].

The recognition of gestures or speech commands is designed by a detector–observer principle (figuratively described in figure 4.9). Every gesture intended by the developer gets an instance of a class that is derived from the *"NiDetector"*–class assigned. In these classes, the actual calculations based on the users movement to recognize a gesture takes place.

```
class NI::NiDetector
```

All detectors get saved in a list through which the application cycles every frame during the *"glut-MainLoop()"*–function of the GLUT–library, calling the *"detect()"*–method of every detector in the list (see figure 4.9). Aside from detecting gestures, the main–loop function gets the tracking information out of all devices and SDKs included and renders the window in figure 4.8.

The detectors themselves only calculate possible gestures for the type of skeleton they were assigned to (primary or secondary). If no gesture is recognized or no skeleton of the type the detector is assigned to exists, no extra calculations are being done. If, on the other hand, a gesture is recognized by a detector, it creates an instance of a class derived from the *"NiDetectionInfo"*–class. These classes are created for every gesture or other command as well and aim at storing the information that is important for the recognized command (e.g. hand position, movement velocity) and then passing them on to an observer.

```
class NI::NiDetectionObserver
```

The *"NiDetectionObserver"*–class is again the base–class for every observer for all possible gestures or commands. The information that the detectors calculated is thus passed on to the observers by the call of their *"notify()"*–method, providing the newly created *InteractionInfo*–instance as parameter. Therefore, the detector has to have a reference pointing to what observers to notify – for this purpose, observers are attached to detectors where they are saved in a list.

```
gripGestureObserver.attachTo(gripGestureDetector)
```

It is thus possible to have a single detector notify multiple observers. As soon as an observer is notified, it needs to fulfill the purpose for what the developer implemented it – in most cases, the observers send the information to a producer so it can reach the main application (see figure 4.9). Other purposes include, for instance, the change of the primary skeleton or to enable and disable other commands (push–to–talk).

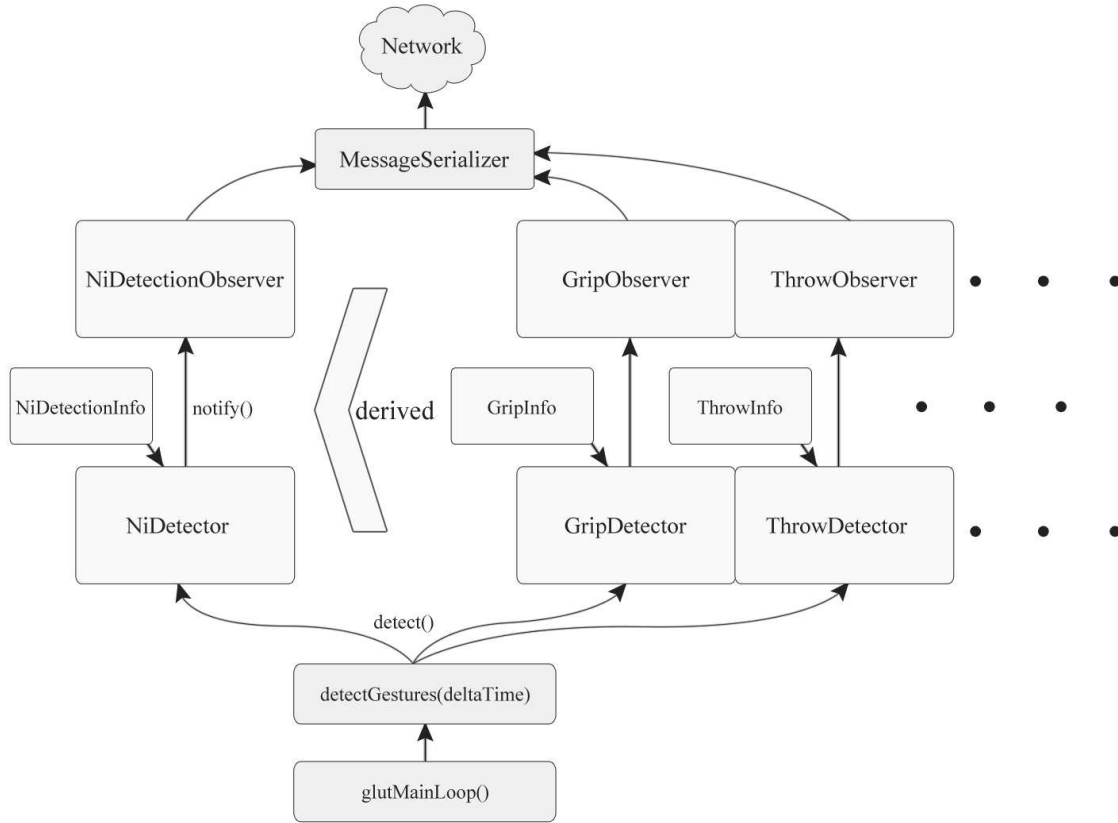


Figure 4.9: The detector-observer principle used in the NIFramework.

4.4.4 Main Application

The main application of the system is running on the rendering machine and includes the volume renderer as well as the fusion engine methods. It was explicitly developed for the purposes of the present work by the author. At the start, the application first loads the data that is to be visualized, then creates the scene including the volume renderer and establishes the network connection.

The class responsible for handling the scene is defined as

```
class SceneManager
```

and it includes the "*VolumeVisualizer*" object of the volume renderer and its parent "*TransformationMatrixNode*" object (a class of the OpenSceneGraph-library that is used as scenegraph [16]). It further manages creation and deletion of objects such as data assets or handpointers and keeps a list of all sections currently in the scene or volume. Sections are defined by two integer-characteristics: First, the type of the slice defined by the volume renderer through a simple enum and second, its position in the volume.

```
typedef std::pair<SeisViz3D::VolumeDataVisualization::LineType, unsigned int> Slice
```

The unsigned integer suffices as position definition due to the nature of seismic data: There is only a finite number of values recorded in every dimension which is why the visualization of such data is most effective if the values just get represented by integers starting at zero.

The information coming from the NIFramework is processed in a similar way using observers (see figure 4.10). The interaction information arrives at the consumer registered under the same topic as the producer of the NIFramework. It is then passed on to a MessageListener, which distinguishes depending on the information that the message holds (which gesture or command), then creates an object of the corresponding class derived from the *"InteractionInfo"*-class and afterwards notifies an observer providing the InteractionInfo as parameter. For this, the developer registers observers for specific types of input that arrives, where the type is specified in the message itself. The speech observer, for instance, gets registered for the type *"Speech"*:

```
xmlMessageListener.addObserverForType(&speechObserver, "Speech")
```

The messages that arrive from the NIFramework therefore have a type information in them, causing the MessageListener to forward the information to the particular Observer.

