Bauhaus-Universität Weimar
Faculty of Media
Degree Program Computer Science and Media

Can't Touch This - A Prototype for Public Pointing Interaction

Master Thesis

Michael Frank Pannier
born 19th December 1984 in Dessau

Registration Number 51755

1st Supervisor: Prof. Dr. Eva Hornecker
2nd Supervisor: Prof. Dr. Sven Bertel

Date of Submission: 2nd December 2014

Abstract

Museums can be perceived as old fashioned. Potential audiences therefore often do not consider visiting one for themselves. Nevertheless, there are many modern and open minded ones, which are willing to experiment with new possibilities, to get rid of their dusted reputation and to evolve.
In order to increase interactive potential of exhibits behind glass, I implemented a novel information interaction system for a museum of pre- and protohistoric history. The challenge was not only to develop a working, intuitive prototype, but also consider low maintenance and robustness for everyday use. The system I developed employs the natural behavior of visitors by detecting potential users and enabling them to interact with the system via pointing gestures. Additionally, the museum personnel can easily set up and maintain the system themselves.
Interaction of the system is initiated automatically with a visitor walking up to the installation. No additional devices on the users side are required, they only need to point at one of the interactive exhibits inside the showcase. The system then determines which exhibit is addressed and displays corresponding information in the form of explanatory texts and detailed images on a screen.

Contents

List of Figures

List of Tables

Affidavit

Affidavit

I hereby declare that this master thesis has been written only by the undersigned and without any assistance from third parties. Furthermore, I confirm that no sources have been used in the preparation of this thesis other than those indicated in the thesis itself, as well as that the thesis has not yet been handled in neither in this nor in equal form at any other official commission.

Michael Pannier

Abbrevations

IMI	Interactive Museum Installation
MS	Microsoft
RFID	Radio-Frequency Identification
FSD	Functional Specification Document
MIT	Massachusetts Institute of Technology
SDMS	Spacial Data-Management System
WYSIWYG	”What you see is what you get”
GUI	Graphical User Interface
SUI	Single-User Interface
MUI	Multi-User Interface
HCI	Human Computer-Interaction
TUI	Tangible User Interface
VR	Virtual Reality
3D	three-dimensional
HMD	head-mounted display
DOF	degrees of freedom
AR	Augmented Reality
SDK	Software Development Kit
2D	two-dimensional

BCI	Brain-Computer Interface
MVT	Museumsverband Thüringen
HDD	Hard Disk Drive
PDLC	Polymer Dispersed Liquid Crystal
IR	infra-red
FUBI	Full Body Interaction
UI	User Interface
wpm	words per minute
cpm	characters per minute
IV	Independent Variable
SD	Standard Deviation
ID	Identifactor
AOA	Area of Affinity

1 Introduction

In this work, I describe the development of an IMI. The system presents a novel way to augment public displays with a system that is intuitive to use, easy to maintain, and inexpensive. Natural interaction without additionally required devices on the users end lowers inhibition and frustration. Simultaneously, awareness for the displayed contents is raised.

Before this system could be developed, a collaboration with a local museum had to be established. Because the installation would be based on in- and output modalities that actually make sense in a museum, visitors and staff had to be observed and interviewed. After looking at several suitable museum candidates in Weimar. I chose the one with the most promise in fitting properties as well as institutional openness for my purposes. Together, we conceived some ideas for possible installations. Not all of them were applicable an some were too far off my expertise. Nevertheless, there were two concepts for augmenting the *gravesite of Haßleben-showcase*, that we were very interested in and excited about.

The first concept, *"Interaction with Tangibles"*, was directly addressing visitors' haptic perception. Therefore, it was planned to use Microsoft (MS) Gadgeteer-hardware[1] as embedded components of tangible devices. A number of reproductions could be placed outside the showcase. Each interactive tangible could then be manipulated or placed on a pedestal to gain information about its corresponding exhibit. Here, certain exhibits could have been photogrammetically scanned in three dimensions. After that, the digital model could be scaled to a handy size and otherwisely modified. Ultimately, the tangible

1 MS Gadgeteer is a modular system of various hardware-components distrubuted by GHI Electronics. It resembles Arduino- and other microcontrollers.

could be printed or casted. Such an object could then be enhanced by using Radio-Frequency Identification (RFID)-technology[2]. In order to make it interactive, it would be fitted with such a RFID-tag. There is a RFID-module for Gadgeteer, which would have allowed identification of each tangible. The corresponding information could then be provided by any medium compatible with Gadgeteer.
A different approach was based on an assumption of natural behavior of visitors. After a meeting at the museum, a second concept of *"Interaction by Pointing"* emerged. Later the underlying assumption was confirmed by the observation of visitors' behavior around showcases. Visitors do not only talk about exhibits, but they also point at certain exhibits during interaction with each other. Therefore, a device should be build or utilized for users to point with, enabling them to select a certain exhibit inside the showcase. Additional information about the point of interest would then be displayed in an appropriate manner.
While all involved understood these concepts were fairly comprehensible, their technical realizations were unclear at first. Throughout further investigations, the work turned from testing various modes of input to a more technical approach. Both ways of input are special and revealed different challenges along the way.

Throughout the following chapters I documented my proceedings during the development of the aforementioned system. Chapter 2 gives background information about the fields of study which are included in my work. Thus, there is a brief outline about the progression of technologies employed by museums, users behavior around public interfaces and with tangibles. In addition, a brief overview of virtual reality-techniques is given.
Afterwards, I describe my goals for the development of this system. Before I come to explain the schematics and evaluation of my implementations, I give a short review of my partnering process. Thus, Chapter 3 deals with finding a suitable museum for a collaboration.
Chapter 4 explains the whole development-process of the system's schematics and functionality. It begins with possible system designs and explains their possibilities and constraints. In the end of Chapter 4, the final concept is shown along with necessary

2 RFID-transponders or -tags do not require any batteries, are cheap and robust. In addition, their range is very limited, which allows several tags on one tangible.

obligations such as an Functional Specification Document (FSD) and the contract between me, the university and the museum.
Chapter 6 addresses the implementation of the system's functionality. Therefore, all libraries and softwares are explained in more detail.
Experimental lab-installations and the final museum-installation are described in Chapter 5. Therefore, measurements, hardware specifications, and other influential criteria are presented in detail.
The final installation is evaluated in Chapter 7, where visitors were observed and interviewed before and after alterations by the system. Chapter 8 then deals with the discussion of the evaluation's findings.
In the end, I thought about future work, which could improve, extend, and follow my system. In Chapter 9, I would also like to mention reactions and suggestions I encountered along my work.

2 Background and Motivation

Over time, public places became more and more enriched with all kinds of technology. Nowadays, on nearly every corner something is beeping or blinking and buttons, leavers and knobs make us - their potential users - interact with our environment. This trend does not spare anyone or anything. Even traditionally calm and sophisticated places open up to the possibilities of contemporary technologies.

2.1 Museums

Museums, much like libraries, are foremost seen as a place of knowledge and its preservation. Hence, visitors behave in a very reserved manner. Whilst applying for libraries, museums are willing to involve people instead of merely providing information. Many museums therefore employ guides, who give tours and tell visitors about the exhibits. In addition to their factual knowledge, they also provide interesting anecdotes and other exciting information needed to bond with a certain topic. Apart of instructive and teaching staff, museums have tried many other ways to involve their visitors. One of those is utilizing technology. With time technology evolved, and so did technological augmentations in museums.

The name "museum" comes from the ancient greek's "Museion". It refers to a sanctified place in honor of a muse [33]. Basically, museums are collections of arts and science or at least parts of them on display. In modern history, those collections were of an artistic nature and mostly private. Later, scientific and otherwise cultural museums were established for the general public [ibid.].

One of the first high-tech installation of the modern age was the *Diorama*. In 1821,

Louis Jacque Mandé Daguerre[3] and the painter Charles Marie Bouton partnered up to develop this spectacle. It is an elaborate combination of painting and lighting [44]. Through ingenious lighting, the paintings became vivid. This way, a diorama could simulate the moods of a whole day within minutes. Thus, it is seen as an early predecessor of the cinema. Even today, although in much smaller size, dioramas are still of certain interest [38].
The first interactive displays appeared at the *Urania* in Berlin around 1889, when they introduces visitor-activated models and a scientific theater. In 1907 the *Deutsches Museum* in Munich also began experimenting with film and mechanical models, which were operated by visitors [29]. Later, other museums all over the world followed. Since then the

> *"[...] wider museological community's understanding of nature and purpose of interactiveness"* [42]

has taken shape.

> *"This understanding almost invariably involves:*
>
> 1. *The presence of some technological medium.*
> 2. *A physical exhibit which is added to the main display.*
> 3. *A device which the visitor can operate, involving physical activity."* [42]

As electronics and microchips evolved, computers became popular and affordable. The technological equipment of museums grew with what was available and new kinds of devices and installations appeared. Today, nearly every museum has a certain guide system such as an audio guide. It either leads visitors through the museum on a predefined course or a visitor can choose the track according to a given code for each included exhibit. In 2004, Chou et al. compared different museum guide systems in various categories, which were considered necessary to provide a user-friendly and informative experience. Expositors, tape machines, CD-players and a PDA were judged. The PDA was most versatile and easy to use system [8] (see Figure **??**). The described system had the portability of an audio guide, but due to position recognition the PDA would always

3 Daguerre is a scene painter and stage designer by trade. He also is the inventor of the first photographic process called daguerreotypy

present the current exhibit. The system could replace common audio guides and immobile information terminals all together. In addition, it still was able to give predefined tours depending on the user's interests.

F I G U R E

Yet another chapter was opened, when the internet and wireless communication were introduced. Museums began to also maintain websites. Burgard et al. went a step further, included robotics and build an autonomous tour-guide robot called *RHINO* [7] depicted in Figure **??**. It was able to navigate through the museum freely and without bumping into visitors. On demand, RHINO worked as an information terminal for present visitors and it could be used as a tour-guide as well, because it had a simple build-in web interface. Thus, the museum's contents where simultaneously used by the website and the robotic tour-guide. RHINO was deployed at the *Deutsches Museum Bonn* in 1998 [ibid.].

F I G U R E

In 2002, a group from the *University of Limmerick* made a survey in *Hunt Museum*. The museum is owned and run by the Hunt-family. Its tradition is to involve the visitors since its early days. Therefor, they had so-called *cabinets of curiosity* [9], special compartments within the exhibition, where additional exhibits were hidden. For example, a curious visitor had to open drawers in order to find a collection of plates. Via this exploration, the visitors became involved. Inspired by their observations, Ciolfi et al. implemented a completely new and interactive part of the exhibition in 2003 [14]. Two new rooms were introduced. First, there was the *study room* with three interactive devices for getting further information about certain exhibits. They were disguised as a chest, a painting and a desk. The second room, the *room of opinion*, was plain white with plinths, on which visitors could record their interpretations of the intended function of certain exhibits. In order to manage all the data, a third and hidden room was used to host all the data-servers.

F I G U R E

The *medien.welten*-exhibition at *Technisches Museum Wien* was not only showcasing technical devices from all eras and genres of modern media, it also invited visitors to make use of some. As Hornecker et al. described in [22], throughout the exhibition users were given opportunities to produce their own medial content. The interfaces ranged from an abacus over a telegraph to a whole rebuild of a news-studio from an Austrian TV-channel. Figure **??** shows those and other installations. Visitors could not only use the devices, but store some of their produced contents in a *digital backpack* [ibid.]. This way, visitors did not only have an exciting experience, but also something to remember it by later.

F I G U R E

2.2 Interfaces and Interaction

People visiting a museum come from different backgrounds and in various numbers. There can be large groups like a school class on a field trip or a single person strolling around. Their technical and physical abilities might also vary. Hence, an installation's ease of use, especially in a museological context, is of great importance. Because the interface is the only connection between users and the system, it has to be as intuitive and easy to use as possible. It dictates the way of interaction and, therefore, whether the whole systems works or not. A ground breaking system is worth nothing without proper interaction between its operator and it. This presents the need for suitable kinds of input and feedback. Thus, Ben Shneiderman once introduced his *eight golden rules of interface design*:

1. *"Strive for consistency.*
2. *Enable frequent users to use shortcuts.*
3. *Offer informative feedback.*
4. *Design dialogs to yield closure.*
5. *Offer error prevention and simple error handling.*
6. *Permit easy reversal of actions.*
7. *Support internal locus of control.*
8. *Reduce short-term memory load."* [39]

Following this guideline should yield a well operable interface. However, there still might be particular difficulties for specialized or novel systems. Especially public user-interfaces bring new factors, which are not explicitly included in Shneiderman's rules. How should an interface behave,

- in order to invite users?
- on a user's first encounter?
- if there are multiple users?

The ideal interface should be as intuitive and naturally to use as possible. This requirement was already addressed in 1980 by Richard A. Bolt. He described the *Media Room* at Massachusetts Institute of Technology (MIT) shown in Figure **??** as an office with a chair, a wall-sized screen and other analogue or electronic installations. The room's equipment allowed the user to navigate through the *Spacial Data-Management System (SDMS)* called *Dataland*. The user would sit down and had several input-devices at its disposal. One of them was a small device that could measure its position and orientation in space. The device was used to calculate where on the big screen the user was pointing. In combination with simple voice-commands the user was able to create and manipulate geometrical primitives [6].

F I G U R E

Since then, this seemingly futuristic furnishing could not be established as a common way of interaction. Keyboard and Mouse are still the most widely used input-devices. Meanwhile, touchscreens and voice-recognition are closing the gap though. The principle of "What you see is what you get" (WYSIWYG)[4] became of ever greater importance. Many interfaces are designed with Shneiderman's rules and usability in mind. Developments in Human Computer-Interaction (HCI) seem promising. More intuitive devices and interfaces are developed and tested thoroughly.

There are basically two types of interfaces. A Single-User Interface (SUI) is designed to be operated by only one user, whereas a Multi-User Interface (MUI) can be operated by

4 WYSIWYG first came up in the 1970s, when the first office-programs appeared. Due to different resolution-densities of displays and printers, it was necessary to show correct relations of letters and page. Later on, the term was synonymously used for Graphical User Interface (GUI)-elements.

a group of users at once. However, there is no strict distinction between the two. People might look over a SUI's user's shoulder and give instructions, or a lone person could operate a MUI on its own. Another factor that influences how people use an interface is the occasion. Public interfaces, such as a kiosk system at a cinema or for photo-developing provide a GUI on a touchscreen. Those systems are intended to be used by a single user, but might also be confronted with groups. Azad et al. investigated how groups behave around such kiosks. Most groups approached the interface asynchronously. Meaning, one member is interacting with the system, while the others watch. As time passes, the rest of the group might get more active due to *intra-group communication*. They further observed, that

> *"there is a semantic, profound difference between pointing and touching. Users who point are communicating ideas within a social group and may not want the technology to treat it as input."* [5]

Moreover, inter-group communication is also of great importance. Shyness or frustration may inhibit an individual or group from interacting with a system. Strangers can ease the use, when they act as an example or explain their actions to those shy or frustrated [21].

