
Rasmus Höfer
and
Bogi Panait Niclasen
Kinect Me!
An investigation into interaction techniques
for Virtual Environments
Master Thesis, May 2011
Medialogy, Aalborg University Copenhagen

This page intentionally left blank.

Preface

The following Master thesis has been written in partial fulfillment of the requirements for the 10th semester at Medialogy, Aalborg University Copenhagen in spring 2011.

The thesis builds on research done in the area of interaction techniques within Virtual Environments. The driving force of the project was to explore the interaction potentials of the Microsoft Xbox Kinect camera and NITE tracking framework.

For my wife Corina, my joy and support through life.

Bogi

Rasmus Höfer

Bogi Panait Niclasen

This page intentionally left blank.

Readers guide

This thesis is divided into six main parts that will be outlined in the following. After a brief introduction to the subject, describing the motivational drive for this project, the first part *Pre-analysis* will introduce the research in the area of interaction techniques in Virtual Environments (VE). Further research covering principle-based design and other design related topics is presented. The concluding remark to this part is that selection techniques in VE shall undergo further scrutiny.

The second part *Analysis* has two main areas of focus; the first is a systematic task decomposition of selecting objects in a VE, this taxonomy will lead to a construction of a design model of selection techniques. Secondly the model will be regarded in respect to user oriented design principles that will mount out in a design requirement specifications for selection techniques.

Design and Implementation gives details into how the five selection technique prototypes were designed and implemented in regard to the findings in the previous analysis.

The fifth part *Test* focuses on the final summative evaluation that was conducted on the five selection techniques. The methodological approach, practical underpinnings and results of the evaluation will be presented in this chapter.

The final part *Discussion* and *Conclusion* discusses the evaluation results and takes a more general reflection in regards to the constructed selection techniques. This part is concluded by a list of design guidelines for selection techniques in Virtual Environments.

Note that since there is a large amount of terminology introduced in the thesis, a foldout glossary is attached to this report.

The source code to the constructed application and the results from all evaluations can be found on the attached CD.

Table of Contents

Part V - Test

Part VI - Discussion & Conclusion

1 Introduction

The development of immersive virtual environments (VEs) debuted in the late 1960s, when Ivan Sutherland presented his prototype using real-time 3D rendering and a tracked head-mounted display. Since then, VEs has developed and been applied through a vast range of application areas, such as simulations, architecture, psychotherapy, entertainment, data visualization and education. One important aspect of VEs is the user interface (UI), through which the user interacts with the application. Such interfaces make use of various sensing (input) and output systems, which during the last years have become affordable i.a. through the availability of depth sensing cameras, lightweight head-mounted displays or projectors. These interfaces may include the users head motion, hand motion and even full-body motion to provide input for endless possibilities of interaction techniques (ITs).

In Human-Computer Interaction (HCI) research much effort has dealt with how to utilize such input for 3D interaction in VEs. One of many goals in this research is to allow for natural interaction, using human motor skills and real-world experiences while letting users focus on their tasks rather than operating the system (Froehlich & Bowman, 2009). 3D interfaces for VEs share the same principles found in many current HCI areas, such as in intelligent user interfaces, ambient intelligence, ubiquitous computing or natural-user interfaces. The shared paradigm for these interfaces is to augment the human intellect, to support thinking - rather than taking away cognitive resources for handling the system (Shneiderman, 2005). The development of such interfaces therefore draws on knowledge from many disciplines, such as psychology, cognition and perception, building on top of users pre-existing knowledge of the world. Often themes of reality are employed to guide the design, e.g. the use of naïve physics, incorporating the users' body, the surrounding environment and other people (Jacob et al., 2008). Especially in the design of VEs and 3D interfaces the usage of such guidelines and principles together with a fundamental understanding of the users cognitive capacities, is of great importance. The inherit complexity of VEs often take much of the users attention - mostly in the visual domain. Designing 3D UIs on an established paradigm as e.g. direct manipulation (Shneiderman, 1983) found in most GUI / WIMP systems may thus result in misusing cognitive resources on handling the system rather than supporting the user in focusing on the tasks at hand. The classical motto of "What you see is what you get" (WYSIWYG) needs rethinking and the current trends in 3D UI design focus on the motto "What you do is what happens" (Kulik, 2009). Thus the focus shifts from visual representation to embodied interaction - embracing direct motor skills in the design of ITs.