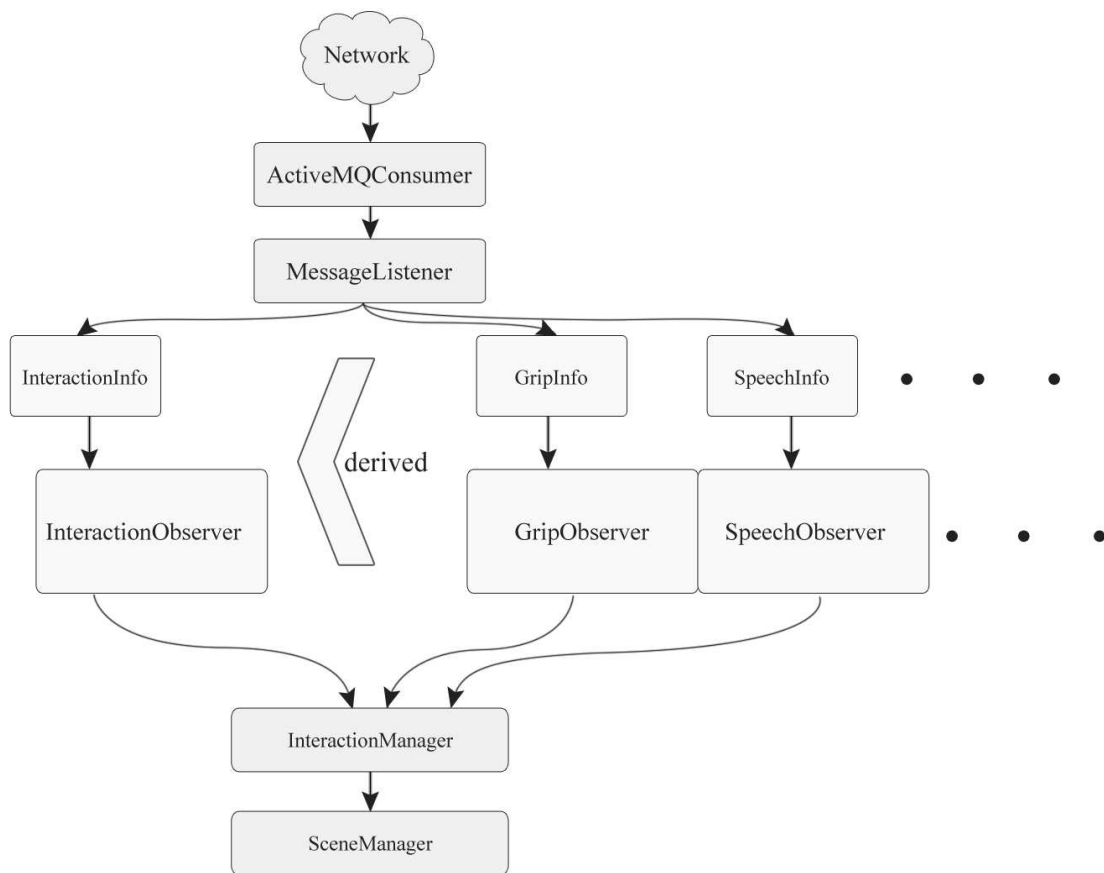


Figure 4.10: The principle functionality of the main application figuratively described.

Once an observer is notified, it processes the information contained in the InteractionInfo-instance and then reacts to certain events by triggering methods of the so-called *"InteractionManager"* (see figure 4.10). For instance, if a grab occurred, the observer lets the InteractionManager know that it should select the object that is currently near the hand position of the newly grabbed hand (in case the system is controlled in a unimodal way).

```
class InteractionManager
```

Instead of letting all observers handle the interaction separately, the `InteractionManager` class gets all the information in a single place and is thus providing the perfect place to implement the fusion engine methods. At this point there are two different types of fusion engine methods implemented in the system: Concurrent and synergistic. Concurrent means that even though the user is able to use multiple modalities in parallel, they are not used to reach one common goal and are therefore not integrated or combined. It could therefore also be considered just interacting in a multimodal way whilst having the choice of multiple modalities to do so (gesture or speech). The synergistic method has speech and gestures fully integrated. A user is therefore forced to use multiple modalities at once to reach a goal that could be reached just by using one modality in the concurrent method. Here, forcing the user is important since having a multimodal system does not necessarily mean that users will interact multimodally [48].

The way the synergistic fusion engine implemented in the system works is by combining commands over time: Usually, a user has to point to the object he or she wants to select, delete or move using his handpointers while saying the phrase *"select this"*, *"move this..."* or *"delete this"*. But what if a user says the phrase first and then points to the object? Or, the other way around, removes his handpointer from the desired object before finishing the phrase? And, lastly, what if the speech recognition takes a little delay or the hand tracking fails in the very moment the phrase is finished? Without an effective way of combination between the modalities, the interaction would just fail in these cases. Thus, the commands are saved for a particular time (e.g. five seconds) giving the system time to react to other parallel interactions. If a user says the phrase *"delete this"* while no object is currently being pointed at, the application saves the delete command until a user shows what object he or she means by resting on them for a moment with a handpointer. On the contrary, recently highlighted or selected objects remain saved by the application in case any object-specific commands arrive a few seconds later.

Commands that are not currently possible need to be discarded in the `InteractionManager` as well since the observers do not need to know what fusion engine method is currently used nor what commands are possible with it. If a user wants to start scaling the volume lens by saying the phrase *"scale volume lens"*, for instance, the `InteractionManager` firstly checks if the user is not holding any other objects in hand or if the synergistic fusion engine is even used at the moment.

To reach high levels of intuitiveness, another important fact to consider is that a user would most certainly expect the system to understand whole sentences. For instance, a user might aim at creating multiple objects at once by saying the phrase *"create a new inline here... and another one here... and a crossline there"* while specifying multiple locations during saying it. For these cases, the system understands phrases like *"and another one here"* or *"and a crossline there"* as separate commands that only trigger a reaction in the main application if the user entered a different command recently which defines what objects he or she wants to create.

Aside from the fusion engine methods, the `InteractionManager` manages the handpointers and their specific states. This has to be managed by the `InteractionManager` since the events that trigger state changes of the handpointers (change of color through object selection) depend on what fusion engine method is currently used. With the concurrent model, state changes are triggered by the user grabbing his or her hands, whereas with the synergistic model these are triggered through speech commands.

At last, the `InteractionManager` manages objects that are currently or have recently been in the users hands by saving the object type and a reference to a slice.

```
typedef std::pair<SceneManager::objectType, Slice*> sceneObject
```

The object type may specify that the user currently has or recently had the lens, slices or the volume in hand. If no slice is selected, no reference is saved.

Picking of 3D objects is handled in a separate class and depending on the 3D position of the hand-pointers, not by the shoot of a ray. That way objects that are occluded by other objects may get picked by the user without having to rotate the entire volume.

The general interaction metaphor used in the system is a virtual desk metaphor: The objects are available to users as if they were flying over a virtual desk in front of the user. Objects therefore get translated or scaled depending on the absolute position of the handpointers instead of using the velocity and movement direction.

In addition, navigation interaction causes the entire volume to get rotated or scaled rather than altering the cameras position and orientation – which in turn requires the positions of the handpointers to be converted to the local coordinate system of the volume if required.

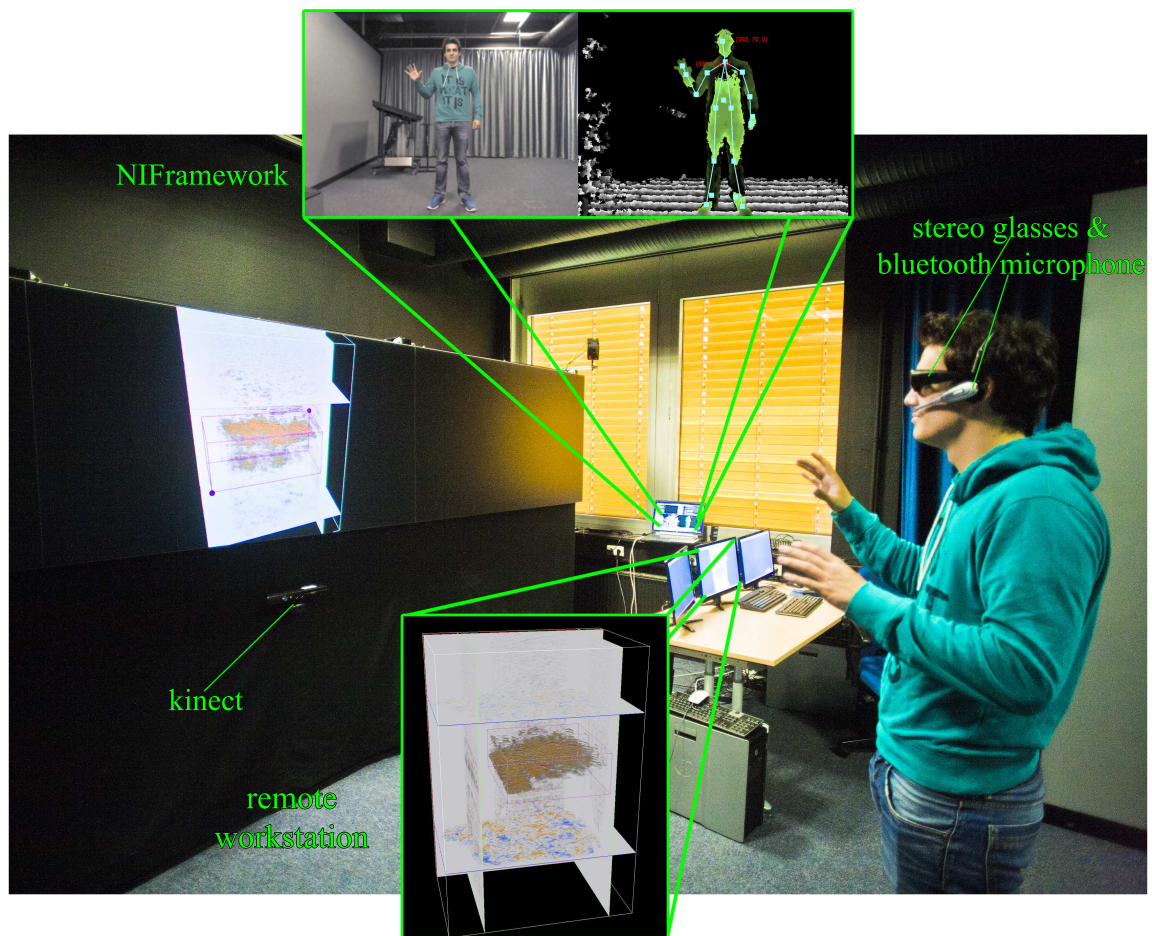


Figure 4.11: A photo of the entire system.