Interfaces can not only be categorized by their intended demographics. Input- and output-devices dictate the kind of interaction. As mentioned earlier, keyboard and mouse are being replaced with novel technologies. Touchscreens and voice recognition have become established means of input as well. Moreover, novel approaches towards interaction are made and influenced by novel possibilities in technology. Museums in particular strive for innovative interfaces to involve visitors and, therefore, tend to explore many fields of interactive possibilities.

> *"Interactive exhibitions are thriving, encouraged by a new approach to museology developed in response to current social demand and a much more participatory philosophy involving a redefinition of the concept of the museum in general and of the science museum in particular. [...] The classic concept of observation has been replaced by that of participation."* [13]

A Tangible User Interface (TUI) is a very natural approach. Here, a tangible object represents a digital entity. This could be a virtual object or something more abstract like an operation or property. Those tangibles can be used to interact with a system or even each other. In [41], Ullmer et al. described the principle of *token and contraint*, which played an influential role during the conception and is revisited in Chapter 4.2. Two name-giving types of tangibles are involved, tokens and constraints. They have two phases of interaction. At first, tangibles can be associated, which means that their physical shape dictates, whether tokens will or will not work with a particular constraint. After that, the constraint's properties dictate the further way of interaction as shown in Figure **??**a,b,c. In comparison to ordinary interfaces a TUI offers more haptic perception. This could increase a user's attention to the interface and as a consequence their involvement with a possibly related exhibit.

F I G U R E

Apperceptive and playful interfaces might raise visitors' involvement. Groups of visitors tend to spend more time with an interactive exhibit, because the majority wants to make the experience for itself [13].

Virtual Reality (VR) provides a less tangible approach. Contemporary VR-systems offer a way to display and manipulate three-dimensional (3D) data in real-time. Users can be immersed into vast, 3D environments, which are projected on huge stereoscopic displays or shown by a head-mounted display (HMD)[5]. Those systems require a special kind of interaction to either navigate through or select objects in the virtual environment. Therefore, special interaction metaphors were developed. Proper navigation can be realized via any input-device capable of six degrees of freedom (DOF)[6] and is rather familiar. But, since navigating to each object in order to select and manipulate it is inconvenient, a separate metaphor for interacting with objects had to be developed as well. Thus, a pointing device was introduced. In order to calculate its correct position and direction it has reflective markers in a unique pattern, which are then tracked by a system of infrared cameras (see Figure **??**a) [4]. A virtual ray into the scene is calculated accordingly. Basically, objects can be selected by pointing at them and triggering selection in some way. *A survey of 3D object selection techniques for virtual*

5 Recent, distinct examples are *Occulus Rift* and *Google Glasses*. While Occulus Rift provides a completely closed solution for 3D VR-environments, the Google Glasses are an Augmented Reality (AR)-approach. Here, additional information is displayed on top of perceived reality.

6 One DOF is either translation along or rotation around one spacial axis. Hence, there are six possible movements in 3D space.

environments showed that 29 of 31 reviewed techniques were based on ray-interaction. The remaining two used a hand avatar like in Figure **??**b. Both of them track a user's hand and one also applies ray-based leverage to extend reach [2].

F I G U R E

As mentioned earlier, keyboard and mouse are established means of input, though there are more natural ways of input though. Humans have been communicating with gestures for ages. With this in mind, *gestural interaction* arose as a concept in HCI. In [28] five basic categories of gesture styles are defined. All those gestures can be applied to two- and three-dimensional interfaces. *Deictic* gestures are defining a location and include all varieties of pointing gestures [ibid.]. They range from the cursor on a desktop PC to the VR-pointing devices mentioned above. *Manipulative* gestures are used for translation, rotation, and scaling [ibid.]. Here, the "pinch to zoom"-gesture for many touch-based devices is probably the most commonly known. More abstract gestures are *semaphoric* and language gestures [ibid.]. They consist of designated signs or movements and can be static or dynamic. The best example for static gestures is sign language. Air traffic controllers use flag waving to guide planes into parking positions by semaphoric gestures. The last category is *gesticulation* [ibid.]. In contrast to semaphoric gestures, gesticulation is purely natural and does not have to be learned. Gestures often appear in combination with speech, why they are also called *coverbial* gestures.
Since gestures are naturally motivated, gesture-based interfaces should be intuitive to interact with. With the introduction of smart phones and tablet computers touch-based interaction and gesture-based Interfaces are getting more and more relevant. In 2003, the *EyeToy for PlayStation2* began to show the possibilities of *free-hand gestures*. A few years later, Microsoft began selling the *Kinect for Xbox 360*. Independent drivers and software for it were developed almost immediately, which enabled developers and researchers to utilize the system's capabilities on regular computers. The official Software Development Kit (SDK) and *Kinect for Windows* followed about half a year later. Other devices like *ASUS' Xtion PRO* followed working with the identical internal hardware. The more detailed hardware specifications will be explained in Chapter 4. This development introduced a low cost solution for user-tracking, whilst the aforementioned tracking systems for VR are very expensive. Several interfaces were developed making use of the hardware's potential. Thus, its performance and suitability for VR-tasks was examined. Ren and O'Neill stated that

> *"More and more information and other content is visualized and manipulated in 3D, bringing a corresponding increase in the importance of effective and usable 2D user interfaces."* [37]

They conducted two kinds of test. First, they looked into two-dimensional (2D) interaction techniques such as touchscreens and *freehand 3D* interaction and described similarities and differences. Accordingly, both techniques' WYSIWYG-approach makes them spontaneous and direct. This natural *walk-up-and-use* interaction style decreases the interaction cost for any possible user. Further, users are able to move freely, because there are no extra devices to pick up and operate [37]. In the second test, the low cost solution was evaluated for common 3D interaction techniques in VR-environments. Thereupon, Ren et al. came up with a *design guideline for 3D freehand interaction* [ibid.].

- Avoid single actions with either high accuracy or keeping the hand up for long.
- Use goal crossing [1] as a trigger.
- Map complex 3D-movements on 2D interaction.
- Use the extra dimension as a trigger.
- Avoid uncomfortable hand or arm positions.

In the end, Ren and O'Neill concluded:

> *"Freehand gestural selection enabled by a single low cost camera is a potentially valuable technique enabling flexible, low configuration interaction in 3D environments, without any requirement for dedicated devices to be worn on or carried by the user. With appropriate designs, freehand 3D interaction can share the appealing fluidity and immediacy of currently popular multi-touch surfaces, [...] enabling walk-up-and-use access to services, holding the promise of wide applications for ordinary users in everyday life."* [37]

2.3 Goal of this Work

After having gathered experiences with several of the aforementioned interaction techniques, I intended to combine those experiences into a unique interface. Especially the intuitive motivation behind tangibles, touch-based interfaces and the immersive nature of VR-environments are interesting fields.

> *"Unfortunately, current user interfaces often lack adequate support for 3D interactions: 2D desktop systems are limited in cases where natural interaction with 3D content is required, and 3D user interfaces consisting of stereoscopic projections and tracked input devices are rarely adopted by ordinary users. The success, both in research and commercial applications, of recent touch-based interfaces raise an interesting possibility. Can the immediacy, control, and expressiveness of recent touch-based natural interfaces be applied to 3D problems?"* [40]

I wanted to answer this question. But before that, several issues had to be dealt with. First, a suitable cooperation partner had to be found. As described earlier, a museum in Weimar was ideal. I needed access to an unbiased audience and topical expertise. Second, we would have to agree on a feasible concept and clarify responsibilities. Subsequently, the system would have to be developed and implemented before the final installation could be evaluated in its real-world environment. Finally, some small adjustments could be made and larger improvements or augmentations should be discussed.
A museum was found, concepts were made and a TUI or an alternative concept for interaction were required. The *SMSlingshot* already combines a handheld device with 3D interaction to some extend. It is a tangible device, which enables a user to splat short massages onto a facade [16]. The device's affordance is clear, since most people are aware of the functionality of both a phone-sized keypad and a slingshot. Nevertheless, the device can only be used by one user at a time. This is where *Shared Encounters* come into play. There are different types of interrelated spaces, which offer different grades of interaction between a user and either the interface or others [15]. In course of my work, those two key concepts will reappear in more detail.

In the end, my answer was using the most intuitive device there is: a user's body itself.

My focus lied more on the presentation than on the administration software, because it will be used most of the time. Nevertheless, I implemented the whole system to be as versatile as possible and to allow others to use the system almost anywhere. It is not a customized solution for this one installation.

3 Partnering process

The very first step after having the idea of introducing a new way for information to be retrieved in public places was to find a partner to realize it with. In order to find the most promising and suitable cooperation, appropriate properties would have to be defined and considered for each institution before partnering with any of them. Afterward, a suitable exhibit and an agreement on a design for the installation would be found.

3.1 Requirement analysis

To determine an ideal partner for a cooperation, a mutual beneficial system of needs and demands had to be established. Therefore, each party's needs and offerings were identified. As Table 3.1 shows, three major criteria were determined. Possible cooperations would be based on those criteria. In addition, special characteristics would be considered as well.

	Museum	**Me**
Needs	Improvement / Innovation	Access to a public space with exhibits and visitors
	New group of visitors	Authentic content
	Publicity / Awareness	Potential test subjects
Offerings	A public space	Technological expertise
	Factual expertise	Development and testing
	Resources	Motivation

Table 3.1: Needs and Demand.

Museums want to get people interested in their respective topics. Thus, reaching more people and raising awareness is one of their main interests. A good way to attract new groups of visitors is to offer something unique and innovative. Although there are

companies offering services like guide- or information-systems, they are either cosmetic, expensive or high-maintenance. On the other hand, a museum has valuable offerings. Usually, they have a budget for renovation and improvements. The staff is highly skilled and experienced concerning the exhibits and visitors' behavior around them. Finally, a museum offers a public space, where a system can be tested under natural conditions.
The Bauhaus-Universität and specifically the chair for HCI as well as myself wanted the final system to work in a real-life environment, but not as a lab-study alone. Hence, we needed access to a public place in order to reach a broad variety of people. Those would be unbiased toward the nature of interaction and content as well. Meanwhile, we could provide our knowledge of interaction design and the suitability of contemplable technologies. And lastly, I was highly motivated to develop a working system.
After finding a cooperation partner, a FSD would be made, which includes the system's properties ordered by necessity. In addition, a contract between all parties would be drawn up to register each party's contributions and obligations.

3.2 Potential partner museums

According to Museumsverband Thüringen (MVT) [34] there are more than 50 museums in Weimar within a few kilometers distance from the town. Table 3.2 only shows museums registered at the MVT and the three towns with the most of them. Other towns have between one and six registered museums. Further, it is most likely that there are more museums than those in this list. It provides a good starting point, though.

Town	**Museums**
Weimar	26
Erfurt	12
Jena	12

Table 3.2: Museums in and around Weimar.

Regarding the high amount of museums in Weimar alone, it seemed promising to start looking for a suitable cooperation partner right here. Since 26 museums are too many to investigate thoroughly, a preselection had to be made. In the first step, the focus was on flexibility. This meant, only a small administrative apparatus could guarantee fast decisions and less organizational meetings with boards and other decision makers. Hence, all the *Klassikstiftung*'s museums were crossed of the list, narrowing it down to only 10 remaining candidates. Next, and after some further research, museums with less interesting topics or inconvenient concepts were withdrawn. This included the tiny *umbrella museum* and *Weimar Haus*, a place glutted with animatronics. Afterward, the list of candidates was down to five (see Table 3.3). A personal visit to each of these museums was indispensable now.

Museum
Deutsches Bienenmuseum
Kirms-Krakow-Haus
Museum für Ur- und Frühgeschichte Thüringens
Palais Schardt
Pavillon Presse

Table 3.3: Remaining cooperation candidates.

Gathering impressions in person was a process of three stages. In the first stage, I would visit a museum and noted its technical and pedagogical equipment. This was directly followed by the next stage, an informal introduction to some of the staff containing a chat about my plans and the respective person's attitude towards them. The final stage was a formal introduction-meeting between my professor, me and the administrative staff of each museum, that had expressed serious interest. This serious interest wasn't shown by the Kirms-Krakow-Haus and the Pavillon Presse. Hence, the aforementioned meeting only took place at the Deutsche Bienenmuseum, Museum für Ur- und Frühgeschichte Thüringens and Palais Schardt. We introduced ourselves at each venue, because a discussion about what might be done was more efficient directly on site.

3.3 Decision for a partner museum

A formal introduction-meeting went as follows: First, I explained some of my previous projects, related installations in other museums and the general intent of the professor's chair. Next, the staff explained their museum's concept and which subject area they would like to emphasize. After that, we discussed potential concepts. Those ranged from augmentations of existing exhibits to completely new installations.

Deutsches Bienenmuseum The museum is run by the beekeepers association of Thuringia. The staff we encountered was very skilled with the craft of beekeeping, but less professional concerning museum education and design. They listened to my remarks and we had an inspiring discussion about potential topics and their feasibility. Unfortunately, the association's chairman and we could not agree on a specific project. Also, because bees hibernate, visitor attendances are seasonal and also fluctuant. Hence, the Deuschte Bienenmuseum was out of the picture.

F I G U R E

Palais Schardt This venue is owned by a family, which exhibits multiple collections of art and crafts as well as the building itself. In addition, they operate a cafe and use the adjacent hall for events. The husband is a restorer by trade and gives talks about the building and its historic significance, while his wife handles planning and the cafe. Events are regular and the cafe supplies casual customers and visitors. Both were very interested in a cooperation and had some ideas for installations. But the monument protection of the building and minor financial issues complicated feasibility. Therefore, Palais Schardt also had to go.

F I G U R E

Museum für Ur- und Frühgeschichte Thüringens Since the state office for preservation of historical monuments and archeology of Thuringia is the bearer of the museum, all personnel is very competent at their field of work. In addition, the museum employs special staff, that maintains the exhibition, gives tours and is present for arising topical questions during opening hours. Classes of 5th and 6th grade visit regularly for field trips as well as visitors from all age groups. The exhibition was already altered by several media installations. Moreover, the director was very enthusiastic from the first meeting on and had several ideas, of which exhibits to emphasize.

F I G U R E

Summarizing, the Deutsche Bienenmuseum and Palais Schardt were deemed less interesting and lacking feasibility. The Museum für Ur- und Frühgeschichte Thüringens was chosen as the cooperation partner, because it checked the most boxes of the previous

Requirement Analysis (see Chapter 3.1), while the others lacked at least once in the *Needs*- or *Offerings*-category. It was the most professional and ambitious candidate with promising resources and conditions.

4 Conception

After the *Museum für Ur- und Frühgeschichte Thürigens* was chosen as a partner, all previous ideas had to be analyzed more thoroughly with feasibility in mind. Thus, impractical, and too complex or too simple ideas were eliminated in two rounds of review. At first, vague ideas were either improved or discarded. Hence, a screen displaying only information about a fossilized fireplace was eliminated. The idea of a system for digitizing stone carvings was considered too complex to realize and therefore discarded as well. Afterwards, some of the museum's staff and I looked at the contents, that could be provided for the remaining candidates. This left us with two remaining possibilities, that were promising enough from an educational as well as a technical standpoint. The first one was the reproduction of the *Fürstengrab von Haßleben*, which contains replicas and original artifacts from a 1700 year old grave of a Teutonic princess. A close second was a workshop, which should have shown how archeologists and restorers work behind the scenes of a museum. Here, the latter consisted of too many single parts and a lot of questions remained unanswered.