Research concerning interaction for VEs has since the late 1990's focused on defining formal frameworks and guidelines for the design of ITs. Bowman identified in his work (Bowman, 1998) three task categories in which most VE interactions fall: *Navigation*, *selection* and *manipulation*. In the last decade much effort in the UI community has dealt with further developing, identifying, describing and categorizing fundamental principles for designing these ITs. Our intention with the efforts and work in this master thesis is to further support the 3D UI research, by focusing on the controller free interaction possibilities in VEs. Considering most recent developments in affordable sensing hardware like depth cameras e.g. (Kinect, 2010) or (PrimeSensor, 2011) the trend of body controlled interfaces has been introduced to enter the living room. The applications currently supporting such hardware mainly focus on entertainment e.g. (Kinect Games, 2010). It is the intention of our work to utilize such affordable hardware and investigate the possibilities in the broader context of 3D user interfaces and Virtual Environments.

Issues regarding the deployment of required hardware, which in many cases is costly and not easy transportable and additionally may require maintenance staff, may be one of the reasons hindering the broad usage of VEs (Drettakis et al., 2007). A development which we think will change dramatically in the near future - partly through the ongoing availability of affordable hardware, but more important through the development of alternative interaction possibilities which focus on users' needs and which are carefully designed to help in thinking.

Though body controlled interactions have gained large popularity in current entertainment applications, the use of such interactions in tools and applications is almost exclusively limited to research. In this thesis we will focus on identifying possible reasons and describe solutions for the use of body controlled interactions in VEs. The benefits of VEs are manifold and some reasons are among others the enhanced sense of presence (Slater & Steed, 2000) and enhanced spatial understanding (Schuchardt & Bowman, 2007) (Phillips et al., 2010). Using body controlled interaction may even benefit in usability and learning (Antle et al., 2009).

The motivation for the work presented throughout this thesis was inspired by the release of the Microsoft Kinect, which in the last months inspired a variety of developers around the globe to do alternative projects in all kind of directions. With this hardware the motivation is to investigate its possibilities as interaction device for VEs; an interaction device which does not require the user to wear heavy equipment wired with cables and which is affordable providing possible usage in future outside the lab applications.

In the following pre-analysis, we will revisit the literature to identify possible shortcomings and to establish an understanding of the state of the art in 3D interaction and VEs, starting with an investigation of the principles behind 3D interfaces, going to a review of ITs used in VEs. As a guiding problem for this pre-analysis the following initial problem statement is used:

Which interaction techniques are commonly used in Virtual Environments and what are the principles behind them?

PART I - Pre-analysis

The pre-analysis will start with a broad perspective on interaction design, investigating the subjects of principle-based design, the affordances that are needed to have a good interaction design and perceptional and human cognition consideration in regards to VEs.

In order to finalize this thesis focus, research in the area of interaction techniques will be presented and analyzed, the structure of this section has been inspired by Bowman & colleagues task taxonomy of interaction techniques (Bowman et al., 2005).

Since the focus of the thesis is towards interaction techniques that utilize Microsoft's Kinect depth camera together with the NITE tracking framework, the chosen input and output sources will be described and analyzed in regards to the design consequences.

The pre-analysis chapter will have a closing discussion concerning the presented research and derive to this thesis focused problem statement of an investigation into interaction techniques in VEs.

2 Interaction principles

The description and definition of principles with the intention to guide and inform the design of human-computer interactions is of great importance for successful design. The amount of literature on formal design guidelines is vast, whereas only few regard the higher principles governing these guidelines. Principles can be seen as the fundamental pillars holding the more specific guidelines. Dennis Wixon (Microsoft Researcher) commented in his talk on principles and guidelines at the UX Week 2008 as the following:

> *"The principles are what drive the design. The guidelines are simply specific derivations of the principles for an individual context. Principals are what are important. Principals and data drive successful design."*
> *(Wixon, 2008)*

The principles can be used to guide every phase of the design process; they can help generating ideas and concepts. These principles can be seen as the taxonomy of the core ideas driving the design.

In this section we will have a look on some of the principles that can be applied to VEs: *Reality-based interaction* (Jacob et al., 2008) and *Imagination-based interaction* (Kulik, 2009).

2.1 Reality-based interaction

With their RBI framework Robert Jacob and colleagues (Jacob et al., 2008) cover many emerging interaction styles found in present post-WIMP interfaces. The foundation of this framework is that many of these interaction styles draw their strengths by building on users pre-existing knowledge of the real-world. Four RBI themes are used as the basis for their framework: *Naïve physics, Body awareness & skills, Environment awareness & skills and Social awareness & skills*.