A photo of the entire system can be seen in figure 4.11: It has large spatial capabilities, features simultaneous speech and gesture input and renders seismic data with volume- and section-based rendering methods in stereo on a large display wall. In addition, the remote workstation with three HD monitors that are connected via VNC and the tracking machine which shows the frames of the Microsoft Kinect [38] as debug information are both visible in the photo in figure 4.11.

05

USER STUDY

5. USER STUDY

This chapter describes the user study that was performed to evaluate the system and the hypotheses declared in the present work. It is divided into three parts: The objectives, the realization and the results.

5.1 Objectives

Generally, the user study aims at evaluating the hypotheses of the present work. These 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 (see section 1.3).

No combination of modalities would mean that the comparison takes place between multimodal and unimodal interaction paradigms, where the user can use only a single modality in the unimodal paradigm. However, this would cause the multimodal paradigm to have an unfair advantage: With just the use of gestures, objects of different types cannot be created without the use of a GUI or some complicated dictionary of gestures since users have to have the ability to choose what kind of object they want to create from a predefined pool (system control task). Speech, on the other hand, would grant such capabilities but inhibit precise placing of objects since phrases like *"Move the second slice in x-direction by a small amount"* or similar would never grant as much precision to the users as gestures would. In the multimodal paradigm, both precision and system control tasks can be realized at the same time using gestures and speech in a single interface.

Thus, the unimodal interaction must feature this capabilities as well, otherwise the comparison is not fair due to different constraints. The unimodal interaction therefore also features both gesture and speech, although the modalities are not combined or integrated – for every task, a user is only using a single modality, either speech (for system control tasks – object creation) or gestures (for object specific tasks – object translation and such).

One objective of the user study is therefore to compare the tasks separately giving the unimodal interaction, which can occur either with speech or with gestures, the same constraints as the multimodal interaction.

The choice for the comparison that was evaluated using the system fell on unimodal and multimodal since there were already multiple user studies for other fusion methods in the past (e.g. comparison between sequential and simultaneous usage of multiple modalities [67, 68, 50]). The evaluation therefore has to feature all tasks that are relevant unimodally and separately multimodally. If users would have the choice of interacting with the system either with multimodal interaction or with unimodal interaction at the same time, the test would not be conclusive since users might choose to interact unimodally if there is the possibility to do so even though they were asked to interact multimodally (or the reverse) [48]. Therefore, there have to be two scenarios which both hinder users to use the ITs from the other scenario.

Another objective of the evaluation is therefore to evaluate both interaction paradigms (unimodal and multimodal) in two separate groups of users – the first group can interact unimodally but not multimodally, whereas the second group is only able to interact multimodally. What user gets assigned to which group should furthermore be decided randomly to ensure an equal division.

In addition, users should be unaware of the fact that there are two groups since telling them that the test is to evaluate if the multimodal interaction paradigm is more intuitive it might influence their feedback.

Thus, a user should only get tested for one interaction paradigm at a time without knowing what the purpose of the test is.

Furthermore, the most important fact to be evaluated is which of the two paradigms is more intuitive. Evaluating the intuitiveness of an interaction paradigm is a challenging task, however (since asking the user if the interface is intuitive is obviously not enough). A user might not even know what *intuitive* means or what the designer of the test wants to find out by asking how intuitive an interaction is. Telling users is again not an option since they might be influenced by knowing the goals of the study.

For this purpose, there has to be some kind of objective measurement for evaluating the intuitiveness. A look at the definition of "*intuitive*" reveals that measuring the time a user requires to finish a task after the task is initially explained seems like a good way to start – the less time a user needs before mastering an interaction paradigm, the more intuitive it is. Of course, if the interaction is ideally intuitive, no explanation would be needed at all. Though measuring the time a user needs to master the interaction after explanation indicates if an interaction is intuitively modeled or not.

In addition, counting the times a user does an error while interacting could further indicate if that interaction paradigm is intuitive since a user who masters the interface immediately will not perform any interactions wrong.

Thus, the evaluation should test both objective and subjective characteristics, where subjective can be obtained by the use of a questionnaire and objective by measurements.

An additional objective is to have the same explanation for all participants in either of the two groups. That way, the possibility of a tutor influencing a participant can be eliminated.

At last, if the results indicate that there could be a significant difference in one of the results, the significance should be statistically evaluated using a student's t-test [35].

5.2 Realization

The realization of the user study met all requirements explained in section 5.1: Two separate groups with separate applications inhibiting participants to use the ITs of the other group, explanation for all participants in the exact same way, measuring subjective and objective characteristics and performing statistical t-test for results that seem greatly different.

The study was performed with a total of 10 participants and with most of them being employees at Fraunhofer IAIS. The first few questions in the questionnaire were designated to determine a few characteristics of the participants: The results of those show that all participants were daily computer users, two of them were females and eight of them males. The average age of all participants was 26. Furthermore, about half of the participants were experienced with gestural interfaces such as the Microsoft Kinect [38].

The participants were equally divided into two groups – “*Unimodal*” (*U*) and “*Multimodal*” (*M*). For both groups there was a separate application started which featured the respectively ITs – not letting the participant use any IT from the other group. Participants in both groups knew neither what was the purpose of this test nor that there were two groups. To which group a participant would belong or which ITs they would use in the system was picked randomly. After completing the tests, participants were handed a questionnaire to evaluate their subjective feeling and opinion on the interaction paradigm (see *Appendix*).

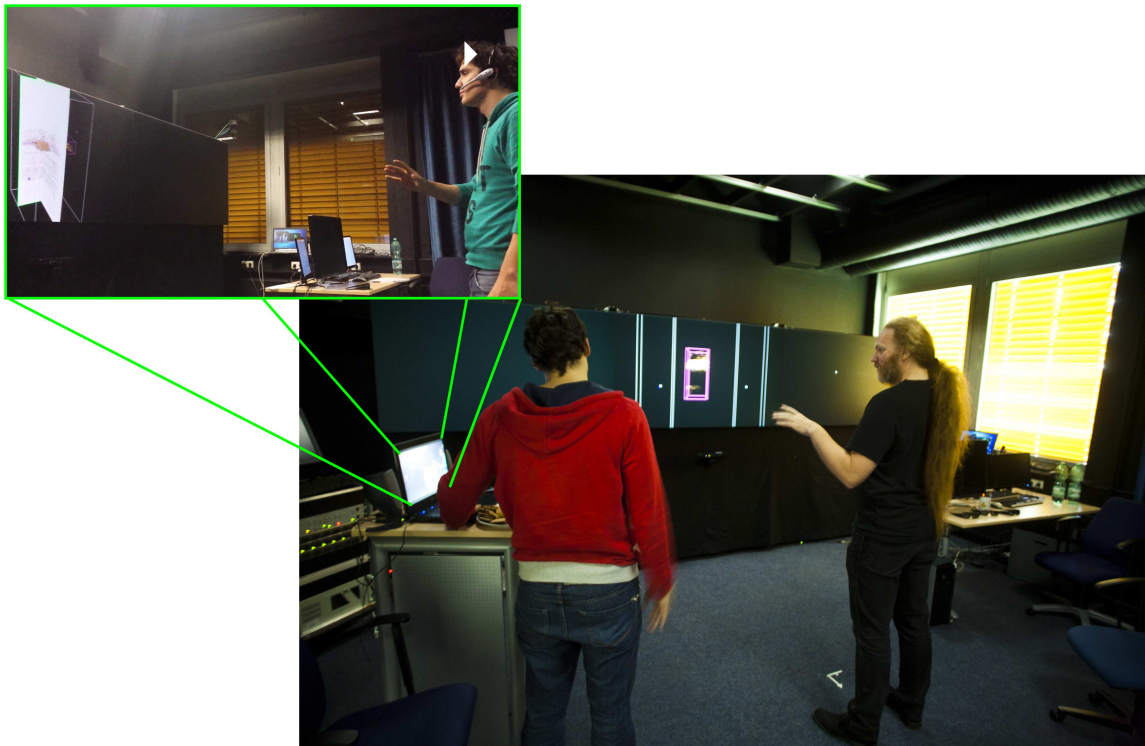


Figure 5.1: A photo of a participant currently watching the explanation video on an extra monitor.