According to the aforementioned review, the Fürstengrab von Haßleben was most promising and therefore chosen in the end. It contains many special relics from ordinary, Teutonic pottery to rare, Roman coins and jewelry. There are original artifacts and replicas on display inside the showcase, which I am collectively referring to as *exhibits* throughout this work. Some of these exhibits are shown in Figure **??**a, b and c. The apparent eclecticism is, what makes the grave so special though. It is a sublime showcase for thriving trade and cultural exchange between Teutons and Romans as far east as Thuringia. Further, it proves how Teutons began adapting roman traditions, such as burials. In order to emphasize this insight, an interactive system was to be developed. Unfortunately, the showcase is located on the second floor. Thus, it does not get the attention it deserves. People are often tired after having visited the first floor. Hence, the museum's staff asked for an installation that would reactivate the visitors' attention.

4.1 System Preconditions

The system was to be developed and tested by me, and the museum-staff is responsible for its future maintenance. The full range of visitors' backgrounds cannot be foreseen. Some visitors might not have the proper technical experiences to operate contemporary interfaces. Consequently, it was crucial to design the system with that in mind. It had to be operable by absolute lay persons, who have no prior experience concerning information technologies. Hence, the interface had to be as intuitive and natural as possible. Four major points had to be considered.
First, established and abstract input devices, such as keyboard and mouse, had to be replaced by something more natural. In order to be intuitive, the interaction was designed to capture and use the natural behavior of visitors. Outputs, on the other hand, had to be as discreet and as conservative as possible to not disturb or interfere with the exhibition. Thus, invasive technologies such as speakers and animatronics were excluded by the museum from the beginning. This consideration only left visual and haptic channels for output. The third point was, that daily operations at the museum were not to be compromised. So, it was not possible to develop the prototype inside the Haßleben-showcase itself and a full-size mockup had to be build somewhere else. Furthermore, the showcase and its precious exhibits had to be protected from any possible decay and nothing was to be rearranged. Thus, I measured the showcase and acquired a room in which a mockup could be placed for the prototype's implementation and testing[7]. Finally, the system's components, in- and output devices, had to be robust enough to cope with daily use. Moreover, they should also stay in their intended place. This meant that they had to be somehow attaching to the showcase.

In summary, the requirements for the final system were narrowing down the possibilities right from the beginning. Hence, we came up with several ideas and followed up on all of them, until one promised to be the most feasible.

7 For a further description of the lab-setup see Chapter 5.5

4.2 Concept Development

Developing the system, we followed two initial approaches. They were supposed to lead us to an intuitive, easy to use interface, which would be very naturally operable. The first concept featured the development of tangibles. Interactive objects would be placed outside the showcase and visitors would be able to interact with them. Haptic feedback would enable visitors to experience the exhibits in an unusual way. By touching replicas of otherwise locked up exhibits a deeper involvement is highly likely. Meanwhile, the other concept was based on gestural interaction. With this concept, visitors are enabled to interact with the showcase by pointing. This approach was based on the natural behavior of visitors. Like the previous approach an uncommon experience should raise visitors' involvement and attention.

Tangibles The early idea behind this work was to work with MS Gadgeteer to develop a tangible interface for and with a museum. Thus, we first thought about how to include those Gadgeteer-modules. Therefore, I built the demo device shown in Figure **??**, which was based on a *FEZ Spider Starter Kit* [18]. In addition, it utilized an RFID-reader [20] and a potentiometer [19]. The RFID-transponders were attached to an old 2,5" Hard Disk Drive (HDD) and a wireless mouse. When the RFID-tags were recognized, an image of the object was displayed on the screen. By turning the potentiometer's knob the angle of view changed accordingly. This gave an impression of the possibilities of the hardware. Unfortunately, we only had two RFID-tags that had the size of a credit card. After some research though, I found some tags for the correct frequency band and in sizes from a grain of rice over credit cards to key chains [23]. Hence, including RFID-tags in tangibles was feasible.
The shape and size of the tangibles were still up for debate. Another point was, whether the hardware would be placed inside or outside the tangibles. This decision dictates the shape and size of the tangibles and therefore the interaction. If it would be placed inside, the tangibles would have to be big. They would have turned out at approximately the size of a box of milk. Such an *active tangible* would be handy and a whole system could be concentrated in one device. On the other hand, they would be prone to damage and maybe even theft. Hence, the tangibles would have to be tough and in some way attached to the showcase. In addition, batteries would have to be either charged or

changed. This would take a certain amount of maintenance.
With the hardware outside the tangibles and hidden in a pedestal in front of the showcase, the tangibles could be smaller. Moreover, *passive tangibles* grant more flexibility concerning the shape as well. As described earlier (see Chapter 2.2 in [41]), the tangibles could have different features depending on certain properties. In this case, several RFID-tags could be placed in each tangible. Depending to their *constrained* collocation on the RFID-reader, different reactions of the system could be triggered. In contrast to Ullmer at al., 2005, though, this affordance would be hidden and thus less obvious. The tangibles would have to be attached to the pedestal as well, although they would be less expensive to replace.

Both approaches had their advantages and disadvantages and none of them was concrete enough to make a decision. Thus, we continued to specify the concepts depending on their strengths and weaknesses. We did this, by anticipating probable relations between the exhibits inside showcase and the behavior of visitors behind the glass. There are several things visitors tend to do, if they are interested in an exhibit. They would like to inspect it up close. First, this means they would like to touch an exhibit and feel it. Second, they want to see it in more detail and from different angles. Next and induced by restrictions, visitors talk about an exhibit or request further information. This could be anything from its age to where and how it was found.
An active tangible could provide nearly all of those qualities in one package. It could - like the demo device - be fitted with a display and an RFID-reader. The corresponding RFID-tags could then be placed close to the device in order to trigger a particular output. Those outputs could be saved either on the device itself or provided by a server. The question of how to trigger different reactions was to be answered next. The device could either be placed on a pedestal equipped with RFID-tags or the tags had to be brought to the reader in any other way. As mentioned earlier, an active tangible would be sizable and it would have to be related to the showcase's topic as well. Hence, it would be reasonable to combine those two criteria and fabricate enlarged reproductions of exhibits from the showcase. In order to fit the whole hardware, an active tangible would have to have a simple shape. This unfortunately excluded several of the more interesting exhibits, such as coins, a golden ring and other jewelry. Some options remained though. There was a skull, pottery and the metal remains of two jewelry boxes.
The passive tangibles did not appear to cause this much consideration. Any exhibit

could have been 3D scanned[8], turned into a digital model, appropriately altered to fit an RFID-tag and then printed or milled out. The printed or milled reproduction could be used as a positive to produce casting molds, afterwards. Thus, replacing damaged or otherwise lost tangibles would be more cost-efficient. In addition, it could be done by the museum-staff themselves. One or more RFID-readers could be placed in a pedestal in front of the showcase. Depending on the RFID-reader and a tangible's tag, the system would display the corresponding output.
During those considerations, a third possibility came up. A hybrid approach that combined both principles was possible as well. The reproduction of a jewelry box could be turned into an active tangible and passive tangibles could be put inside to trigger an output. The RFID-reader would be placed underneath the box's floor and the display in the lid. In order to provide different types of content, we thought about also producing two different types boxes. A more or less *authentic reconstruction* made of wood and metal fittings could provide authentic information about a passive tangible's cultural background. Meanwhile, the other box could be constructed of transparent material, which would allow the user to see the hardware. This *futuristic reconstruction* could provide statistical content for the same passive tangible.

However, the main problem remained with all approaches. Some kind of pedestal would have to be built and placed outside the showcase to hold the active and/or passive tangibles. Although passive tangibles would have been more cost-efficient to replace than active ones, maintenance was rated too high. Furthermore, if the pedestal was not to obscure the showcase, it would have been too low[9] to grant satisfactory access for any visitor.

Pointing The alternate concept took a completely different approach. It was more related to VR and the interaction in 3D environments, where users are pointing at an object to select it [2]. The underlying idea was to develop an information system that would be based on pointing-based interaction. A user points at an exhibit inside the showcase, the system recognizes the gesture, calculates the intended target and displays

8 The scans could have been done in the labs of the chair of Computer Vision and Engineering at Bauhaus-Universität Weimar.
9 The height of the showcase floor is about 65cm. For more details see Chapter 5.

the corresponding information.
Since the display should not interfere with the exhibits or occlude them, we had to make decisions about the position, size and type of the display. In order to not occlude exhibits, the display should not be placed in front or above the exhibits. Directly behind the glass panel would also have been problematical. It should have been placed along the visitors' viewing direction as they already would be looking into the showcase. This way, it would still imply coherence through visual proximity. A monitor on the one hand, and a projector on the other were two possible technologies to choose from. Both came with their own challenges. While a projector would have been easier to conceal than a monitor, a monitor would produce less heat and noise. Because most of the visitors approach the showcase from the long side and tend to stay there for most of the time, the display should be visible from this direction. This meant placing the projection plane or display on the opposing wall. Another solution for a projector came up during this consideration. A Polymer Dispersed Liquid Crystal (PDLC) switchable film [27] could have been placed on the glass panel. Whenever the system was activated, the film and projector could have been activated as well[10]. Unfortunately, this solution would have been too expensive and difficult to install. A projection in the other direction was also disregarded, because the cost and heat issues caused by a projector were considered to high. Heat produced by a projector causes issues regarding the artifacts' conservation and is a safety risk for the sealed showcase. Therefore, we decided to install an LED-screen. It should be placed inside the showcase close to the exhibits.

Object selection in VR-environments [2] and the *SMSlingshot* [16], nearly always use a *pointing device* of some sort. With such a device, a potential user could directly point at the original exhibits within the showcase and trigger the corresponding reaction of the system - displaying related information. As described in detail in Chapter 7.1, I observed interactions between visitors and the showcase as well as between each other. During the pre-study, it turned out that visitors often pointed at the particular exhibits they were talking about. The interface could be designed to emulate this natural interaction between visitors and incorporate of the natural behavior.
The first intention was to rebuild the SMSlingshot with Gadgeteer-hardware. The tan-

10 A PDLC switchable film can be switched between a transparent and an opaque state. In its opaque state, it can be very well be used as a projection surface [27].

gible was equipped with a microcontroller, a small display, a keyboard, a green laser, a wireless transmitter and of course batteries. A PC was used to put all the information together and render the output. Therefore, it had a camera to track the point a user was aiming at and a corresponding transmitter to receive the fired messages [16]. All those modules could be provided by Gadgeteer except the laser. A laser could have been controlled with a *Breakout module* [17] and a relay. However, shooting a laser into the showcase was a delicate issue. Hence, this solution had to be revisited, because for safety reasons it was not feasible. There could have been injuries of visitors' eyes or some of the precious exhibits might have reacted to the laser's energy in a corrosive way. We did not want to take those risks, but we were very keen on the idea of pointing interaction. Thus, I looked for other tracking methods. We could have used a tracking system similar to the aforementioned ones used in VR. Those systems are expensive to install and maintain, though. Moreover, a proper compatibility with Gadgeteer was doubtful. So, I started looking for alternatives to Gadgeteer, too. Two established systems immediately came to mind. First, the *Nintento Wii*, which uses a wireless device with pointing capabilities and additional inputs. Second, the *MS Kinect*, which is able to recognize free-hand gestures and might not require any device. Both are comparably inexpensive to acquire, have experienced support and communities and use less dangerous infra-red (IR) light. The decision between the two was made according to the same criteria as mentioned above. Pointing with no device should be a more intuitive way to interact with the exhibition and other visitors than any handheld device. Furthermore, the restraint to use the system should be reduced. No tangible or pedestal would have to be created and attached to the showcase, which decreased cost for maintenance. Hence, the MS Kinect was chosen.

There is a Kinect for MS Windows along with a special SDK for MS Visual Studio. As it turned out, the hardware inside the MS Kinect was developed by *PrimeSense* and is also used by the *ASUS Xtion PRO*. This 3D-sensor is less expensive and smaller, which allows to be less intrusive inside the showcase. Besides, we already had some of them at the faculty, which meant that I could start developing right away. Another change was the decision for an open source SDK called *OpenNI*[11], which in combination with its add-on *NiTE* enabled me to use *skeleton tracking*. This was critical for my approach,

11 OpenNI was co-founded by PrimeSense, a hardware developer that produces 3D sensing hardware. In November 2013 PrimeSense was bought by Apple, whereupon OpenNI was shut down.

because I needed to have a 3D vector in order to be able to calculate where a user was pointing. Skeleton tracking would deliver the joints of a tracked person. Hence, I was able to retrieve the directions a limb is oriented in. If this vector was extended, I was able to calculate its possible intersection with an exhibit. More about used software and the exact calculations can be found in the next chapter.

The last topic that needed addressing was *feedback*. Since there would be no haptic or acoustic feedback, and no *glowing dot* produced by a laser either, future users would need another visual feedback in order to be able to see where they were pointing and determine how to correct that. Once more, Gadgeteer could have provided a solution. Our first idea was to replace the laser's dot by a spotlight. The system would calculate the position a user was pointing at and transmit it to a Gadgeteer-system. It would then move a special highlight to this position within the showcase. Only two actuators would be sufficient. The maintenance of this kind of installation could become very complicated though, because the system would have to be installed on the ceiling of the showcase. Actuators need to be calibrated regularly and mechanical gearing will wear out. Hence, this realization concept was dismissed. Nevertheless, the principle should remain the same. Thus, the aforementioned position would now be shown on an overview of the showcase on the display.

4.3 Final Concept

The final system consists of a *depth sensor*, a *PC* and a *display*. All of the hardware is placed inside the showcase. In addition, an active tangible to remotely activate and deactivate the system should be developed as well. It was only intended to be a feasibility study, which determines if and how active Gadgeteer-tangibles might be incorporated into the system, later. Suitable components were recommended by me and provided by the museum after to mutual agreement.

The system requires two pieces of software. The first software is of an administrative nature and allows the museum staff to define and maintain the whole exhibition. The second software is presenting the exhibition to the visitors. Previously defined exhibits are selectable.
The exhibition can be defined by museum-staff themselves. Therefore, an exhibition

plane has to be defined and validated first. After that all the exhibits' positions on the plane can be defined and validated. Those positions can be defined in the same way users later interact with the system, by pointing. To exclude a certain inaccuracy when defining a position, it would have to be defined from different angles and validated afterwards. The whole process will be described in Chapter 5 and the technical execution in Chapter 5.3. Furthermore, the corresponding contents such as explanatory texts and detailed images are provided by the staff. Contents and positions can be changed, removed from or reloaded into the exhibition.
When one or more visitors enter the area in front of the showcase the system recognizes them and reacts in an inviting fashion. A defined interaction space enables the user to interact with the system by pointing at an exhibit. No devices outside the showcase are needed.