Basing interactions on *naïve physics* means to take advantage of the users' common sense knowledge about the physical world. This includes concepts like gravity, friction, velocity, the persistence of objects and their relative scale (Jacob et al., 2008). In relation to VEs this concept can be directly transferred by using novel physics engines, which simulate such concepts.

Body awareness and skills refers to the familiarity and understanding that people have of their own bodies, independent of the environment (Jacob et al., 2008). This concept draws it strengths from proprioception; a user is aware of the position of limbs and the range of motion. Interactions can be built on this strength including the users' body in the interactions, using two-handed or even whole-body interaction. This concept is especially important in VEs and many of the interaction styles found for navigating, selecting and manipulating are employing it.

Environment Awareness and skills builds on the concept that humans are aware of their physical presence in a real world environment, surrounded by objects and landscape (Jacob et al., 2008). The clues provided by the surrounding environment facilitate a human's sense of orientation and spatial understanding. Such clues include for example depth clues like shadow, lighting, fog, atmospheric color and horizon. In VEs a common usage is e.g. reference object or landmarks to provide users with an understanding of position and distance. Also modern graphics engines are able to provide for real-time high detailed shadows and lighting to provide consistent clues for spatial awareness. For VE interactions the user's location and orientation in the environment can be further utilized to e.g. display information corresponding to this position. This concept also relates to the spatial awareness of objects, which in turn can be altered or manipulated by users, thus including skills such as picking up, positioning and arranging objects, which also draw concepts from the body awareness and naïve physics.

Social awareness and skills relates to social interaction skills and a users general awareness of the presence of others. This concept includes verbal and non-verbal communication, the ability to exchange physical objects, and the ability of collaboration (Jacob et al., 2008). Such interaction styles can be supported by direct co-located communication or mediated communication. In VEs there is possibility for both, though the use of technology has a big influence as e.g. HMDs mostly not allow for direct visual communication or contact. Often the use of either video images or even avatars is used in such environments to support communication, whereas the latter can support the communication by making the users actions visible through e.g. whole-body interaction.

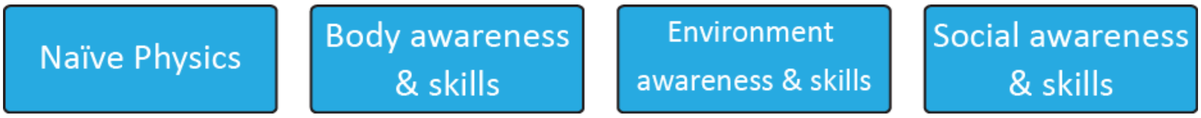


Figure 1 - Elements of Reality-based Interaction framework

In their work Jacob et al. also describe some of the tradeoffs that reality based interaction adheres. The mimicking of reality is not always desired and the RBI principles should be traded in return for other desired qualities, such as: *Expressive power* (users can perform a variety of tasks within a given application domain), *Efficiency* (users can perform tasks rapidly), *Versatility* (users can perform many tasks from different application domains), *ergonomics* (users can perform a task without physical injury or fatigue), *accessibility* (users with a variety of abilities can perform a task), *practicality* (the system is practical to develop and produce) (Jacob et al., 2008). These considerations will be included in the next section, which describes some of the concepts of Imagination-based interaction.

2.2 Imagination-based interaction

Many researchers describe that building VEs that try to approach the richness of 3D reality may often fail to address usability and ergonomics (Shneiderman, 2003), (Jacob et al., 2008), (Kulik, 2009). Though the most realistic VE interactions may be easy to comprehend, since they build on users' common knowledge, they may be unpractical in many senses. Jacob and colleagues described some of the desired qualities which should be traded for RBI (as described above), but they state that these qualities only should be traded explicitly and only in return for these qualities and they do not suggest a structured methodology for using these tradeoffs (Jacob et al., 2008). Alexander Kulik on the other hand suggests to explicitly exploit the use of what he coined *imagination-based interaction* (IBI) (Kulik, 2009). Most of the concepts and metaphors in WIMP based systems emerged from office work and most of them are inadequate for 3D applications. Alexander Kulik suggest to include other sources for comprehensible metaphors to take inspiration from popular science-fiction stories, common knowledge about technological achievements, and the arbitrary combination of aspects from previously learned patterns.