The study consisted of 4 tasks: Navigation, manipulation of objects, creation of objects, and deletion of objects. Since the navigation task was by far the most complex task in the system (3D navigation), it was handled separately and participants were told to answer the questions in the questionnaire without regards of it.

A video was recorded for each of the four tasks explaining how the ITs for the particular task work which was then showed for each task separately to every participant (see figure 5.1). The explanation of the navigation task was processed first and in the same way as the others so participants have the chance to get used to the explanation via a video and afterwards testing it themselves.

During play back of the video, participants were told not to immediately test out the explained ITs but instead wait until the explanation has finished – the video was then paused after finishing the explanation for each task. Immediately after pausing the video, participants were shown a screenshot of the application which showed a situation specifically designed for the current task (two examples are shown in figure 5.2). Participants were then asked to reach the same situation using the just explained ITs – measuring three variables in the process: Time taken for the task, errors made by the participant and how often a participant would approach the tutor asking for assistance.

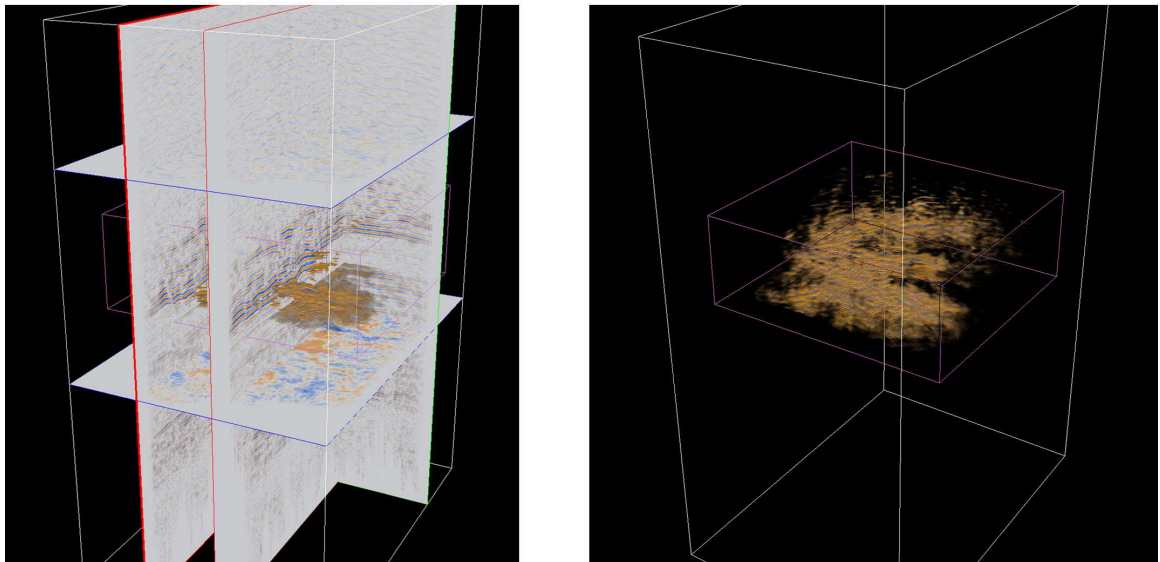


Figure 5.2: The screenshots that were shown to participants of the user study before they were asked to reach the same state in the application. Object creation (left) and object manipulation (translation and scaling, right).

By explaining the ITs first and then immediately giving a task where they are needed, the measurement of the time taken directly correlates to the time participants need to master the ITs. If, for instance, a participant requires very much time for completion the task due to many falsely recognized commands or gestures, it could indicate that the interaction is not intuitive since the user needs much time to adapt to it.

The questionnaire uses a score rating of 1 – 5 for each question – e.g. *"rate how much fun to use the system was"*, where a score of 1 means that the user had no fun at all and a score of 5 means that a user found the ITs very much fun to use. The first two questions were designed to determine from what scope the participants were (*"How often do you use a computer?"* and *"How well are you familiar with gestural interfaces such as the Microsoft Kinect?"*).

5.3 Results

The results of the user study can be divided into two subparts: objective (measurement) and subjective (questionnaire).

5.3.1 Subjective

Subjective results were evaluated using the questionnaire in the *Appendix*. Regarding the questions how much fun to use the system was (fun to use) and how easy it felt for participants to master the ITs (easy to master), both group's mean was fairly similar. The mean score of both groups and the combination of them is shown in figure 5.3 – there is no large difference visible.

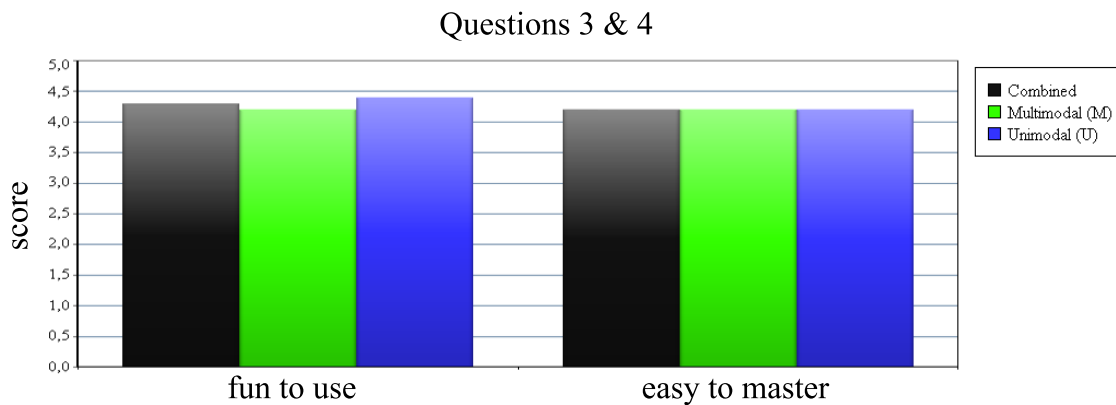


Figure 5.3: Chart showing of the mean result of questions 3 ("Rate how much fun to use the system was") and 4 ("Rate how easy it felt for you to master the shown interaction paradigm").

Similarly, the questions that could only be answered with either yes or no (score "5" or "1") do not show a clear difference. The idea behind these two questions was to determine if users think that the system was intuitive (system is intuitive) and if users could imagine using the system on a regular basis (use on regular basis) (see figure 5.4).

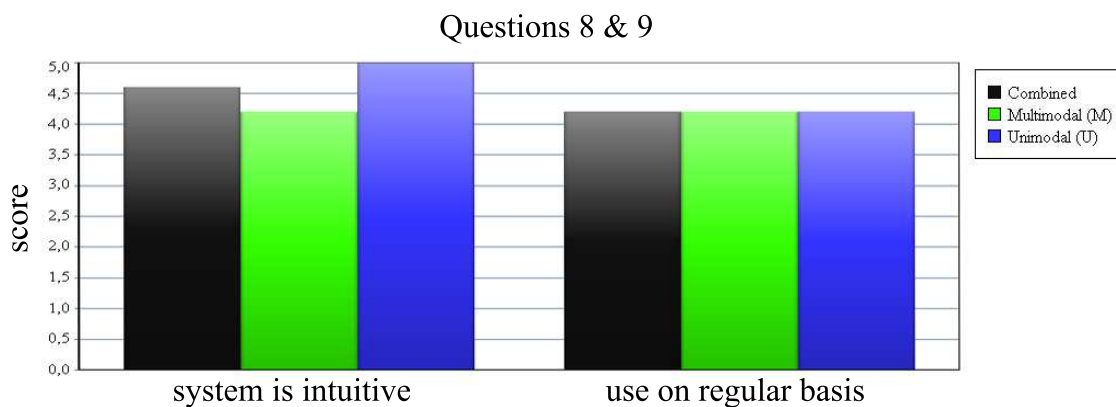


Figure 5.4: Chart showing of the mean result of questions 8 ("The shown interaction is intuitive. Would you agree?") and 9 ("Could you imagine using the application on a regular basis?").