Functional Specification Document (FSD) The final concept all parties agreed on was written down by me in an FSD and responsibilities were covered by a contract. The document states, which features of the final system must, should and must not be implemented and working.
The necessary features or *must-criteria* where that the the system would have to have separate modes for administration and presentation of an exhibition. The visual feedback of the interaction would be provided by the display. Visitors would be automatically recognized by the system, but only one user at a time would be able interact with it. The whole system would be maintainable by the museum's staff and will start and shut down automatically.
Preferable features *should* be realized, but would not be mandatory. Thus, there should be a system's manual. For guided tours, it should be further possible to switch the system into a 'blind' mode, where it does not react to people. Extensive exhibits should have a slide show. The system should be operable with either the left or right hand. In addition, statistics about the system's use should be logged for later analysis.
There were also criteria that were not requested, and therefore *must not* be implemented. Any free-hand gestures other than pointing must not be recognized by the system. Further, the lighting inside the showcase must not be controlled or influenced by the installation. Feedback has to be only visual and not auditive or haptic. Hence, speakers or tangibles must not come to use.

Furthermore, the FSD describes system requirements concerning hard- and firmwares, data formats and other organizational parameters.

In addition to the FSD, a contract was drawn up by the university's layer's office. It sorted responsibilities and was later signed by the museum's director, my professor and me. Both documents can be found in the appendix.

5 Processes and Setups

In this chapter I will describe the system in detail. This involves the *interaction paradigm*, the *interface designs* and the *technical principles* behind the system.

5.1 Interaction with the IMI-System

Presentation Software One or more visitors can be involved in a typical *presentation scenario*. At the moment they enter the room of the showcase, the IMI-system recognizes them and changes its appearance. The screen displays a short and explanatory text about how the system should be used. In addition, an ideogram visualizes the description by showing a figure pointing at a plane in front of it. This view can be seen in Figure **??**. If there are no visitors present or detected, the screen disguises itself by displaying an image of the showcase's background. Because the system can only be used by one person at a time, there are footsteps on the floor (see Figure **??**) to distinguish the user from other visitors.

F I G U R E

F I G U R E

In [15], Fischer at al. describe a variety of spaces around large, public, and interactive installations. I used an adapted approach to initiate interaction between visitors and our installation. Hence, the area in front of the showcase could be interpreted as an *activation* or *potential interaction space*, because at least the system reacts to the visitors' presence and shows itself. This is the initialization of a possible interaction. Here, the *interaction space* is predefined and thus there is no *potential interaction space* [15]. Other spaces such as *gap* and *social space* are also available (see Figure **??**).

F I G U R E

Once a user enters the interaction space, the system reacts again. This time, the screen will show the outlines of the exhibits inside the showcase. The position of each selectable exhibit is semi-transparently displayed over the outlines. Now, the user can point at exhibits of interest. The system will track a user's movements, and will calculate the position he or she is pointing at. This position is also overlayed on the outlines as shown in Figure **??**. To be distinguishable, exhibits' positions are highlighted in blue, while the current pointing position is red. This visual feedback can be used to correct gestures in order to 'hit' an intended target.

F I G U R E

An exhibit is selected after the user constantly points at it for over half a second. This *dwell time* prevents unintended selections. Otherwise, any exhibit would be immediately selected, whenever a user's pointing position strokes over it. After selecting an exhibit a corresponding description is displayed alongside detailed images of the exhibit (see Figure **??**). While the text remains stable, the images are displayed as a slide show. Afterwards, the system goes back to the exhibition's overview again. The user does not have to end a slide show and can wait until it is over or select a new exhibit while the slide show is still running - without visual feedback though.

F I G U R E

If a user is done or others want to try the interface out for themselves, users can change any time. They only have to switch places on the footsteps. The same applies for ending the session. Visitors can leave at any point during the interaction. As soon as the system does not recognize any visitors around the showcase anymore, it goes back into its disguised appearance again. Again, no additional input devices are needed except people's natural behavior.

Administration Software An *administrative scenario* is more traditional and requires a keyboard and a mouse. Nevertheless, it also makes use of pointing gestures. This software can be used by one experienced staff member alone. But the ones that are less advanced might need a second person's assistance. The administration software can be used to create and edit IMI-exhibitions. Therefore, all necessary data can be defined with it. This data is saved in *configuration files* of XML-format. These files are used by the presentation software and can be reloaded for editing by the administration software.

Expert administrators can make changes with a text editor as well.
An exhibition consists of two main components. Both the *exhibition plane* and the *exhibits* are defined by pointing. Upon the application's start, an existing exhibition can be loaded or a new one can be created by naming it. In case of a new exhibition, the first thing to be defined is the exhibition plane. Therefore, the administrative user again has the choice to either load an existing plane or define a new one. Should the admin decide to define an exhibition plane, the procedure will be explained in a short instruction, before the calibration begins.It tells an admin, to point at three of the planes corners from three different positions. Further, the administrative user is told to follow the instructions during the process. To start the procedure, the administrative user has to be recognizable by the sensor. In addition, ideograms like Figure **??** depicting the positions to go to and the corners to point at are shown during the process. Once the process is started it will guide the admin from position to position and corner to corner in fixed intervals between three and five seconds. After having pointed at the corners once, a second run is needed to validate the corners' positions. This is necessary, because a position could be defined wrong. If the validation is successful, the exhibition and its plane are saved. The second main component are the exhibits. They can only be defined or loaded, if a raw exhibition with a valid exhibition plane is already available. The definition of an exhibit is similar to that of the exhibition plane. The admin points at the respective exhibit on the earlier defined exhibition plane from three different positions. Afterwards, the position is also validated.

F I G U R E

One more necessary setting is the prospective *user position*. It is one of the exhibition's settings. The admin can choose the option to define the user position from the drop-down menu and confirm with a button press. Subsequently, the administrative user gets the instruction to stand where future users are supposed to be in the interaction space. After confirming and a short wait to get into the user position, the software will safe the user position. Only visitors standing close to this spot will be able to use the system. When defining the user position, there should only be one person in the scope of the camera.
The remaining parameters of both exhibition and exhibits have default values, which can be edited later. There are specifically labeled buttons, which lead to the corresponding drop-down menus. Only the exhibition plane cannot be changed this way.

5.2 Interfaces of the IMI-System

Presentation Software There are no conventional control elements like buttons or menus in the User Interface (UI) of the presentation software. It only has the four different states shown in Figure ??.

F I G U R E

The first one is the *standby state*. It is active, if no person in front of the showcase is recognized. The corresponding view's task is to mask the system. Therefore, the screen will only show an image of the showcase's background in order to disguise itself.
The next state gets activated, if visitors are present and none of them is inside the dedicated interaction space. This *introduction state*'s task is to make visitors aware of the systems presence and invite one of them to interact with the installation. As shown in Figure ??b, only a short introduction and an ideogram are needed to explain how the system has to be used. In order to seamlessly integrate into the general style of the museum, the font was chosen to be the museum's corporate font. Further, the background color was chosen to fit with the ambient color of the showcase, and to reduce the contrast between text and background. Moreover, the font size was increased to ensure readability. The associated calculations can be found in the next chapter.
The system gets into its third state as soon as a user is inside the dedicated interaction space. It can be described as an *overview state*. Figure ??c shows the selectable exhibits' positions on top of all exhibits' outlines as blue ellipse and the user's pointing position as a red dot. While the exhibits' positions are always marked, the red *feedback position* only appears, if a user is really pointing at that particular spot of the exhibition plane. Otherwise, the red dot will not be shown.
The fourth state is the actual *presentation state*. It is similar to the introduction, but here the text and images depend on which exhibit had been selected. The corresponding texts are limited to approximately 300 characters for two reasons. Because the screen space of the text is restricted to half the display, readability would suffer for more characters[12]. The second reason for this character restriction is that visitors do not have to read extensively long descriptions. Meanwhile, detailed images of the chosen exhibits

12 Further explanation of this matter can be found in Chapter 5.3.

are displayed on the right half of the screen (see Figure **??**d). They change every four seconds. The time an image is shown can be adjusted either by the administration software of by hand. After all images have been shown, the system will go back to any of the previous states. Where it goes depends on the visitors behavior. If a user is present in the interaction space, the system will go to the overview. If there are visitors present, but no one qualifies as a user, the introduction will be displayed. As soon as every potential user left, the system will go into standby and hide again - even during a presentation.

The administration software is used to manage all data of an IMI-exhibition. Hence, the interface has to be more complex than that of the presentation software. It is more conventional, too. The amount of different views is low, though.
Since it has to be usable by lay persons, similar tasks share a *view pattern*. For instance, settings of an exhibition and an exhibit have identical controls in an identical layout. Only the header of the views and the contents of the drop-down menus vary as can be seen in Figure **??**. The same principle applies for any definition of positions (see Figure **??**). No matter if the corners of an exhibition plane, the position of an exhibit or even the user position are defined, the view is always the same.

F I G U R E

F I G U R E

The interface is designed for task-driven use. This means, that an administrative user probably will have a particular reason to use this software. Such a task includes changing a particular property. Hence, the GUI offers the user to navigate to this property step by step. Therefore, a hierarchical structure is used, which results in less control elements in one view and thus more clarity. There is less choice but a little more actions. Given the example, an exhibit should get an additional image. As shown in Figures **??**a through d, a user loads the exhibition, chooses the exhibit from the drop-down menu, chooses "New Image" from the drop-down menu and copies with a button press. After that, a dialog opens and the user can choose the image to be added. Finally, the image appears in the images' drop-down menu. Images can also be deleted from the exhibit by choosing an existing image and pressing "Delete". From this view, all other properties of the exhibit can be set. By pressing "Back", the GUI goes back to the exhibition's view and the next task can be attended.

F I G U R E

Every time anything is changed, it is immediately saved. There are no undo- or redo-buttons. They are not necessary, because the GUI always shows the current state of all parameters and they can be changed with no more effort than using such a button. Moreover, more control elements could cause confusion. This is why only controls that are required at a particular stage are visible. Nevertheless, there is always the option to go back to the higher level menu.

5.3 Technical Principles

There are certain technical principles, which the systems performance relies on. They range from operations necessary for fundamental calculations to background knowledge involving a proper interface design.

2D ist doof, weil...

Pointing Challenges Visitors can walk up to the showcase and point at exhibits. These exhibits are lying on an exhibition plane. A plane, although usually flat and only 2-dimensional, can be defined by three Points in 3D space. Thus, to define three corners of the exhibition plane, a way to define a point in 3D space was needed. Therefore, I chose the same type of input visitors would later have to use as well. Each exhibition plane's corner is defined by pointing at it from various positions. To calculate where a person is pointing at, the system needs a vector. A vector can be determined by two points in 3D space. One defines the start and the other the direction.

After observing how people point, the elbow was chosen to be the start of the pointing vector. The pointing direction would be determined by the hand[13] as Figure **??** depicts.

F I G U R E

13 NiTE's skeleton tracking provides both of these joints, but does not support the tracking of fingers. More details about the implementation can be found in Chapter 6

In order to define a point in 3D space, one vector was not sufficient. But, if two or more vectors would target the same point from different angles, their intersection should yield the targeted point. Figure **??** shows this concept. Hence, a the point of their intersection could be calculated by equating both vectors. However, this is a 3-dimensional problem. The possibility of vectors intersecting in 3D space is very low. They are mostly skew, and only their projections on a 2D plane do actually intersect. Nevertheless, there is a solution. Vectors may not intersect, but there is an area where they are closest to each other. Consequently, there is one point that is closest to both vectors. This quasi-intersection is called *pedal point*. It lies in the middle of the shortest line that is perpendicular to both vectors [10]. In case both vectors do meet, pedal point and intersection are equal. Figures **??**a and b illustrate how the projected intersection and pedal point can vary.

F I G U R E

F I G U R E

The final point in 3D space gets set by the average of all pedal points. A minimum of two vectors is needed to compute a pedal point. This is prone to error, because one bad vector is enough to corrupt the calculation. Hence, a multitude of pedal points can compensate for such a bad vector. Chapter 5.4 shows that three pointing locations and hence three vectors are sufficient.

Users are pointing at exhibits on a flat, horizontal surface. The distance between users and exhibits can vary and so does the accuracy. The further a target is away the bigger the *angular error* gets. This principle can be seen in Figure **??**. For targets that are further away, the angle of impact on the surface α gets more and more acute. Hence, small variations during the pointing process have a much bigger influence with increasing distance to the target. It is an exponential problem and can be described by the *law of tangents*.

F I G U R E

An exemplary calculation shows how big the angular error can get. The scale is an adult user whose shoulder is constantly a = 70cm above the plane. Further, the user stands an arm's length away from the plane and the hypothetical target is 1.3m away from the edge. This results in a horizontal distance of $b_{target} = 2$m between user and target[14].

$$\tan \alpha_{target} = \frac{b_{target}}{a} \quad \rightarrow \quad \tan \alpha_{target} = \frac{700mm}{2000mm} \quad \rightarrow \quad \alpha_{target} \approx 19.3°$$

The hypothetical target's pointing angle of impact α_{target} is approximately 19°. Due to the fact that humans constantly move their limbs, a user's arm will never be completely still. This movement can alternate one degree in each direction, the pointing under- or overshoots. How much this angular error influences the precision can be seen in the sample calculation.

$$\begin{aligned} \tan \alpha_{point} = \frac{b_{point}}{a} \quad \rightarrow \quad b_{point} &= \frac{a}{\tan \alpha_{point}} \\ b_{short} &= \frac{a}{\tan \alpha_{short}} \quad \rightarrow \quad b_{short} = \frac{700mm}{\tan(18.3°)} \approx 1892mm \\ b_{long} &= \frac{a}{\tan \alpha_{long}} \quad \rightarrow \quad b_{long} = \frac{700mm}{\tan(20.3°)} \approx 2116mm \end{aligned}$$

$$\Rightarrow \quad \delta_{short} \approx 108mm \quad and \quad \delta_{long} \approx 116mm$$

This exemplary calculation shows that a variation of $\pm 1°$ can cause an error of nearly 120mm in pointing direction. This led me to the conclusion that, if a person can not point with pin point accuracy, the system does not have to be calibrated to this accuracy. We would need to define a *sufficient accuracy* though. The angular error would be bigger for shorter users, such as children. It would additionally increase for targets further away. Nevertheless, to define the position of an IMI-exhibit as explained earlier, it has to be defined and validated. This means that both positions have to be within a threshold determined by sufficient accuracy. A radius of 100mm[15] has been chosen as threshold and thus the aspired sufficient accuracy for this system.

Yet another conclusion was to not calibrate each position to its correct position in relation to the depth sensor's coordinate system and measure if users come close to them. It is counter-intuitive to force a user to re-adjust to hit those positions. Instead, I chose to use the corresponding positions that were defined by users themselves. To hit a target,

14 Here, the shoulder is taken as the rotational joint for illustrative reasons. However, the principle remains the same. Further, this configuration will reappear in Chapter 5.4.

15 A radius of 100mm matches the dimensions of a football, with the intended position in its center.

users do not need its accurate position in relation to the depth sensor, they need the accurate position of where everybody else is pointing at. Hence, all positions are *virtual representations* of the real positions. Relations do not have to be correct, as long as the majority of users hit their designated target. This principle is illustrated by Figure **??**.

F I G U R E

Readability Any user should be able to read the instructional and explanatory texts displayed by the installation. Therefore, the font size and the time a text is shown had to be taken in consideration.