> *"Unrealistic actions aren't necessarily impossible. As long as they're computable and users can imagine or vaguely understand the resulting functionality, they might be useful." (Kulik, 2009)*

In his work he also suggests themes complementary to RBI, which are: *suspension of naïve physics, scaling of geometry and motion, automation, magic spells,* and *mode changes.*

The *suspension of naïve physics* refers to the qualities possible in VEs extending beyond the laws of physic. This can include many interactions, among others flying through air, walking through walls or placing objects in mid-air and many other "super-powers". Constrains can be defined freely and how they best support the task, instead of using the constrained metaphor of physical reality.

Geometric scaling refers to the possibility of scaling computer graphics to provide more detail or improve accuracy. In multi-touch interaction this concept has become broadly accepted, whereas in 3D UIs there is still no scaling concept with broad acceptance. In difference to 2D or monoscopic 3D UIs a full stereoscopic 3D display can offer the illusion of real scaling, whereas the others may only be perceived as a form of zooming (Kulik, 2009).

Motion scaling is related to scaling the motion input in order to tackle the trade-off between accuracy and rapidity. This concept can take advantage of influencing the motion of pointers and other motion operators and can greatly improve usability. The tradeoff is that the real motion of a person cannot be influenced and therefore it may result in confusion. This concept is widely used in WIMP systems where the mouse pointer has increased acceleration under rapid motions (Control:Display ratio (C:D)).

Automation generally refers to the computers capability to automate processes. This can take various forms of semi- to full automated interactions. In VEs examples can be found in navigation, where a distance is travelled automatic by e.g. selecting a target destination on an overview map or by selecting landmarks.

Magic Spells refers to computationally dynamically assigning meaning and automated processing to any involved object, which can be adapted to a variety of tasks and interaction styles (Kulik, 2009). In a VE this could for example involve selecting an object and assigning a color, without physically painting the object. Any arbitrary process could be applied and many interesting metaphors may be found in current computer games, which often utilize the concept of magic.

Mode changes refer to the properties of objects and tools in computer applications which might frequently change. This is especially important for avoiding confusion and a clear representation and distinguishable interface states are core properties. Emphasizing the transitions of system states can help support the user's cognitive processes (Kulik, 2009).

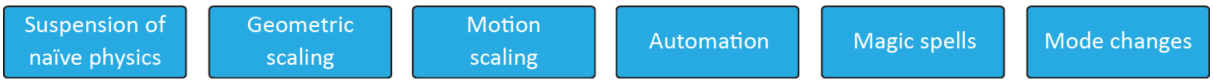


Figure 2 - Elements of the Imagination-based Interaction

2.3 Discussion

The principles of RBI and IBI have been chosen as a starting point for finding out which principles and metaphors could describe interactions for VEs. It has to be noted that there is a large amount of literature about design guidelines and methodology, and we could include many other principles which has been proposed in the field of HCI. One thing we noted is that many of these principles or guidelines are either very specific for a given area, or they are to general for guiding 3D and VE interface design. Also we did not find any principles particularly suited for body controlled interaction, except the described ones. Since this form of interaction is fundamentally different from the form of human-computer interaction we are used to (mouse and keyboard), we are eager to find out if there could be a set of principles suited for this form of interaction. We recognize the RBI and IBI principles could be used for guiding VE interaction design, yet we miss a set of principles which unify the concepts of reality (RBI) and imagination (IBI). Also the proposed principles are theoretically and the authors did not prove their applicability, by e.g. basing a VE design on them. Such theoretical design principles are most likely derived by thinking about them and thus there exists a gap between theory and practice.

3 Affordances, constrains, good mapping and feedback

While the principles described in the previous section may apply to a large range of present emerging interfaces there are a range of principles, which can be applied to almost any human-computer interface as well as many everyday artifacts. We will in the following describe some of these principles that are of importance for VEs. We will also have a look on their origins from psychology, with a focus on the concepts of embodiment and cognition from an ecological point of view.

Donald Norman described many of the well-established concepts and principles used across most HCI disciplines in his work (Norman, 1990). The principles described by Norman are based on a discussion of everyday artifacts and a big part of these principles have their origin in the psychological concepts of James Gibson's ecological approach to visual perception (Gibson, 1979).

Norman identifies four principles which are characteristics for usable artifacts: *affordances, constrains, good mapping* and *feedback*. He also includes some cognitive concepts in his work: *the gulfs of execution and evaluation*.