The combined mean is shown in the two chars (figure 5.3 and 5.4) to show the overall acceptance of all participants. And, even though these results do not favor either of the two interaction paradigms, they show that the user acceptance for NUIs of the participants was high in general and that participants did not get frustrated while using either one, which in turn could influence other results.

However, the resulting mean score of the remaining three questions seem to favor the multimodal interaction paradigm (see figure 5.5). These questions aimed at determining how precise (precision), how stable (robustness) and how responsive (recognized commands) the interaction felt to the participant.

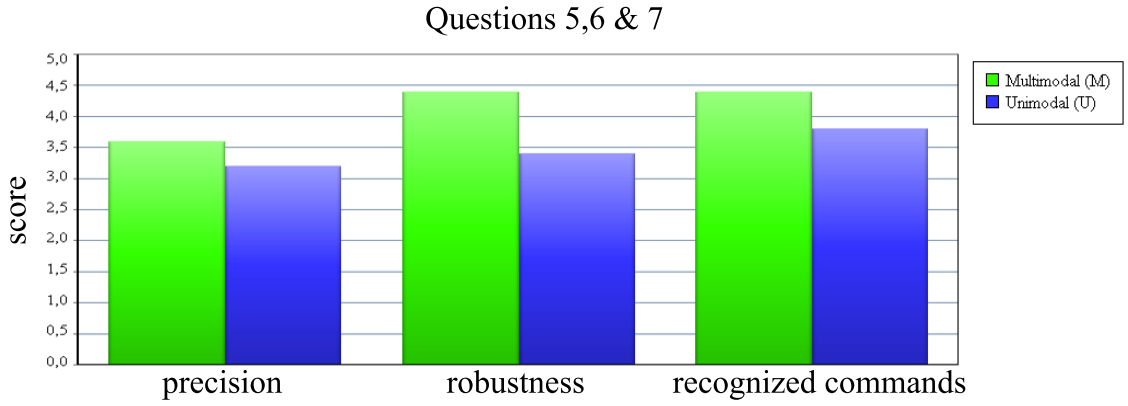


Figure 5.5: Chart showing the mean result of questions 5 ("Rate how precise you were able to place objects."), 6 ("Rate how stable the interaction worked for you.") and 7 ("Rate how many of your entered commands were recognized.").

Although the multimodal interaction paradigm is ahead in all three categories, the evaluation of the difference is significant can only a t-test tell. The results are shown in table 5.1. How the t-test was performed is explained for the objective results on the next page.

Question	\bar{x}_U	\bar{x}_M	s_U^2	s_M^2	t	ρ
precision	3.2	3.8	1.2	1.3	0.566	0.294
robustness	3.4	4.4	0.8	0.3	2.132	0.0328
recognized commands	3.8	4.4	1.2	0.3	1.095	0.153

Table 5.1: T-test results (time taken) for tasks manipulation and deletion.

The t-test results in table 5.1 show that there is a significant difference in the robustness category ($\rho < 0.05$), yet the other two categories proof to be insignificant ($\rho > 0.05$).

5.3.2 Objective

For the evaluation of the time measurement (how long did it take a participant to fulfill each task), the time values were converted into whole minutes (e.g. 00:01:30 = 1,5).

As shown in figure 5.6, there is a large difference between the particular tasks. Here, the more easy tasks such as manipulation and creation only resulted in a minor difference between the means of the two paradigms, whereas the more complex task (object creation, which involves system control) resulted in a large difference in means. It took participants in the unimodal group twice as much time for the object creation task as the participants in the multimodal group (figure 5.6).

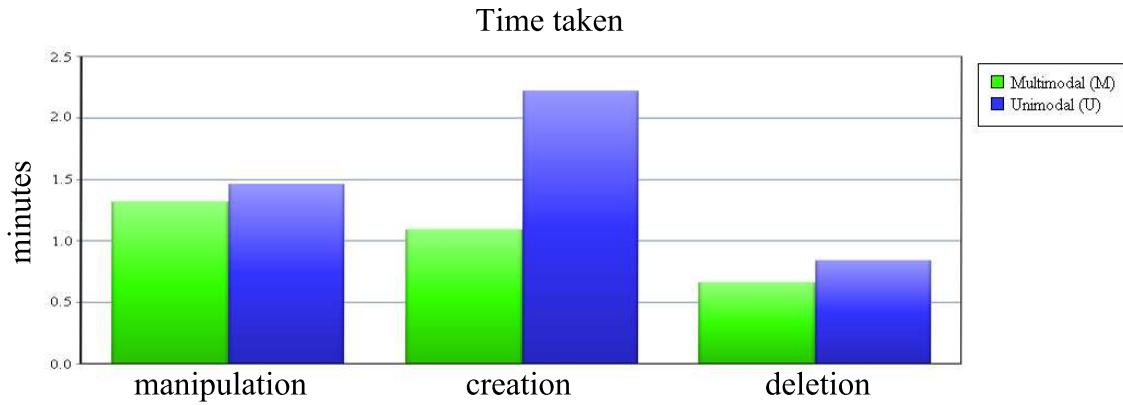


Figure 5.6: Chart that shows the mean in how long users needed for each separate task.

Here, a clear differences are visible and though, the need to perform a t-test for further evaluation of the results was present. How the test was performed is explained for the creation task below.

The null-hypothesis is defined as

$$H_0 : \mu_U = \mu_M$$

and means that the mean time taken to master the interaction paradigm is equal for the entire sample population for both unimodal (μ_U) and multimodal (μ_M). The alternative hypothesis, which directly correlates to the hypothesis of the present work, is defined as

$$H_A : \mu_U < \mu_M$$

and means that the average time taken is less when interacting multimodally, which in turn could indicate that the multimodal interaction paradigm is more intuitive, more easy to learn. The alternative hypothesis defines this test as a one-tailed t-test since it predicts a direction. To reject the null-hypothesis, the results of the user study have been used as a sample population. The sample means have the values

$$\bar{x}_U = 2.457$$

and

$$\bar{x}_M = 1.093$$

The sample size of both groups is equal to

$$n_U = n_M = 5 = n$$

The last thing needed for the t-test equation is the variance. It's equation is defined as

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

where the sum with x_i is cycling through all values for each participant. After putting in the values, the variances equal

$$s_U^2 = 0.376$$

and

$$s_M^2 = 0.349$$

and are therefore almost equal, which makes the t-test of type 2: Different sets of participants for each group with equal variances. The equation for calculating the actual t-value is then defined as [18]

$$t = \frac{\bar{x}_U - \bar{x}_M}{\sqrt{\frac{(n-1)s_U^2 + (n-1)s_M^2}{n+n-2} \cdot \frac{n+n}{n \cdot n}}}$$

Using this formula (or one of the many online calculators available for t-tests (e.g. [57])), the resulting t-value for the creation task equals

$$t_{creation} = \underline{\underline{3.582}}$$

That makes the significance level (ρ)

$$\rho_{creation} = \underline{\underline{0.004}}$$

Which means that the difference between the means is highly significant ($\rho < 0.01$) which in turn causes the null-hypothesis ($\mu_U = \mu_M$) to be rejected and shows that the time taken for the task is highly significantly less with the multimodal interaction paradigm than with the unimodal.

For the remaining two tasks the t-test results for the time taken measurement are shown in table 5.2.

Task	\bar{x}_U	\bar{x}_M	s_U^2	s_M^2	t	ρ
Manipulation	1.463	1.320	1.093	0.288	0.273	0.396
Deletion	0.840	0.663	0.227	0.103	0.687	0.256

Table 5.2: T-test results (time taken) for tasks manipulation and deletion.

For these two tasks, the results are therefore not significant ($\rho > 0.05$). Thus, showing that the difference becomes more significant the more complex the task is.

The next thing that was measured was how many errors did a participant make in each task. These errors could include saying the wrong phrase, accidentally deleting or creating an object etc. The counting of errors made by the participants could further show how fast they were able to adapt to the interaction paradigm.