F I G U R E

The minimal visual angle at maximum contrast is one minute of angle or 0.017° [43]. This only describes the ability to differentiate between a white and black dot. Readability involves whole letters and therefore needs a bigger visual angle. For people with normal vision this means five minutes or 0.083° of visual angle [ibid.]. This value only applies for people with normal vision, which means they are able to recognize a single letter 20 feet away (see Figure **??**). People with normal eyesight have 20/20 vision. If people have 20/10 vision, they need to be 10 feet away to recognize the particular letter. A vision of 20/15 is more realistic for an average population [ibid.]. This principle can be applied to the Haßleben-installation as well. Given a distance from eye to letter of $b_{20/20} = 2.8$m and the visual angle $\alpha_{20/20} = 0.083°$, the height of a recognizable letter would be $a_{20/20}$ and calculated as follows:

$$\tan \alpha_{20/20} = \frac{a_{20/20}}{b} \quad \rightarrow \quad a_{20/20} = b \cdot \tan \alpha_{20/20}$$
$$a_{20/20} = 2800mm \cdot \tan(0.083°) \approx 4.1mm$$

Those 4mm are equivalent to a font size of 25[16]. A text of this size is unreadable for the majority of the population though. Hence, I considered people could have bad eyesight or no glasses at hand and recalculated the font size for 20/10 and 20/5 vision. Since the tangent is not linear, the corresponding visual angles $\alpha_{20/10}$ and $\alpha_{20/5}$ could not be

16 The system's display has a resolution of 159dpi. Thus, font size[pt] := $a_i \div (\frac{24.5mm}{159dpi})$

doubled or quadrupled. They had to be calculated according to the definition mentioned above.
$with\ b_{20/10} = 2800mm \div 2 = 1200mm \quad and \quad b_{20/5} = 2800mm \div 4 = 600mm$

$$\tan\alpha_{20/10} = \frac{a_{20/20}}{b_{20/10}} \quad \rightarrow \quad \tan\alpha_{20/10} = \frac{4.1mm}{1200mm} \quad \rightarrow \quad \alpha_{20/10} \approx 0.171^\circ$$
$$\tan\alpha_{20/5} = \frac{a_{20/20}}{b_{20/5}} \quad \rightarrow \quad \tan\alpha_{20/5} = \frac{4.1mm}{600mm} \quad \rightarrow \quad \alpha_{20/5} \approx 0.343^\circ$$

Those visual angles can be used to calculate the correct sizes of letters under 20/10 and 20/5 conditions.

$$a_{20/10} = 2800mm \cdot \tan(0.171^\circ) \approx 8.3mm$$
$$a_{20/5} = 2800mm \cdot \tan(0.343^\circ) \approx 16.8mm$$

The corresponding font sizes are 51 and 105. Unfortunately, neither of the calculated font sizes satisfied our requirements. Texts with a font size of 51 was not well enough readable, but with a font size of 105 there was insufficient room for a proper instructional or descriptive text. We compromised on a font size of 70, which allows for 300 characters and is still well readable.
Font sizes have to be adjusted for displays with other resolutions or sizes accordingly. Also different distances between visitors and display have to result in adaption of the font size.

Another factor influencing the readability of texts is the timespan for which they are displayed. If a text is shown for too long, visitors will get impatient, but if it is displayed too short they will get frustrated, because they could not finish reading. Hence, average reading speed should dictate for how long a text is shown. The average reading speed for German texts is 250 words per minute (wpm) [32]. With an average length of 5.7 characters per word in the German language [12], this results in 1425 characters per minute (cpm). With a maximum of 300 characters per description, an exhibit's presentation should take no longer than 13 seconds. Not every exhibit is described by a text with exactly 300 characters. In case of a shorter text, the presentation should be shortened accordingly.
However, there are also images in an exhibit's presentation. To observe an image, we

estimated a default timespan of four seconds. Unfortunately, the system does not know, whether a user reads the text or looks at the images. Hence, there is no way of knowing when to change an image during the slide show. Consequently, the time to inspect those images should be added to the time to read the description. In this case, extensive exhibit presentations would take a long time, which could make visitors impatient. Therefore, an exhibit's presentation takes the maximum time of either the reading task or the slide show. A user can select the exhibit again, if there is further interest in the exhibit.

5.4 Iterative Development

Before anything could be installed or evaluated, the aforementioned principles had to be implemented and their reliability tested. Therefore, I researched suitable environments for an extensible system. Because most SDKs for PrimeSense's hardware are implemented in C++ or C# and Gadgeteer uses Microsoft's .NET framework and C#, the final system should be implemented in C#. However, an SDK written in C# was to be found. After having tried several open source frameworks, the Full Body Interaction (FUBI) developed at Universität Augsburg proved to fit our needs best. FUBI came with a C#-wrapper, which incorporated all functionality of OpenNI and NiTE that was necessary to achieve our goals. Moreover, its leading developer, *Dipl.-Inf. Felix Kistler*, kindly explained how to incorporate FUBI to our new system.
FUBI provides a mechanism to access the coordinates of every tracked persons joints. NiTE tracks and calculates the positions of 25 joints for each person. Further, those joints can be separately updated in real time. Hence, it is possible to only get the positions of necessary joints. This reduces requests and consequently saves performance. Details about the implementation of updating joint positions and further use can be found in Chapters 6.2 and 6.3.

First Test The first test was designed as a proof of concept. The basic feasibility of the aforementioned principle of pointing gestures and the gestures' accuracy were tested. Therefore, a subject had to define a point in 3D space.

The setup that was used for the first test can be seen in Figure **??**[17]. The sensor was positioned at about the same position it would later be inside the Haßleben-showcase. Centered in front of it, a turquoise token was attached to the white surface of the *lab setup*. It has a diameter of approximately 2cm. 18 subjects, who ranged from 160cm to 188cm in height, took part. All of them are either students or researchers of the faculty of media at Bauhaus-Universität Weimar.

F I G U R E

The definition of a point in 3D space was tested with each subject. Only one Independent Variable (IV) was changed throughout a session. It was the amount of positions from where the point had to be defined. Five rounds of defining the token in 3D space for both three and five positions were done by each subject. They were shown the token and told to point at it from three and later five positions. Moreover, subjects had to point with their right arm. The positions were marked by metal braces along the surface's edge opposite the sensor. Subjects were not given any visual aid, but an audible countdown in form of beeps was provided instead. At the end a subject either had to point at the target or move to the next position. Subjects had to point straight at the target for one second. During this second of pointing, ten pointing vectors were recorded[18].

As expected, subjects defined points in 3D space by pointing at the given target. However, there were difficulties concerning the accuracy in both conditions. Although histograms of the acquired data (see Figures **??**a and b) suggest a normal distribution, according to the Kolmogorow-Smirnow test this is not the case. There are too many outliers and hence, the defined points are not normally distributed.

F I G U R E

As it turned out, the kind of distribution was not relevant for the further analysis of the outcomes. As depicted by Figure **??**, there was a dense cloud of points for each condition. In the center of each cloud lay the average point for each condition. Unexpectedly, the values for three pointing locations are more densely distributed, than those for five. This impression can be supported by the similar means and medians, but different Standard Deviation (SD) listed in Table 5.1, 5.2 and 5.3 for three and five positions.

17 The exact specifications are described in the next section of this chapter.
18 The sensor has a sampling rate of up to 60Hz.

F I G U R E

Positions	X-axis	Y-axis	Z-axis
3	60.06mm	-674.73mm	604.28mm
5	81.93mm	-666.20mm	669.74mm

Table 5.1: Means for three and five pointing positions on each axis.

Positions	X-axis	Y-axis	Z-axis
3	52.08mm	-681.75mm	566.67mm
5	41.05mm	-659.33mm	577.18mm

Table 5.2: Medians for three and five pointing positions on each axis.

Positions	X-axis	Y-axis	Z-axis
3	98.86mm	155.86mm	203.29mm
5	98,77mm	202.30mm	284.87mm

Table 5.3: Standard deviations for three and five pointing positions on each axis.

Although, means and SDs are no appropriate measures to be drawn from non-uniform distributions, they confirm the observations. Three pointing positions are less prone to error than five. In retrospect, the five positions were closer together. Consequently, the angle between pointing vectors were more acute. Hence, some pedal points had been calculated to be several meters away from the others. This probably led to the higher dispersion in comparison to that of three pointing positions. Accordingly, it was decided that three pointing locations are sufficient to properly define a point in 3D space by pointing. The process still had to be drastically improved though. As Table 5.4 of a *simulated validation* shows, not even 15% of the defined points would have lied within the aforementioned threshold of 100mm with each other. This means, only in 13 out of the 90 cases[19] the target would have been properly defined and validated. Even for twice the threshold, less than half of all points validated each other. Hence, the accuracy was not sufficient enough for all subjects to hit an identical target.

19 18 subjects · 5 rounds per subject = 90 points

Threshold	Validation
100mm	14.43%
120mm	20.40%
140mm	27.57%
160mm	33.31%
180mm	39.58%
200mm	44.69%

Table 5.4: Successful validations for increasing thresholds.

A positive conclusion can be taken from the *validation maps* depicted in Figure **??**, in which all validation attempt's results are visualized by color coding. Full validation on each axis is green, two successfully validated axes are yellow, one is orange and none is red. Each small square is one defined point. The points are sorted in ascending height of their defining subjects. It can be seen that the subjects' height has no apparent influence on the validation. Hence, size does not matter.

F I G U R E

Two more observations occurred during the tests. First, there were several defective measurements recorded in the raw data. Second, most subjects appeared to point too high and thus overshoot. Both observations had to be addressed to improve the accuracy of the definition process.
The first observation could be handled by improving the algorithms that were responsible for sampling the pointing gesture of a subject. In some cases, the subject's joints were not updated properly, which led to zero vectors pointing nowhere. Those vectors are filtered out since by an improved sampling mechanism. Another aspect was the aforementioned countdown. Several subjects got confused on what to do next. Thus, they did not point during sampling or were not ready. The issue was addressed with improved feedback on what to do during the next test.
The second observation however, turned out to be more complex and yet helpful than anticipated. What had been observed, was a common issue in VR-environments. It is referred to as the *eye-hand visibility mismatch* by Argelaguet et al. [3]. They explain that VR pointing techniques can be classified in two groups. Namely, they are *hand*- and *eye-rooted* techniques [ibid.]. In 3D VR-environments not only devices are tracked, but

also users' heads[20]. Pointing devices' rays and users' view angle are calculated according to their tracked position and orientation. The eye-hand visibility miss match describes the problem that not everything that is visible for a user, can be reached by a device's ray. Other objects might occlude the ray, which prevents selection [ibid.]. Argelaguet et al. propose a solution as well. They introduce a *selection* and a *display ray*. The selection ray originates at a user's eye position and has the device's orientation. Any intersection with an object in the scene will be the display ray's point of aim. The display ray provides the visual feedback and originates at the device leading to the point of aim (see Figure **??**). By implication, Argelaguet et al. introduced aiming as an alternative to pointing.

F I G U R E

As a consequence, I incorporated this principle to my approach and added a second vector to the system. Aiming vectors have a subject's head-joint as origin. The orientation is determined by the head- and hand-joint. Thus, only one more joint had to be updated.

Second Test The following test had to show, if the improvements mentioned above have had a positive effect on the systems performance and precision. The setup and task remained the same. Because five positions had been eliminated as a condition, subjects only had to do five rounds from three positions. Both pointing and aiming vectors as well as the computed points in 3D space were recorded at the same time thus ensuring comparability. This time, 19 subjects participated. Their heights ranged from 157 to 198cm. Although several subjects had taken part in the first test, no learning effects were expected due to the intuitiveness of the task at hand.
In addition to the two pointing paradigms being tested, their combination arose as a solution for an earlier consideration. The *angular error* could be corrected by combining points defined by pointing and aiming. The conjunction of pointing and aiming depicted in Figure **??** could increase the percentage of successful validations.

F I G U R E

20 Users wear special 3D-glasses. For a correct stereoscopic view, those glasses are tracked as well to calculate a correct perspective view angle into the scene.

As Table 5.5 shows, the improvements over the first test worked. While the points defined by pointing were successfully validated twice as much as before, aiming was not as efficient as estimated. Points defined by aiming were only about half as often successfully validated than those defined by pointing. However, aiming was still better than the results from the first test. A combination of the two pointing paradigms resulted in a combined point.

Threshold	**Test No.1**	**Pointing**	**Aiming**	**Combined**
100mm	14.43%	29.83%	16.55%	23.21%
120mm	20.40%	39.86%	21.86%	31.88%
140mm	27.57%	50.35%	26.72%	39.48%
160mm	33.31%	63.91%	31.28%	45.91%
180mm	39.58%	65.74%	35.83%	51.82%
200mm	44.69%	70.03%	39.78%	56.52%

Table 5.5: Comparison of successful validations for increasing thresholds.

The visual observation of subjects during the test revealed a bias toward one of the paradigms. This indicates that subjects tend to either point or aim at a target. Hence, the combined point has to be weighted accordingly. In order to achieve a proper weighing of the points defined by pointing and aiming a subject's bias has to be recognized first. A weighing of the classified points can then be applied to gain the biased point in 3D space.

Third Test The last test that took place under lab-conditions, was intended to find the most efficient combination of classification and weighing of points defined by pointing and aiming. Moreover, the three most suitable corners for defining an exhibition plane were chosen according to the results of this final test.

Each of the 24 subjects of this test had to define all four corners of a fictional exhibition plane. These corners were, as depicted in Figure **??**, marked with colored tokens. Similar to the two earlier tests, they had to point at every corner from each of the three previous positions. During a session, they were given textual advice telling them the position to go to or the corner to point at. The sequence of actions was like in the final definition process of an exhibition plane. Subjects went into position and then consecutively pointed at each of the corners. Intervals between changing positions or pointing at the next target were about 4 seconds. All subjects did five rounds, but this time they defined four points instead of one.

F I G U R E

After all data had been gathered, they were processed by a tool to generate all possible combinations. Therefore, point-tuples defined by pointing and aiming were classified and then weighted. The previous combination was computed for comparison as well.

Classifications were either static or biased. *Static classification* implies a balanced weighing. There was a progression in static classification, though. Whereas *direct* classification only computed, whether two points were validating each other for a particular threshold or not, *centered* classification was checking if both points would validate a virtual center between them.

Biased classification on the other hand compares each of the defining points axis. If there is a difference larger than a particular threshold, there is a probable bias toward either of the points. The dominant point cannot be determined by the distance between the two alone. A decisive criterion was needed and the absolute value was chosen[21]. Either the biggest or the *smallest absolute value* of an axis was considered dominant. Hence, it was given a higher weight than the other point's axis' value. The weight-ratios in favor of the dominant value are 60:40, 70:30, 80:20 and 90:10. Another type of weighing is *automatic weight calculation*. Here, the ratio is calculated with respect to the actual distance between the points. Thus, if the distance is just above the threshold the weight-ratio is 60:40. For values further than twice the threshold apart, the maximum ratio of 90:10 is used. All other distances in between yield a respective weighing.

21 A more profound investigation of biased free-hand pointing will be the topic for further research.

Class.	**Weighing**	**Point #1**	**Point #2**	**Point #3**	**Point #4**	$\sum$
Direct	**50%**	3.70%	9.25%	47.97%	28.95%	22.47%
Center	**50%**	17.23%	39.28%	76.21%	59.55%	48.07%
Big	**60%**	19.93%	38.11%	75.47%	61.03%	48.64%
Big	**70%**	21.40%	35.58%	74.41%	62.09%	48.37%
Big	**80%**	21.64%	32.10%	73.79%	61.46%	47.25%
Big	**90%**	21.75%	29.60%	72.85%	59.74%	46.98%
Big	**Auto**	19.56%	26.49%	73.59%	58.72%	44.59%
Small	**60%**	15.77%	39.14%	77.35%	59.43%	47.92%
Small	**70%**	13.87%	38.03%	77.39%	57.98%	46.82%
Small	**80%**	12.03%	36.58%	76.88%	56.38%	45.47%
Small	**90%**	10.31%	34.37%	76.06%	54.03%	43.69%
Small	**Auto**	13.13%	37.56%	77.03%	53.09%	45.20%

Table 5.6: Successful validations for threshold of 100mm. Comparison of classifications and weighing.

As Table 5.6 shows, centered classification alone more than doubled the validations in comparison to direct classification. Here, successful validations were significantly increased for points #1 and #2, which lay in the far left and right corner. Thus, angular error could be drastically reduced for points further away.
Points #3 and #4 were placed at the front edge of the surface. Hence, the angular error was low and these points were defined and validated much more efficient.
The table further shows that both classification types have similar percentages of overall successful validations. Classification by biggest is a little stronger than by smallest absolute value, though. The advantage comes from a better performance in successfully validating the far left corner.
Although "Big60" hast the best overall performance, "Big70" was chosen for the final version of the administration and presentation software. The successful validations by this combination of classification and weighing show a more homogeneous distribution.
The aspired majority of successful validations within a threshold of 100mm could not be achieved for all corners of an exemplary plane. Pointing accuracy close to a subject is well, though. Nevertheless, subjects were not provided with any feedback about the position they were pointing at and only relied on proprioception alone. Hence, when users are provided with feedback and can adjust their gestures accordingly, the system

is operable with sufficient accuracy[22].

5.5 Development Setups

Three installations were build. One lab setup for development, one makeshift setup was placed in the faculty building's lobby, and the final one was installed inside the showcase at the Museum für Ur- und Frühgeschichte Thüringens. The various setups differed in dimensions and were run with different hardware configurations. Testing of technical principles and computations were conducted with the lab setup. The lobby setup was used for a stress-test during an open door-event at the faculty, whereas the final evaluation took place in the museum. Only the presentation software was evaluated.

The Haßleben-showcase was measured. The measurments were then transferred into the groundplan depicted in Figure **??**. With a width of 439cm and a depth of 344cm, a mockup of the showcase was quiet large and would need an even larger room to fit in and to be still operable. The showcase has a floor-to-ceiling height of 293cm and its exhibition plane is 65cm above the floor.

F I G U R E

The laminated glass is 7mm thick. A main concern at the beginning was, whether the sensor would be able to work through glass or not, because the panels have no anti-glare coating. Figure **??** shows that it worked through two sheets of glass at an acute angle (a) as well as straight trough it (b).

F I G U R E

Lab Setup A room with adequate dimensions had to be found. Moreover, the mockup had to be build, and equipped with all necessary Hardware.
There were three unoccupied rooms. One at the museum and two at the faculty. The room at the museum was the attic, not insulated, and had no network connection. Hence

22 Two actual exhibits in the Haßleben-showcase were defined about 7cm from each other and are distinguishable by users of the presentation software. Further improvements of the determination of points in 3D space are discussed in Chapter 9.

it was not suitable. Both rooms at the faculty could fit the mockup and an additional desktop to work at. The smaller room was reserved, though. Thus, the big room became the testing environment. It provided constant lighting and helpful features such as access to the internet and proximity to experts at the faculty.
Until the final hardware was acquired by the museum and available for testing, similar hardware was preliminarily lend to me by multiple sources of the faculty. The museum's carpenter fabricated a pedestal consisting of surface plating and feet. The plating is fabricated from four 9mm-press boards. The feet seemed too unstable and thus were replaced with one desk rack for each board as can be seen in Figure **??**.

F I G U R E

The mockup is 315cm wide, 264cm deep and 75cm high. These are not the exact measurements of the showcase. I did not reproduce the whole showcase, but only the exhibition plane within. Therefore, the reproduction is adequately proportioned. The additional height increases the probability of angular error compared to the original. Therefore, the final setup's error is estimated to be lower, because pointing vectors' angles of impact are less skew.
The sensor is mounted on a tripod, which is placed at approximately the position it would later be installed inside the showcase. From there, it faces the subjects at an angle of about 100°. This elevated position is necessary to get reliable readings. Otherwise, a sensor installed at shoulder-height can not properly acquire depth information of occluded joints.
The desktop to the right of the mockup was the main workplace from where the tests were supervised. It provided a good overview of subjects and hardware alike. The computer running the test software was placed on this desktop and so was the screen. For later tests, a second screen was placed beneath the tripod's legs.

There were two hardware configurations used during the development of the system. The first test was conducted on an *ASUS Eee PC 1215B*-netbook [24]. Following tests were conducted on the designated desktop-PC[23].

23 Full specifications of hardware and the development environment can be found in Chapter 6.

Lobby Setup The first test under aggravated conditions was conducted during *Summæry*[24]. Therefore, a makeshift setup was built in the faculty's lobby. It consisted of three tables forming the exhibition plane and a cocktail table, on which the desktop-PC and a tripod with the sensor on top were positioned. The interaction space was defined by markings on the floor. There were three targets - a candy bar, a stack of coins, and a stack of fliers - lying on the plane (see Figure **??**).

F I G U R E

After assembling the hardware and sample exhibits, a sample exhibition was defined with the administration software. It was given a name, the exhibition plane, user position and exhibits' positions along with images and short descriptions were defined. The whole process took only 45 minutes. Subsequently the presentation software was started.
The lobby presented the system with a densely populated environment. Many visitors walked past it or stopped to try it out. The system crashed several times during this ordeal. Two reasons could be identified after having observed the events and reviewing the log-files.
First, the system could have had too many people in its sight. As it turned out, OpenNI was designed for 15 users, yet never tested for more than three [11]. It is not clear what happens, if more than 15 subjects are recognized and therefore, I tried to reproduce the error. Unfortunately, we could not manage to get more than 12 persons recognized by the system, whereas the error did not re-occur.
The other problem appeared after having seen some log-files, which implied that non-existent targets were selected by the system after people left. This smaller bug was fixed immediately by checking if a selected target was referring to a valid IMI-exhibit.

5.6 Museum Setup

Figure **??** depicts the installation of the final setup at the Haßleben-showcase. The aforementioned desktop-PC was attached behind the maintenance door to the right of the showcase. Because the keyboard and mouse necessary for maintenance are wireless,

24 Summæry is an open door-event at the faculty of media, where all chairs present their work throughout the faculty-buildings.

the only wires were the USB3.0-cable to connect and power the screen and the PC's own power supply.

F I G U R E

The display's prior position was changed after the museum staff had concerns about it having too much influence on the overall appearance of the showcase. There was a fear of it being too distracting or disturbing the overall picture. Hence, it was anchored on the back wall at eye-height (see Figure **??**c). This compromise was deemed sufficient for the display to still be recognizable in users' peripheral vision when they look inside the showcase.

The footsteps defining the interaction space are not centered in front of the showcase and screen, but shifted to the left. This way, visitors can get closer to the showcase and still read the instructions and look at the ideogram. Moreover, the readability of the display was compromised by reflections in the glass panel of the showcase if users stood right in front of it.

6 Implementation – Interactive Museum Installation

The IMI-system consists of two main parts. First, the hardware, which involves the physical tracking and computing of that data in the background. Second, the software, which includes the IMI-libraries and pieces of software utilizing them.

The hardware consists of and PC, the sensor, a screen and peripheral input devices for maintenance. The designated PC is an *ASUS VIVO VC60-B013M*. It employs a *Intel Core i5 3210M* with a clock speed of 2x2,5GHz and 4GB DDR3 SDRAM [26]. Since the system does not have complex graphics, there was no need for a sophisticated graphics card. Hence, the PC is compact and does not need much electricity.
As a display, we chose an *ASUS MB168B+* with a size of 15.6" and es resolution of 1920x1080 pixels. The display further houses its own graphics chip, which enables it to be connected to a PC via a USB3.0-cable [25]. This configuration eased the installation, because of the smaller plug the holes drilled into the back wall of the showcase could be smaller than for a usual display-cable. The display is supported by a modified bookend, which is attached to the back wall with two screws. An additional hole has been drilled for the cable. This hole is hidden by the display. Although it was planned to camouflage the display with a foil, it did not seem necessary, because lighting at the new position of the display is already dim.

The soft- and firmware used for developing and running the system is based on *MS Windows 7 Professional x86*. As mentioned earlier, the software was included into FUBI. This framework is available as an *MS Visual Studio 2010*-project. The all software is written in *C#* and hence makes use of the *MS .NET Framework 4.5.1* [30] and further *MS XNA Framework 4.0* [31] for certain functions throughout certain regions of the software. Moreover, FUBI utilizes the functionalities of *OpenNI 2.2.0.33* [36] and *NiTE 2.2.0.11* [35].

Maintenance is reduced by automatic booting and shutdown. The system's BIOS boots the PC at 8:30 in the morning. A tool shuts the system down at 4:45 in the afternoon. Hence, no member of the staff has to access the PC to start or shut the system down.

6.1 Libraries

IMI includes a collection of libraries, which can be used to implement interactive applications. At the time the functionality is restricted to free-hand pointing gestures. Other mechanisms can be easily added. An IMI-exhibition is modularly build by those libraries. The Exhibition- and Exhibit-class have structural functionality and manage an exhibition's data. Meanwhile, the Handler-classes use this data to generate and operate the interaction between user and system.

Preliminaries Terms used hereafter are depicted by Figure **??** and defined as follows:

- A point or corner is a point in 3D space of type `Point3D`.
- A plane is defined by three points or corners.
- A position is a point on a plane of type `Point3D`.
- A pointing position is the position a user is pointing at.
- A pointing location it the location a pointing user is standing at.
- A kernel is an exhibit's membership function. It has a size and radius.
- A target is an exhibit's representation on the exhibition plane. It is determined by its position and kernel.

F I G U R E

Exhibition.cs Members of the Exhibition-class are needed to define an IMI-exhibition. All necessary information is contained and managed in this class.

The `exhibitionPlane` is the most essential component of an IMI-exhibition. All exhibits' positions are defined on this fundamental property. It is constructed by three corners. The structure of such a Plane is declared by the GeometryHandler-class. The Exhibition-class further holds a list of `Exhibit`-objects. In this `List` all selectable IMI-exhibits of an IMI-exhibition are stored. In addition, the earlier mentioned user position that defines the interaction space is also saved in this class as a point. An IMI-exhibition's name and file path are members of the Exhibition-class, too. So are the images, which are used to camouflage the display against the background and the overview of the exhibition plane with sketches of exhibits in their respective positions. Furthermore, data crucial for interaction are also members of this class. So are the default values for the threshold needed for defining exhibits and the dwell time for target selection. Timespans used by the presentation software are `lockTime`, which determines a short period of paused interaction after selecting a target and `slideTime` which determines for how long a single image is shown during an IMI-exhibit's presentation.

The characteristic functionality of the Exhibition-class is its get- and set-methods. They allow for reading and writing all those essential variables.

Exhibit.cs The virtual representation of an IMI-exhibit is structured by the Exhibit-class. This includes data relevant for interaction with the system and an IMI-exhibit's presentation upon selection. Like an Exhibition-object, an Exhibit-object stores its own file path as well. This is necessary for organizational reasons.

The most important member of an IMI-exhibit is its position. In combination with the two kernel-defining members `kernelSize` and `kernelRadius` the SessionHandler-class is able to compute which target a user is pointing at.

Members relevant for the presentation of an IMI-exhibit are its name, descriptive text and collection of images. The images are stored in a `Dictionary` as a pair of an image's file path and the image itself.

The Exhibit-class, just like the Exhibition-class, is of structural nature and therefore only has the same administrative methods that get and set members.

FileHandler.cs The objective of the FileHandler-class is *saving* and *loading* data of an IMI-exhibition, its exhibits, and all their properties. Therefore, a temporary Exhibition- and Exhibit-object are needed. Exhibitions along with all related information are saved in *XML-format*. The hierarchical structure of XML-files allows adding, changing and removing elements without much effort.

To save an IMI-exhibition and all its properties several XML-files have to be written. The main file contains the aforementioned defining properties of the exhibition. This means that each IMI-exhibit is stored separately. It is saved as an XML-file as well. Each IMI-file contains all properties as attributes or *forwarding file paths*. Equally, corresponding images are saved as file paths leading to the actual image. The exhibition plane is also separately saved as an XML-file.
Loading works in reverse to this procedure. As en IMI-exhibition is being loaded, its properties are either saved into a *temporary Exhibition-instance* or forwarding file paths are followed. A *Temporary Exhibit-instance* is used to load all of an IMI-exhibits.
The modular composition allows the IMI-system to be flexible. IMI-exhibits exist as independent entities and can be easily removed from or added to an existing IMI-exhibition. The same applies for the exhibition plane. Figure **??** depicts the whole data-structure behind an IMI-exhibition.

F I G U R E

An additional function of the FileHandler-class is reading TXT-files. This feature is used to load pre-written descriptions of IMI-exhibits.

GeometryHandler.cs The GeometryHandler-class is solving essential computations of analytic geometry-problems. Therefore, the two structs `Vector` and `Plane` are declared, because standard types of C# do not deliver the complexity needed for those computations.
A `Vector`, in contrast to a standard `Vector3D` or `Vector3`, is not only a *direction vector*. It also has a *starting* and a *terminal point*. Hence, a `Vector` V is defined by a starting point P_S and either an endpoint P_E or a direction vector leading towards it $\overrightarrow{d_E}$, where

P_S is either a user's tracked head- or elbow-joint and P_E is its tracked hand-joint.

$$V = P_S + \lambda \cdot \overrightarrow{d_E}$$
$$P_V = P_S + \lambda_P \cdot \overrightarrow{d_E}$$

As the factor λ is varied, any point P_V along a `Vector`'s direction of propagation can be expressed.
A `Plane` is defined by one starting corner C_1 and either two additional corners C_2 and C_3 or two additional `Vectors` $\overrightarrow{d_{C_2}}$ and $\overrightarrow{d_{C_3}}$ towards those corners. In short, a `Plane` E is defined by two `Vectors` V_{C_2} and V_{C_3} of identical origin.

$$E = V_{C_2} + V_{C_3} \quad \rightarrow \quad E = C_1 + \lambda \cdot \overrightarrow{d_{C_2}} + \mu \cdot \overrightarrow{d_{C_3}}$$
$$P_E = C_1 + \lambda_P \cdot \overrightarrow{d_{C_2}} + \mu_P \cdot \overrightarrow{d_{C_3}}$$

Any position P_E on a `Plain` can be expressed by varying λ_{C_2} and μ_{C_3}.