The results are shown in figure 5.7: The unimodal group is slightly ahead in the deletion task and the multimodal interaction paradigm is again largely ahead when looking at the most complex task – creation. Yet only a t-test will reveal if the difference is statistically significant.

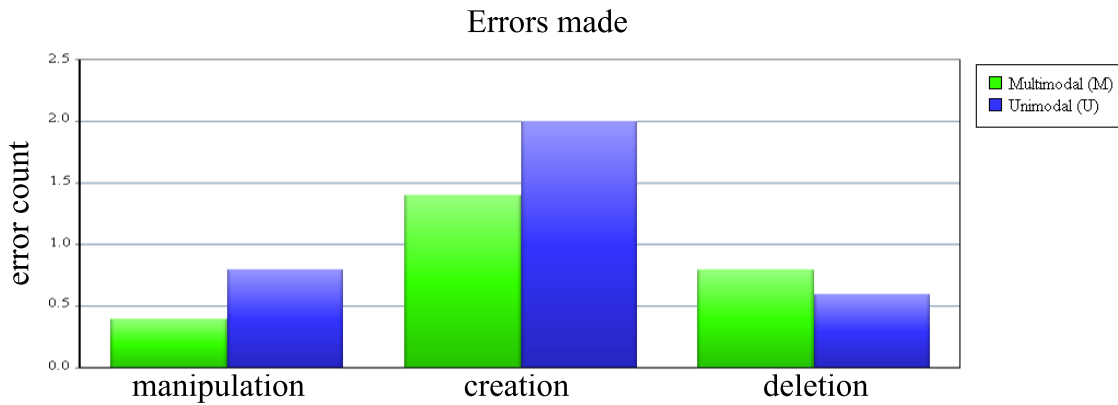


Figure 5.7: Chart that shows the mean of how many errors were made averagely for each separate task.

The results of the t-test for the errors made criteria are shown in table 5.3.

Task	\bar{x}_U	\bar{x}_M	s_U^2	s_M^2	t	ρ
Manipulation	0.8	0.4	0.7	0.3	0.447	0.333
Creation	2.0	1.4	1.5	0.8	0.885	0.201
Deletion	0.6	0.8	0.3	1.7	0.316	0.380

Table 5.3: T-test results (errors made) for tasks manipulation, creation and deletion.

It is evidently visible that due to the large differences in variances and the low number of participants, none of the tasks show a significant difference in the means of errors made ($\rho > 0.05$). Yet the overall difference comparing the three tasks again shows that the creation task is to be considered the most complex task of the three.

Although the unimodal group was ahead in the mean of errors made while performing the deletion task (figure 5.7), the insignificant difference ($\rho < 0.05$) shows that this could, for instance, be caused due to too few participants.

The last thing that was measured during the user study is the amount of times a participant would approach the tutor for help. Looking at the chart in figure 5.8 it is fairly easy to see that no t-test is needed here since the means have simply too less of a difference (looking at the t-test results for the errors made characteristic, a difference of 0.6 was already considered insignificant):

$$\bar{x}_{Umanipulation} = 0.2$$

$$\bar{x}_{Mmanipulation} = 0.6$$

,

$$\bar{x}_{Ucreation} = 0.2$$

$$\bar{x}_{Mcreation} = 0.4$$

and

$$\bar{x}_{Udeletion} = 0.0$$

$$\bar{x}_{Mdeletion} = 0.0$$

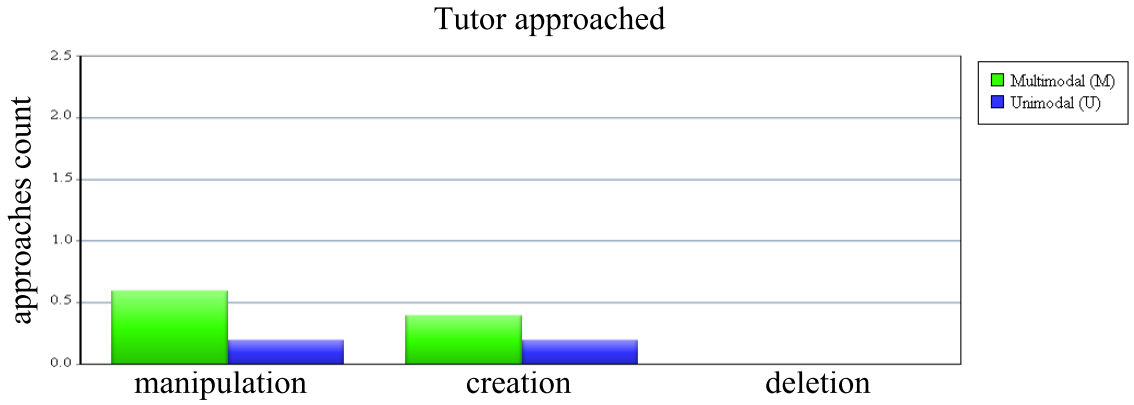


Figure 5.8: Chart that shows how many times Participants approached the tutor.

However, an interesting fact that is visible in the chart in figure 5.8 is that it shows that the multimodal group seemed to have approached the tutor more often than the unimodal group. This could indicate that the dictionary of multimodal interaction gets complicated fairly fast, yet no conclusion can be drawn from this results in the current state.

Overall, the results indicate that there are more user studies with more participants needed, yet even with five participants in each group, the result in the mean of time taken for each task shows a highly significant difference in favor of the multimodal paradigm ($t_{creation} = 3.582, \rho_{creation} = 0.004$).

06

CONCLUSION AND FUTURE DIRECTIONS

6. CONCLUSION AND FUTURE DIRECTION

6.1 Conclusion

In this thesis, a comparison between unimodal and multimodal interaction techniques was evaluated. For this purpose, a complete overview of the current state-of-the-field was presented first.

This included commonly used practices and theory for both of the topics at hand: Multimodal interaction and scientific data visualization. In addition, related work and often referenced milestones from the past were presented which also featured both multimodal interaction (or natural interaction) and scientific data visualization.

Afterwards, the system used for evaluating the hypotheses of this work was presented. The system featured seismic data visualized on a large, stereoscopic display and interaction with the Microsoft Kinect [38] and speech recognition. The system was described to enable readers to implement such a system themselves – including the requirements, the hardware used and the software architecture.

In the end, the hypotheses of the present work, which favor multimodal interaction over unimodal interaction in both fun to use and intuitiveness, were finally evaluated with the use of a user study. The user study was performed using common approaches: Playback of an explanation video, division into two groups, clearly defined tasks and analysis of each task afterwards using a student's t-test [57]. Although the performed user study had a fairly limited number of participants (5 for each group), it is believed that a user study with 5 users suffices in most cases [42].

The results of the user study indicate that multimodal interaction is more intuitive and easy to master than unimodal interaction in complex tasks, whereas in less complex tasks, the two paradigms seem somewhat at the same level of intuitiveness. Seeing that even the most complex task in the present work (object creation with the use of system control) is fairly simplistic, studies with more complex tasks are required to further evaluate the hypothesis which states that the multimodal interaction is more intuitive. Yet the results already show that there is a highly significant difference between the two paradigms and that the multimodal paradigm seems ahead of the unimodal paradigm.

As for the second hypothesis, which states that multimodal interaction gets more user acceptance than unimodal interaction (since it is more fun to use), the results could not show a significant difference between the two. This should further be evaluated with the use of more complex interaction techniques tasks, however. Here, the unimodal interaction could prove to be more frustrating as the user has to remember too many commands for a single modality (or as the cognitive load is too high).

Another significant difference was found in terms of robustness: Participants were asked to rate how stable the interaction paradigm worked for them. The significant difference in that particular question ($p > 0.05$) could indicate that due to the multimodal paradigm only reacting to an interaction if multiple modalities are used at once, the error rate can be reduced – creating a more robust solution than a unimodal interaction paradigm (here, a single movement could already be falsely recognized as intended gesture).

At last, the subjective results further indicate that the multimodal interaction paradigm creates a more precise and stable solution as the unimodal interaction paradigm.

6.2 Future Direction

Since the user study was performed using just 10 participants in total and just gesture and speech as modalities, there is a lot of room for future research: For instance, it would be interesting to add more modalities to the equation such as eye gaze or haptic. It would also certainly be good practice to repeat the test with a larger number of participants or with the use of more complex interaction techniques or tasks. Furthermore, other fusion engine methods could be added to the equation – for instance, comparison between a synergistic and an alternate method.