The GeometryHandler-class has two types on members. First, *input-members* are pointing- and aiming points or positions. Second, *output-members* are lists of classified and weighted points or positions.

However, the main task of this class is to calculate *pedal points* between `Vectors` and *intersections* of a `Vector` with a `Plane`.
A pedal points is calculated to define a point. Therefore, a subject has to point at this point in 3D space from two different locations. Vectors from both locations should meet in the point to define. Intersections in 3D space are very unlikely, though. Hence, a substitute point has to be found. The closest two skew vectors come to each other is in the pedal point. Thus, this point is computed instead of an intersection. Given two `Vectors` V_A and V_B with starting points S_A and S_B and their respective directions $\overrightarrow{d_A}$ and $\overrightarrow{d_B}$.

$$V_A = S_A + \lambda_A \cdot \overrightarrow{d_A} \quad and \quad V_B = S_B + \lambda_B \cdot \overrightarrow{d_B}$$

A normal is a vector, which is perpendicular to its original vector. To get the shortest distance between two `Vectors`, their normals $\overrightarrow{n_A}$ and $\overrightarrow{n_B}$ have to be perpendicular to both `Vectors`.

$$\overrightarrow{n_A} = \overrightarrow{d_A} \times (\overrightarrow{d_A} \times \overrightarrow{d_B}) \quad and \quad \overrightarrow{n_B} = \overrightarrow{d_B} \times (\overrightarrow{d_A} \times \overrightarrow{d_B})$$

To compute the pedal point, and stay in the metaphor, the two foot points F_A and F_B are needed. These points lie on either `Vector` and in between them lies the pedal point. Hence, the factors λ_{F_A} and λ_{F_B} have to be altered.

$$
\begin{aligned}
F_A &= S_A + \lambda_{F_A} \cdot \overrightarrow{d_A} && \textit{and} \quad F_B = S_B + \lambda_{F_B} \cdot \overrightarrow{d_B} \\
F_A &= S_A + \frac{(S_B - S_A) \bullet \overrightarrow{n_B}}{\overrightarrow{d_A} \bullet \overrightarrow{n_B}} \cdot \overrightarrow{d_A} && \textit{and} \quad F_B = S_B + \frac{(S_A - S_B) \bullet \overrightarrow{n_A}}{\overrightarrow{d_B} \bullet \overrightarrow{n_A}} \cdot \overrightarrow{d_B}
\end{aligned}
$$

The only step left is to calculate the pedal point P_{AB} between the two foot points [10] F_A and F_B. Since the feet already represent the points on each `Vector` that are closest to the other `Vector` and P_{AB} is in the middle, P_{AB} is the average of them.

$$P_{AB} = (F_A \cdot F_B)/2$$

An exemplary configuration of the problem is depicted by Figure **??**.

F I G U R E

The IMI-system's interaction is based in the principle of intersections of a `Vector` and a `Plain`. Positions of IMI-exhibits are defined this way. Furthermore, a user's pointing and aiming position is calculated exactly the same. Thus, this computation is crucial for interaction. Given a `Vector` V and a `Plane` E are not parallel, they intersect in a position P_I, where $P_I \in V \wedge E$.

$$
\begin{aligned}
V = P_S + \lambda \cdot \overrightarrow{d} \quad &\rightarrow \quad P_I = P_S + \lambda_I \cdot \overrightarrow{d} \\
E = C_1 + \mu \cdot \overrightarrow{d_{C_2}} + \nu \cdot \overrightarrow{d_{C_3}} \quad &\textit{with} \quad \overrightarrow{n_E} = \overrightarrow{d_{C_2}} \times \overrightarrow{d_{C_3}}
\end{aligned}
$$

A `Vector` $\overrightarrow{u}$ between P_I and any other position on E has to be perpendicular to E's normal $\overrightarrow{n_E}$. Further, a `Vector` $\overrightarrow{w}$ from V's origin to any position on E is needed. It describes V's origin with respect to E. In both cases, C_1 can be taken as a known position on the `Plane`.

$$\overrightarrow{u} = P_I - C_1 \quad \textit{and} \quad \overrightarrow{w} = P_S - C_1$$

As Figure **??** shows, going from C_1 to P_I by $\overrightarrow{u}$ is the same as by $\overrightarrow{w}$ and then along $\overrightarrow{d}$. Since $P_I \in E$, $\overrightarrow{n}$ also stands perpendicular on $\overrightarrow{u}$. Hence, it is also perpendicular to its equal path from C_1 to P_I.

$$\begin{aligned}\overrightarrow{u} = \overrightarrow{w} + \lambda_I \cdot \overrightarrow{d} \quad \textit{and} \quad & 0 = \overrightarrow{n_E} \bullet \overrightarrow{u} \\ \Rightarrow \quad & 0 = \overrightarrow{n_E} \bullet (\overrightarrow{w} + \lambda_I \cdot \overrightarrow{d})\end{aligned}$$

This equation can be solved for λ_I, which is then put back into the pointing `Vector`'s equation.

$$\begin{aligned}\lambda_I = \frac{\overrightarrow{n_E} \bullet \overrightarrow{w}}{\overrightarrow{n_E} \bullet \overrightarrow{d}} \quad \textit{in} \quad & P_I = P_S + \lambda_I \cdot \overrightarrow{d} \\ \rightarrow \quad & P_I = P_S + \frac{\overrightarrow{n_E} \bullet \overrightarrow{w}}{\overrightarrow{n_E} \bullet \overrightarrow{d}} \cdot \overrightarrow{d}\end{aligned}$$

F I G U R E

The GeometryHandler-class utilizes the MS XNA framework. It features specialized analysis functionality such as computation dot and cross products and the normal of a plane. These functions replaced manual calculations and increased the efficiency of the class.

The third feature of the GeometryHandler-class is classifying and weighing points and positions. As mentioned in Chapter 5.4, the focus of this work is not the investigation of *biased free-hand pointing*, but the development of an interactive system using free-hand pointing.
Pointing and aiming points and positions are classified by comparing each axis' values separately. In the current state, the biggest absolute value is considered the dominant value. The combined point or position is then computed by weighing the two inputs 70:30 in favour of the dominant value.

CalibrationHandler.cs Points, planes and positions of an IMI-exhibition are defined and validated by the CalibrationHandler-class. Therefore, two members are used. The first stores the samples per position from which an administrative user is pointing. It is needed to organize the inputs of aiming and pointing `Vectors`. The other member saves the threshold for which points, planes and positions are defined and validated.

To define a `Plane`, three corners are needed. Those corners are points in 3D space. A Point is defined by pointing at it from different pointing locations. This way `Vector`s from various angles define what can be described as a *cloud of intersections*. They are pedal points between every `Vector` with every other `Vector` and computed by an instance of the GeometryHandler-class . Since there are three pointing locations, there are also three `Vector`s for each corner. Hence, each corner's cloud consists of three pedal points. These pedal points are averaged to form a center point.
This process is repeated to validate the once defined corners. Therefore, an administrative user defines the same corners again and the previously defined corners are compared to the newly defined validation corners by checking, whether they lie within the saved threshold. If every pair of corners validates each other, the `Plane` can be saved as the corresponding member of the IMI-exhibition.

Defining the position of an IMI-exhibit is similar to the definition of a corner. An admin points at the exhibit on the earlier defined exhibition plane. This is done from three different pointing locations. Only this time the GeometryHandler-instance does not calculate a pedal point, but the `Vector`s' intersections with the `Plane`. After that, a center of the three points is computed.
A position on an IMI-exhibit has to be validated as well. Hence, the defining processes is repeated and the two centers are compared according to the threshold of the respective IMI-exhibition. If they validate each other, the position is saved for the particular IMI-exhibit.

Should corners or a position not successfully validate, the whole definition process has to be repeated all over again. Another validation round is not sufficient, because the first defined corners or position may be flawed.
The definition and validation for both corners and positions are done for pointing and aiming. Thus, there are always two values for each corner or position. These two get classified, weighted and combined into one corner or position after validation. Hence, the reliability of a corner or position is higher, because only validation of both pointing modes yield validation of the combined corner or position value.

SessionHandler.cs All data necessary for interaction is computed by the SessionHandler-class. It determines a user from surrounding visitors and calculates whether a target is pointed at or not. Furthermore, this handler also calculates a user's *feedback position* on the exhibition plane. Thus, a user is able to adjust his or her way of pointing.

For all these calculations the SessionHandler-class needs various members. To determine who is a user amongst visitors, the class stores the predefined user position of an IMI-exhibition as a member upon initialization. Along this particular point it further saves the radius of the interaction space around the user position.
Additional members that are necessary for interacting with the IMI-exhibition are the exhibition plane, the screen and canvas[25] size. The size of the feedback buffer is a constant member for smoothing a user's feedback position on the display.

A short but important function is the determination of a user. Without the identification of a user, all further interaction is not possible. All trackable visitors recognized by the system have an Identifactor (ID). A `Dictionary` with this ID as `Key` and the respective person's hip-joint position as `Value` is given to a function. This function checks if any of the visitors' hips is within the interaction space defined by the `userPosition` and `radius`. It then returns the ID of the user closest to the IMI-exhibition's user position. Hence, the software knows the ID of the user to further track and hence interact with.
Intersections of `Vector` and `Plane` are calculated by the GeometryHandler, but the intersection is not enough. It has to be checked, if some target was hit or not. This task is done by the SessionHandler. It is very hard to hit just the position of an IMI-exhibit. This is where the aforementioned kernel of an `Exhibit` comes into play. As depicted by Figure **??**a the kernel defines an *Area of Affinity (AOA)*. The closer an intersection is to the center of the AOA, the higher the affinity for its corresponding `Exhibit`. Thus, the actual size of an IMI-exhibit does not influence its selectability. Only its AOA's properties determine how well an IMI-exhibit is selectable. Especially for densely arranged exhibits, this is a useful feature. A small but important object can be given a big kernel size and radius and hence attract more selections than a less important object in its direct proximity (see Figure **??**b).

F I G U R E

To select a target, it has to be determined which AOA has the strongest affinity for

25 A canvas is an XAML-element on which the overview, exhibits' positions and later the feedback position are drawn. More about the process in Chapter 6.3.

the position in question. Therefore, the affinity of each `Exhibit` at this position has to be calculated and it has to be distinguished which one's is the highest. To do that for th ewhole exhibition plane and all exhibits on it in real turned out to require a lot of computing power. Hence, the efficiency was improved by using a *pre-processed lookup-table of all AOAs*. Therefore, the `Plane` is rasterized by 1000 steps in each direction. In this case each step between the matrix dots is about 3mm. Then, the all affinities for each of the matrix dots are calculated. Subsequently, the index[26] of the `Exhibit` with the highest affinity is stored in a *lookup table*. This lookup table is a `Dictionary` with a position as `Key` and the corresponding `Exhibit`'s index as `Value`. After all positions are processed, all positions referring to no IMI-exhibit are removed from the lookup table. Thus, in this case, the number of *pre-determined affinities* is reduced from 1.000.000 to about 50.000. Thereafter, affinities do not have to be computed each frame, but positions are compared[27] with those in the lookup table. If there is a match, the corresponding index is returned.
Every position a user is pointing at is buffered and a smoothened *feedback position* is returned. This feedback position is already converted in `Canvas`-coordinates to match the dimensions of the overview. As Figure **??** shows, `Plane`-coordinates have to be converted into `Canvas`-coordinates with the ratios of `Plane` to `Screen` to `Canvas` in mind. Depending on the ratio of the `Plane`, the ratio between `Screen` and `Canvas` changes.

F I G U R E

The `planeCanvasRatio` is used to calculate the correct position for the representation of the feedback position in an IMI-exhibition's overview. Moreover, it is also used to correctly scale and display the AOA of each `Exhibit` it in the defined position on the overview.

StatisticsHandler.cs Statistical calculations are done by the StatisticsHandler-class. It computes the *mean*, *variance*, *(empiric) standard deviation* and standard error for a sample of `Point3D` or `double`-values. This feature was used for informal analysis of raw

26 All `Exhibits` of an `Exhibition` are stored in a `List`.
27 Pointing positions might not appear in the lookup table. Therefore, the closest position is interpolated.

data during the iterative development-process. Nevertheless, it might be used for more representative long-term evaluation.

DataLogger.cs The DataLogger-class can be used for saving all kinds of data of the IMI-system. It was developed to write log-files in TXT-format. Therefore, it is possible to initiate an instance with only a file path. After that, new paragraphs can be set up with either a headline or no headline. These paragraphs can then be addressed with an index. Thus, new lines are addable at any point during the runtime of an IMI-application. A logs can be saved as TXT-files anytime, too.
The DataLogger-class was used during the iterative development-process to log pointing and aiming `Vector`s as well as defined points and positions in one file for each subject along with related statistics of these values.
The presentation-software, uses the DataLogger-class to save all events during a session. Thus, the museum is able to keep track of its visitors' behavior and interests. These session-logs can be further analyzed by the IMI-statistics tool described in Chapter 6.5.

6.2 Administration Software

An IMI-exhibition and its exhibits are defined with a special tool. The software furthermore enables an administrative user to edit any IMI-exhibition or its exhibits. Therefore, the administration software incorporates the functionality of IMI-system's the depth sensor in order to define points and positions. External files such as images and texts can be loaded to enrich IMI-exhibits. Moreover, properties crucial for interaction are also individually adjustable.

To define any point or position, the depth sensor's tracking date has to be usable in real time without compromising the GUI. Thus, the sensor is only activated when it is needed. For the rest of the time, it is shut off. If tracking data are needed, a special thread is started. The tracking-thread is the standard `while`-loop provided by the FUBI framework. During each cycle of the loop the depth sensor updates its tracking data. This data is then accessible by the GUI-thread via functions of the FUBI framework. However, this is not enough to define an exhibition plane or user position of an IMI-exhibition or the position of an IMI-exhibit. For defining any of these yet another thread is needed to make use of the tracking data. Therefore, the `calibrationThread` is used. It is one thread that starts either of the definition-methods mentioned above. Namely they are `definePlane`, `defineUserPosition` and `definePosition`. The definition of the exhibition plane and the definition of an exhibit's position both work quiet similar, whereas the definition of the user position is less complex. For the latter, an administrative user stands at the designated user position and the coordinates of the admin's hip-joint are saved. Defining a position on the exhibition plane or the plane itself is a more involving process.