Although the evaluation of the interaction paradigms should not depend on the data that is visualized, evaluating the same modalities, tasks and combination with data from a different field could be used as proof.

At last, seeing that the system is expandable as is, it could be easily expanded by, for instance, a small to desktop-sized display featuring modalities that have little spatial capabilities such as pen- and touch-based interaction.

BIBLIOGRAPHY

- [1] Angelopoulou, A., García-Rodríguez, J., Psarrou, A., Mentzelopoulos, M., Reddy, B., Orts-Escolano, S., Serra, J., and Lewis, A. "Natural User Interfaces in Volume Visualisation Using Microsoft Kinect". In *New Trends in Image Analysis and Processing – ICIAP 2013*, pages 11–19. Springer Berlin Heidelberg, 2013.
- [2] Arabzadeh, E., Clifford, C. W., and Harris, J. A. "Vision Merges With Touch in a Purely Tactile Discrimination". *Psychological Science*, 19(7), Jul 2008.
- [3] Barco. "Fully redundant 70" full HD 16:9 LED video wall with 3D - OverView OLS-721 — Barco". <http://www.barco.com/en/Products-Solutions/Visual-display-systems/3D-video-walls/Fully-redundant-70-full-HD-169-LED-video-wall-with-3D.aspx>, 2014. [Online; accessed last on 28-Aug-2014].
- [4] Bennett, K. C. and Rusk, D. "Regional 2D seismic interpretation and exploration potential of offshore deepwater Sierra Leone and Liberia, West Africa". *The Leading Edge*, 21(11):1112–1117, Nov 2002.
- [5] Bethel, E. W., Johnsen, C., Aragon, C., Prabhat, Rübel, O., Weber, G., Pascucci, V., Childs, H., Bremer, P.-T., Whitlock, B., Ahern, S., Meredith, J., Ostrouchov, G., Joy, K., Harmann, B., Garth, C., Cole, M., Hansen, C., Parker, S., Sanderson, A., Silva, C., and Tricoche, X. "DOE's SciDAC Visualization and Analytics Center for Enabling Technologies - Strategy for Petascale Visual Data Analysis Success,". *CTWatch Quarterly*, 3(4), Nov 2007.
- [6] Boeck, J., Vanacken, D., Raymaekers, C., , and Coninx, K. High-Level Modeling of Multimodal Interaction Techniques Using NiMMiT. *Journal of Virtual Reality and Broadcasting*, 4(2007) (2), 2007. ISSN 1860-2037.
- [7] Bolt, R. "Put-That-There". In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262–270, 1980.
- [8] Bowman, D. and Hodges, L. "Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments". *Journal of Visual Languages and Computing*, 10:37–53, Feb 1999.
- [9] Bowman, D., Johnsen, D., and Hodges, L. "Testbed evaluation of Virtual Environment Interaction Techniques". In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 26–33, 1999.
- [10] Bowman, D., Kruijff, E., Laviola, J., and Poupyrev, I. "3D User Interfaces: Theory and Practice". Addison-Wesley, Boston, 2004.
- [11] Bowman, D. A., Coquillart, S., Froehlich, B., Hirose, M., Kitamura, Y., Kiyokawa, K., and Stuerzlinger, W. "3D User Interfaces: New Directions and Perspectives". In *IEEE computer graphics and applications*, pages 20–36, 2008.
- [12] Brooks, F. "What's Real about Virtual Reality?". *IEEE Computer Graphics and Applications*, 19(6):16–27, Nov 1999.

- [13] Bryson, S., Johan, S., and Schlecht, L. An extensible interactive visualization framework for the virtual windtunnel. In *"Virtual Reality Annual International Symposium, 1997., IEEE 1997"*, pages 106–113, Mar 1997.
- [14] Bryson, S. "Virtual Reality in Scientific Visualization". *Commun. ACM*, 39(5):62–71, May 1996. ISSN 0001-0782.
- [15] Bryson, S. and Levit, C. "The Virtual Wind Tunnel". *IEEE Computer Graphics and Applications*, 12(4):25–34, 1992.
- [16] Burns, D. and Osfield, R. "OpenSceneGraph". <http://www.openscenegraph.org/>, 2014. [Online; accessed last on 28-Aug-2014].
- [17] Cao, P., Folk, M., Sobh, N., and Ricker, P. "HDF5-SRB-FLASH PROJECT". http://www.hdfgroup.org/projects/ncsa_srb/ncsa_srb_flash.html, 2006. [Online; accessed last on 28-Aug-2014].
- [18] Caprette, D. R. "'Student's' t Test (For Independent Samples, Rice University Dates)". <http://www.ruf.rice.edu/~bioslabs/tools/stats/ttest.html>, 2014. [Online; accessed last on 18-09-2014].
- [19] Chang, J. and Bourguet, M.-L. "Usability Framework for the Design and Evaluation of Multimodal Interaction". In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interactio*, volume 2, pages 123–126, 2008.
- [20] Chen, F., Ruiz, N., Choi, E., Epps, J., Khawaja, M. A., Taib, R., Yin, B., and Wang, Y. "Multimodal Behavior and Interaction As Indicators of Cognitive Load". *ACM Trans. Interact. Intell. Syst.*, 2(4):22:1–22:36, Jan 2013.
- [21] Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J., and Young, R. "Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The CARE Properties". In *Proceedings of INTERACT95*, pages 115–120, Jun 1995.
- [22] de Souza Watanabe, L. and Vidalón, J. E. Y. "3D MEDICAL DATA VISUALIZATION TOOLKIT". http://www.dca.fee.unicamp.br/courses/IA369E/2s2010/projects/vidalon_watanabe/index.htm, 2010. [Online; accessed last on 28-Aug-2014].
- [23] Delger Lhamsuren. "Interaction Techniques for Immersive Seismic Interpretation". *University of Bonn*, Aug 2014.
- [24] Dumas, B., Lalanne, D., and Oviatt, S. "Multimodal Interfaces: A Survey of Principles, Models and Frameworks". In *Human Machine Interaction*, pages 3–26. Springer Berlin Heidelberg, 2009.
- [25] Glav Soft LLC. "TightVNC: VNC-Compatible Free Remote Control / Remote Desktop Software". <http://www.tightvnc.com/>, 2014. [Online; accessed last on 28-Aug-2014].
- [26] Jaimes, A. and Sebe, N. "Multimodal human-computer interaction: A survey". *Computer Vision and Image Understanding*, 108(1–2):116–134, Nov 2007.
- [27] Keates, S. and Robinson, P. "Gestures and multimodal input". *Behaviour & Information Technology*, 18(1):36–44, 1999.
- [28] Kindlman, G. and Kniss, J. "Multi-Dimensional Transfer Functions". <http://schorsch.efi.fh-nuernberg.de/roettger/index.php/VolumeRendering/Multi-DimensionalTransferFunctions>, 2014. [Online; accessed last on 28-Aug-2014].

- [29] Lalanne, D., Nigay, L., Palangue, P., Robinson, P., Vanderdonckt, J., and Ladry, J.-F. "Fusion engines for multimodal input: A survey". In *Proceedings of the 2009 international conference on Multimodal interfaces*, pages 153–160, 2009.
- [30] Laviola, J. "MSVT: A virtual reality-based multimodal scientific visualization tool". In *Proceedings of the IASTED international conference*, Jul 2000.
- [31] Laviola, J. "3D Gestural Interaction: The State of the Field". *International Scholarly Research Notices*, Oct 2013.
- [32] LaViola Jr, Joseph J. "Whole-hand and speech input in virtual environments". *Brown University*, 1999.
- [33] LeapMotion, Inc. "Skeletal Tracking — Leap Motion Developers". <https://www.leapmotion.com/>, 2014. [Online; accessed last on 28-Aug-2014].
- [34] Lee, D. and Yannakakis, M. "Principles and methods of testing finite state machines-a survey". *Proceedings of the IEEE*, 84(8):1090–1123, Aug 1996.
- [35] McDonald, J. H. "Two-sample t-test - Handbook of Biological Statistics (3rd ed.). Sparky House Publishing, Baltimore, Maryland". <http://www.biostathandbook.com/twosamplettest.html>, 2000. [Online; accessed last on 18-09-2014].
- [36] McMahan, R., Alon, A., Lazem, S., Beaton, R., Machaj, D., Schaefer, M., Silva, M., Leal, A., Hagen, R., and Bowman, D. "Evaluating Natural Interaction Techniques in Video Games". In *IEEE Symposium on 3D User Interfaces (3DUI)*, pages 11–14, Mar 2010.
- [37] Ömer Genc. "Novel Pen & Touch based Interaction Techniques for Seismic Interpretation". *University of Applied Sciences Düsseldorf (FH D)*, Jul 2013.
- [38] Microsoft Corporation. "Kinect For Windows". <http://www.microsoft.com/en-us/kinectforwindows/>, 2014. [Online; accessed last on 28-Aug-2014].
- [39] Microsoft Corporation. "Microsoft Speech API (SAPI) 5.4". [http://msdn.microsoft.com/en-us/library/ee125663\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ee125663(v=vs.85).aspx), 2014. [Online; accessed last on 28-Aug-2014].
- [40] Morrison, K. and McKenna, S. "Contact-Free Recognition of User-Defined Gestures as a Means of Computer Access for the Physically Disabled". In *Workshop on Universal Access and Assistive Technology*, pages 99–103, 2002.
- [41] Navarre, D., Palanque, P., Bastide, R., Schyn, A., Winckler, M., Nedel, L., and Freitas, C. "A Formal Description of Multimodal Interaction Techniques for Immersive Virtual Reality Applications". In *Human-Computer Interaction - INTERACT*, pages 170–183. Springer-Verlag, 2005.
- [42] Nielsen, J. "Why You Only Need to Test with 5 Users". <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>, 2000. [Online; accessed last on 18-09-2014].
- [43] Nigay, L. and Coutaz, J. "A Design Space For Multimodal Systems: Concurrent Processing and Data Fusion". In *Conference on Human Factors in Computing Systems*, pages 172–178, 1993.
- [44] NVIDIA Corporation. "Quadro Quad Buffered Professional Stereo Technology — NVIDIA". http://www.nvidia.com/object/quadro_stereo_technology.html, 2014. [Online; accessed last on 28-Aug-2014].
- [45] NVIDIA Corporation. "NVIDIA Quadro Plex 7000". <http://www.nvidia.com/object/product-quadroplex-7000-us.html>, 2014. [Online; accessed last on 28-Aug-2014].

- [46] Object Management Group, Inc. "Unified Modeling Language (UML)". <http://www.uml.org/>, 2014. [Online; accessed last on 28-Aug-2014].
- [47] Obrenovic, Z. and Starcevic, D. "Modeling Multimodal Human-Computer Interaction". *Computer*, 37(9):65–72, Sep 2004.
- [48] Oviatt, S. "Ten Myths of Multimodal Interaction". *Commun. ACM*, 42(11):74–81, Nov 1999.
- [49] Oviatt, S., DeAngeli, A., and Kuhn, K. "Integration and Synchronization of Input Modes During Multimodal Human-computer Interaction". In *Referring Phenomena in a Multimedia Context and Their Computational Treatment*, pages 1–13. Association for Computational Linguistics, 1997.
- [50] Oviatt, S., Lunsford, R., and Coulston, R. 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*, pages 241–249. ACM, 2005.
- [51] Pea, R. D. "The Collaborative Visualization Project". *Commun. ACM*, 36(5):60–63, May 1993.
- [52] Peters, D. "Interface Design for Learning: Basic Principles A-Z". <http://www.peachpit.com/articles/article.aspx?p=2164586>, 2014. [Online; accessed last on 28-Aug-2014].
- [53] Plate, J., Tirtasana, M., Carmona, R., and Fröhlich, B. Octreemizer: A hierarchical approach for interactive roaming through very large volumes. In *Proceedings of the Symposium on Data Visualisation 2002*, VISSYM '02, pages 53–ff. Eurographics Association, 2002.
- [54] Reda, K., Knoll, A., ichi Nomura, K., Papka, M. E., Johnson, A. E., and Leigh, J. "Visualizing Large-Scale Atomistic Simulations in Ultra-Resolution Immersive Environments". In *Proceedings of the IEEE Symposium on Large-Scale Data Analysis and Visualization*, pages 59–65, 2013.
- [55] Rusu, R. and Cousins, S. "3D is here: Point Cloud Library (PCL)". In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4, May 2011.
- [56] Safari Books Online. "Comprehending the orbit camera - WebGL Game Development". <http://www.safaribooksonline.com/library/view/webgl-game-development/9781849699792/ch05s06.html/>, 2014. [Online; accessed last on 28-Aug-2014].
- [57] Stangroom, J. "T-Test Calculator for Comparing 2 Independent Means". <http://www.socscistatistics.com/tests/studentttest/Default.aspx>, 2014. [Online; accessed last on 18-09-2014].
- [58] Sutherland, I. "The Ultimate Display". In *IFIP Congress*, pages 506–508, 1965.
- [59] The Apache Software Foundation. "Apache ActiveMQ – Index". <http://activemq.apache.org/>, 2011. [Online; accessed last on 28-Aug-2014].
- [60] The Eye Tribe Aps. "The Eye Tribe". <https://theeyetribe.com/>, 2014. [Online; accessed last on 28-Aug-2014].
- [61] The Khronos Group. "GLUT - The OpenGL Utility Toolkit". <http://www.opengl.org/resources/libraries/glut/>, 2014. [Online; accessed last on 28-Aug-2014].
- [62] Turk, M. "Multimodal Interaction: A review". *Pattern Recognition*, 36:189–195, 2013.
- [63] Uthe, N. "Wii Remote - VR-Nerds". <http://www.vrnerds.de/wii-mote-preview/>, 2014. [Online; accessed last on 28-Aug-2014].
- [64] van Dam, A. "Post-WIMP User Interfaces". *Communications of the ACM*, 40(2), Feb 1997.

- [65] Wallace, E. "Finite State Machine Designer". <http://madebyevan.com/fsm/>, 2010. [Online; accessed last on 28-Aug-2014].
- [66] Wikimedia Foundation, Inc. "Natural user interface - Wikipedia, the free encyclopedia". <http://en.wikipedia.org/wiki/Naturaluserinterface>, 2014. [Online; accessed last on 28-Aug-2014].
- [67] Xiao, B., Girand, C., and Oviatt, S. L. Multimodal integration patterns in children. In *INTERSPEECH*, 2002.
- [68] Xiao, B., Lunsford, R., Coulston, R., Wesson, M., and Oviatt, S. Modeling multimodal integration patterns and performance in seniors: Toward adaptive processing of individual differences. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 265–272. ACM, 2003.

QUESTIONNAIRE

USER STUDY “MULTIMODAL INTERACTION”

Background:

The user study “Multimodal Interaction” aims at evaluating if the used interaction paradigm feels “natural” or “intuitive” to the user. For this purpose, it was evaluated how fast users are able to master the interaction. Each task (manipulation, creation, deletion) was explained using a previously recorded explanation video, whereas afterwards the user was given a task that he or she should fulfill.

To evaluate the subjective component (how the interaction felt to the user), this questionnaire was created. Thus, please fill out the following questions to the best of your mind

Group:

1. How often do you use a computer?

never/
very rarely

☐☐☐☐☐

daily/
very often

2. How well are you familiar with gestural interfaces such as the Microsoft Kinect?

not at all

☐☐☐☐☐

very familiar

3. Rate how much fun to use the application was for you.

very little
fun

☐☐☐☐☐

very much
fun

4. Rate how easy it felt to you to master the interaction.

very difficult ☐ ☐ ☐ ☐ ☐ very easy

5. Rate how precise you were able to place objects.

very unprecise ☐ ☐ ☐ ☐ ☐ very precise

6. Rate how stable the interaction worked for you.

very unstable ☐ ☐ ☐ ☐ ☐ very stable

7. Rate how many of you entered commands were recognized correctly

none of them ☐ ☐ ☐ ☐ ☐ all of them

8. "The shown interaction is intuitive." Would you agree?

no ☐ ☐ yes

9. Could you imagine using the application on a regular basis?
(e.g. 3 - 4 days a week for 2 - 3 hours)

no ☐ ☐ yes

10. Here is plenty of room for additional feedback. Don't hold back!

Thank you for participating!