To define an exhibition plane, an administrative user has to be trackable. Otherwise, the button to start the process is not visible and thus cannot be pressed. When everything is in order and the button has been pressed, the `calibrationThread` calls the `definePlane`-function. This function is responsible for initiating the sampling-function and determine whether the `Plane` is defined for the first time or to validate the first attempt. In both cases, the function `sampleVectors` is called. To run, it needs the amount of points to define, the amount of pointing locations, the amount of `Vector`s sampled per pointing location and the *sampling mode*. This mode determines, which kind of samples are taken[28]. The `sampleVectors`-function then initiates the definition process. Therefore, two nested `for`-loops are passed. Both loops provide the instructions for the admin on what to do next. They are displayed on the GUI along with the corresponding ideograms. The outer loop addresses the pointing locations, while the inner one loops over the points to define. This means that the outer loop does three cycles, one for each pointing location. The inner loop does one cycle for each of the three corners. Hence, an administrative user has to go to a pointing location, point at each of the three corners and the move to the next pointing location and repeat the process until each corner has been pointed at from each pointing location. After the first definition, the whole routine to define the exhibition plane is repeated. For the second time the same exhibition plane is defined in order to validate the first ones proper definition. If either of the corners does not lie within the sufficient accuracy, the whole procedure has to be repeated. The same procedure applies for defining the position of an IMI-exhibit. Only this time, there is only one position to define and not three points. Hence, the

28 Either pointing or aiming `Vector`s are taken. The third sampling mode is to sample both, pointing and aiming `Vector`s simultaneously. The final IMI-system works in this combined sampling mode to compute weighted points and positions as described in Chapter 5.4.

position is pointed at from three pointing locations twice to define and validate its value as described by Figure **??**.

F I G U R E

After each definition the depth sensor is not needed anymore. Hence, both threads can be stopped. First the `calibrationThread` and then the `trackingThread` are aborted. The order is important, because otherwise there will be not tacking data to access and thus memory violations, which might cause the software to crash.

Not only exhibition planes and the position of an IMI-exhibit can be defined with the administration software. The application also enables an administrative user to load particular files to enhance an IMI-exhibition. While all *changes are saved automatically*, external files have to be explicitly loaded. Files that can be loaded include images, text files and whole IMI-exhibits as well.
Images are used as the background for the stand by mode and the outlines of the IMI-exhibition for the navigation mode of the presentation software. Furthermore, any detailed photograph of each IMI-exhibit also has to be loaded manually. Images have to be of BMP-, JPG- or PNG-format.
Instead of writing a description for an IMI-exhibit, a previously constructed text can alternatively be loaded into the `textBox`. All files should be TXT-files and coded in UTF-8 to guarantee flawlessly displayed characters.
It is possible to load already existing IMI-exhibits as well. Even if they where defined in the context of another IMI-exhibition, they can be loaded. Afterwards, they will be a valid part of the administered IMI-exhibition. Nevertheless, certain changes might have to be done to properly include them into the new context. Their position might have to be re-defined in order to be selectable, if the exhibition plane or location of the depth sensor are different that before.

All properties of an IMI-exhibit can be edited with the administration software. This includes the aforementioned properties that can be loaded, but also various other ones. Therefore, a special "Properties" section offers each editable property in a drop down menu. The properties vary in type. One the one hand, there are the radius and size of the kernel. Although these are continuous values, the input is restricted to three discreet states to avoid confusion. They are presented as textual entities "low", "normal" and "high" and hence more imaginable representations of a certain percentage of the default

value. The low value is 50%, the normal value is 100% and the high value 150% of the respective default value. On the other hand, there is the position of the exhibit itself. This value can be set by defining and validating it again like it is described above.
The view displaying the properties' drop down-menu of an IMI-exhibition can be reached with the same "Properties"-button in the lower left corner of the GUI's view of the exhibition. An IMI-exhibition has more properties than an IMI-exhibit, because it has to regulate the interaction. Here, three types of values are editable. First, there are the two images that can be loaded from this view. Next, continuous values are adjustable as "low", "normal" and "high". They are the threshold to validate new positions, the dwell time for target selection, the time the tracking is paused after selectig a target and the time a slide is shown for. The final property is the user position. This value can be set by a person standing in the user position. Like the definition of any other point or position, the process can only be started, if the person is trackable.

6.3 Presentation Software

Annotations

- 'What is the presentation-software?'
  - Display information of previously defined interactive exhibits
  - Overview-map of ExhibitionPlane
  - Feedback of exhibits' positions and pointing position
  - Description (Readability, Sehwinkel) and Images as slide show
- 'What does it do?'
  - Check for Exhibition
  - Pre-calculate Lookup for exhibit-selection (saves processing power)
  - Recognize visitors
  - Identify user by predefined UserPosition

6.4 Presentation Remote

A remote to stop an resume the presentation software during guided tours war requested by the museum's staff. Therefore, it should be possible to switch the sensor off and stop tracking visitors. Hence, the presentation software would stay in its hibernation mode and the display would remain camouflaged.

Unfortunately, not finished by now...

But, hardware present and template project available...

6.5 Statistics Tool

The presentation software saves a log-file of each session for and review purposes. These files can be analyzed with the statistics tool. Therefore, the tool only has two buttons. By pressing "Load", one or more log-files can be loaded into the tool and the amount of loaded files is displayed in a small label. All loaded files get analyzed by pressing "Analyze". Consecutively, the tool reads all files and parses them for relevant information. This process takes a few minutes. After parsing all files general statistics of the combined files are computed. The results are displayed within a label of the tool and written to a TXT-file.

The results include the *involvement relation*. It is the relation between empty and evaluable sessions. In an empty session no interaction with the IMI-system occurred. Inferences about the system's recognizability can be drawn from the involvement relation. A low quotient implies a low rate of interaction with the system and appropriate measures should be taken to increase recognizability.
Afterwards, *relevant times* are analyzed with respect to the longest, shortest and average length. The relevant times are the *initiation time*[29] and the *total length* of a session measured from activation to departure. The first time indicates the visitors' inhibition to use the system, while the latter gives an overview of visitors' involvement with the system.

29 Here, initiation time is the time between activation of the system and the first interaction of a user.

Finally, *exhibit selection* is reviewed. Therefore, two separate values are inspected. The amount of *absolute selections* shows how often each IMI-exhibit was selected over all. The ranking of selections implies visitors' general interest. *Target transitions* are computed as well. A transition occurs when a target does not get selected, but instead another target gets marked. In this case, an *external transition* between the two targets occurs. It might also happen that the pointing position of a user leaves a kernel and re-enters it immediately. Thus, the target also does not get selected, and there is an *internal transition*. Transitions indicate difficulties in selecting certain targets. These targets are either hard to hit on their own or too close to another one. In both cases however, kernels of the IMI-exhibits in question can be altered and/or the positions of those exhibits can be re-defined.

7 Evaluation

7.1 Pre-Study

7.2 Study

Annotations

- Pre- and postcondition of exhibition
- Survey of visitors' behavior prior to system's installation and afterwards
  - Interaction between visitors
  - Interaction with display
  - **LOS!** (**LOS!**)
  - Interviews
  - Evaluation-Forms

Bimodale Verteilung der 1. Stichprobe gegen modale Verteilung der 2. Stichprobe

7.3 Post-Study

8 Discussuion

Annotations

- Conclusions
  - Comparison to Conception
  - Comparison to 'Pflichtenheft' see *Ref: Appendix*
- Anecdotes
  - Very short short-time memory → Instruction-sticker
  - Misconception of screen an a simple video and no interaction
  - Inhibitional factors (shyness, frustration, being watched)

9 Future Work

Annotations

- My work in relation to situation described in chapters 1 and **??**
- Outlook of possible further developments or optimizations of the system
    - Multi-user
    - Mobile devices
    - Audio
    - 3-dimensional positioning of objects and users
    - different possibilities of feedback

Bibliography

[1] Johnny Accot and Shumin Zhai. More than dotting the i's - foundations for crossing-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*, volume 4, pages 73–80. ACM, April 2002.

[2] Ferran Argelaguet and Carlos Andujar. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics*, 37:121–136, May 2013.

[3] Ferran Argelaguet, Carlos Andujar, and Ramon Trueba. Overcoming Eye-Hand visibility Missmatch in 3D Pointing Selection. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology (VVRST '08)*, pages 43–46. ACM, October 2008.

[4] Ferran Argelaguet, André Kunert, Alexander Kulik, and Bernd Fröhlich. Improving co-located collaboration with show-through techniques. In *2010 IEEE Symposium on 3D User Interfaces*, pages 55–62. IEEE, March 2010.

[5] Alec Azad, Jaime Ruiz, Mark Hancock, and Edward Lank. Group Behaviours around Public Displays, 2011.

[6] Richard A. Bolt. Requirement Analysis and Implementation of Palm-Based Multimedia Museum Guide Systems. In *SIGGRAPH '80 Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262–270. ACM, 1980.

[7] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. The Interactive Museum Tour-Guide Robot. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 1998.

[8] Li-Der Chou, Chia-Hsieh Wu, Shih-Pang Ho, Chen-Chow Lee, and Jui-Ming Chen. Requirement Analysis and Implementation of Palm-Based Multimedia Museum Guide Systems. In *Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04)*, 2004.

[9] Luigina Ciolfo and Liam J. Bannon. Designing Interactive Museum Exhibits: Enhancing visitor curiosity through augmented artifacts. In *Proceedings of the Eleventh European Conference on Cognitive Ergonomics (ECCE11)*, September 2002.

[10] Ina de Brabandt. Abstand windschiefer Geraden: Formel. `http://www.ina-de-brabandt.de/vektoren/a/abstand-gerade-ws-formel.html`, December 2014.

[11] DIABLODALE. jit.openni maximum user trackable ... HELP! `https://cycling74.com/forums/topic/jit-openni-maximum-user-trackable-help/`, December 2014.

[12] Duden. Durchschnittliche Länge eines deutschen Wortes. `http://www.duden.de/sprachwissen/sprachratgeber/durchschnittliche-laenge-eines-deutschen-wortes`, September 2014.

[13] Guillermo Fernández and Montserrat Benlloch. Interactive exhibits: how visitors respond. In *Museum International (UNESCO, Paris)*, volume 52, pages 440–444. Blackwell Publishers, October 2000.

[14] Kieran Ferris, Liam Bannon, Luigia Ciolfi, Paul Gallagher, Tony Hall, and Marilyn Lennon. Shaping Experiences in the Hunt Museum: A Design Study. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques (DIS '04)*, pages 205–214. ACM, August 2004.

[15] Patrick Tobias Fischer and Eva Hornecker. Urban HCI: Spatial Aspects in the Design of Shared Encounters for Media Facades. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, pages 307–316. ACM, May 2012.

[16] Patrick Tobias Fischer, Eva Hornecker, and Christian Zoellner. SMSlingshot: An Expert Amateur DIY Case Study. In *Proceedings of the 7th International Con-*

ference on Tangible, Embedded and Bembodied Interaction (TEI '13), pages 9–16. ACM, Februar 2013.

[17] electronics GHI. 5x Breakout Module Set. https://www.ghielectronics.com/catalog/product/405, November 2014.

[18] electronics GHI. FEZ Spider Starter Kit. https://www.ghielectronics.com/catalog/product/297, November 2014.

[19] electronics GHI. Potentiometer Module. https://www.ghielectronics.com/catalog/product/275, November 2014.

[20] electronics GHI. RFID Reader Module. https://www.ghielectronics.com/catalog/product/366, November 2014.

[21] Eva Hornecker and Patrick Tobias Fischer. Interaktion in öffentlichen Räumen. In *Informatik Spektrum*, volume 5, pages 440–444. Springer-Verlag, October 2014.

[22] Eva Hornecker and Matthias Stifter. Learning from interactive museum installations about interaction design for public settings. In *Proceedings of the 18th Australia conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments (OZCHI '06)*, pages 135–142. ACM, November 2006.

[23] I-KEYS. RFID-Transponder 125kHz. http://www.i-keys.de/de/Transponder/125-khz.html, November 2014.

[24] ASUSTeK Computer Inc. Eee PC 1215B. http://www.asus.com/Notebooks_Ultrabooks/Eee_PC_1215B/, November 2014.

[25] ASUSTeK Computer Inc. MB168B+. http://www.asus.com/de/Monitors_Projectors/MB168BPlus/, November 2014.

[26] ASUSTeK Computer Inc. VivoPC VC60. http://www.asus.com/Commercial_Desktops/VivoPC_VC60/, November 2014.

[27] IRISFILM. The Introduction of Polymer Dispersed Liquid Crystal Switchable Film, Specification, and Optical Testing Data. http://www.irisfilm.com.tw/upload/newsfile/wkc0mynr20120922011639.pdf, November 2014.

[28] Maria Karam and M. C. Schraefel. A Taxonomy of Gestures in Human Computer Interaction, August 2005.

[29] Kathleen McLean. *Planning for people in museum exhibitions.* Association of Science-Technology Centers, 4th edition, 1993. `http://books.google.de/books?id=mpHuAAAAMAAJ`.

[30] Microsoft. Microsoft .NET Framework 4.5.1 (Offlineinstaller) für Windows Vista SP2, Windows 7 SP1, Windows 8, Windows Server 2008 SP2 Windows Server 2008 R2 SP1 und Windows Server 2012. `http://www.microsoft.com/de-de/download/details.aspx?id=40779`, November 2014.

[31] Microsoft. Microsoft XNA Framework Redistributable 4.0. `http://www.microsoft.com/en-us/download/details.aspx?id=20914`, November 2014.

[32] J. Musch and P. Rösler. Schnell-Lesen: Was ist die Grenze der menschlichen Lesegeschwindigkeit? In *Kognitive Leistungen: Intelligenz und Mentale Fähigkeiten im Spiegel Der Neurowissenschaften*, pages 89–106. Spektrum Verlag, 2011.

[33] Deutscher Museumsbund. Geschichte & Definition. `http://www.museumsbund.de/de/das_museum/geschichte_definition/ursprung_des_museums`, October 2014.

[34] Thüringer Museumsverband. Liste der Museen in Thüringen. `http://www.museumsverband-thueringen.de/museumsverband/museen/orte.html`, August 2014.

[35] OpenNI. NITE 2.2.0.11. `http://www.openni.ru/files/nite/index.html`, November 2014.

[36] OpenNI. OPENNI SDK. `http://www.openni.ru/openni-sdk/index.html`, November 2014.

[37] Gang Ren and Eamonn O'Neill. 3D selection with freehand gesture. *Computers & Graphics*, 37:101–120, May 2013.

[38] Helga Schneider. Das Herbarium als Illusionsspiel - Eine zeitgenössische Untersuchung zum Einsatz der historischen Dioramen, July 2014.

[39] Ben Shneiderman. *Designing the user interface: strategies for effective human-computer interaction.* Addison-Wesley, 3rd edition, 1998.

[40] Frank Steinicke, Daniel Keefe, Antonio Krüger, Jean-Baptiste de la Rivière, and Hrvoje Benko. Foreword to the special section on touching the 3rd dimension. *Computers & Graphics*, 37:A1–A2, May 2013.

[41] Brygg Ullmer, Hiroshi Ishii, and Robert J. K. Jacob. Token+constraint systems for tangible interaction with digital information. In *ACM Transactions on Computer-Human Interaction (TOCHI)*, volume 12, pages 81–118. ACM, March 2005.

[42] Andrea Witcomb. Interactivity: Thinking Beyond. `http://www.blackwellreference.com/subscriber/tocnode?id=g9781405108393_chunk_g9781405108398327`, September 2008.

[43] Jeremy E. Wolfe, Keith R. Kluender, and Dennis M. Levi. *Sensation and Perception*. Sinauer Associates, Inc., 2012.

[44] Derek R. Wood. The Diorama in Great Britain in the 1820s. *History of Photography*, 17(3):284–295, 1993.