> *Affordances "…refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used." (Norman, 1990)*

Affordances provide strong clues to the operations of things, for instance a chair affords sitting, a knob affords turning, slots are for inserting things, etc. (Norman, 1990). *Constrains* are the limitations on the use of an object, with the intention to guide users into proper action. For example the holes in a pair of scissors limit the amount of possible fingers to constrain and guide the action (Norman, 1990). In 3D applications constrains are often used to limit the possible amount of manipulations, for instance translating objects in space is often constrained by one of the three major axis. *Mapping* refers to the relationship between two things, for example between the controls and their movement and the result in the world. This also refers to the conceptual model or metaphor on which the relation between action and outcome is based (Norman, 1990). The term *natural mapping* refers to taking advantage of physical analogies and cultural standards, which should lead to immediate understanding. *Feedback* is about sending information back to the user after performing an action, about what result has been accomplished (Norman, 1990). This concepts also relates to a psychology of causality (cause and effect), where something that happens right after an action appears to be caused by that action. Proper feedback is essential in all applications to not confuse the user and to actually communicate the success of actions.

The gulfs of execution and evaluation refer to the gaps (or gulfs) that separate mental states from physical ones. Each of these reflects one aspect of the distance between mental representation of the person and the physical components and states of the environment (Norman, 1990). The gulf of execution refers to the difference between the intentions and the allowable actions. The gulf of evaluation reflects on the amount of effort that a user must exert to interpret the physical state of the system and to determine how well the expectations and intentions have been met (Norman, 1990). This gap is small when the system provides meaningful feedback, which should be easy to interpret and matches the users' mental model of the system. In relation to the previous mentioned RBI principles this gap is small as most of these principles are based on the users preexisting knowledge from everyday life.

3.1 Ecological perception and Embodiment

Gibson's ecological approach to perception is in direct contrast to the traditional cognitive psychologist's viewpoint on perception. In the traditional view there is direct access only to sensations which integrated with memory provide higher level interpretations of actions or objects. In Gibson's view there is a directly perceivable complex group of features, which he denoted *invariants* (Gibson, 1979). In his viewpoint these invariants are directly perceivable and are not mediated by memory or inference. Though this approach is controversial his descriptions of embodiment is interesting to note for the field of body controlled interaction and VEs.

Affordances are the fundamental concept of the ecological approach to perception (Gibson, 1979). Here the definition is similar to Normans:

> *"Affordances are opportunities for action that objects, events or places in the environment provide for an animal." (Hirose, 2002)*

With this definition he also highlights that affordances are properties of the environment but not of the animal (e.g. human) itself. The means for acting that an animal can use to realize a specific affordance are called *effectivities*. These effectivities may change as the state of the animal changes, because they are properties of the animal (Hirose, 2002). *Tools* can extend or enhance these effectivities, for example a telescope amplifies the visual ability, or a hammer amplifies striking ability. Separated from the body these tools are functional extensions of the environment. In use, the tools are treated as functional extensions of the user; it extends the users effectivities to realize affordances of the environment (Hirose, 2002). By this tools may extend the users *body*; they shift the boundary between the body and the environment. For instance a blind persons stick extends the scope of touch, in use it is regarded as being part of the body, it extends the body. In sum the ecological approach to embodiment regards that the boundary of the body is changeable and can be extended by tools (Gibson, 1979) (Hirose, 2002). In relation to interactions in VEs this concept of extending the body by tools seems interesting; it may reveal different forms of ITs when used as a perspective for investigation. The different ITs used in VEs also draw a large part of their inspiration from this ecological view, where for example users can extend their arms to select and manipulate distant objects. Furthermore a lot of potential is offered by this ecological view, considering VEs where virtual human body avatars can be extended, parts replaced or even bodies completely replaced by any possible mean, giving possibilities for a broader application of ITs.

4 Perception and human cognition

Working with multimodal systems like VE's, further understanding of the processes human undergo while interacting in such systems is needed. Nadine Sarter (Sarter, 2006) argues that most research within human-computer interaction has treated system input and output as different domains. Similar division into two directions is found in the literature concerning the human processes: perceptual and cognitive psychology. Perceptual psychology (the input) studies how human perceive their surroundings through their senses. Cognitive psychology (reasoning and output) focuses on the mental processes, how people learn, reason etc. According to (Buxton et al., 2011) these two processes are interlinked, and we cannot discuss one process without the other.

Within the research area of human perception, there is a common understanding that perception is a multimodal process (Pai, 2005). According to Pai (2005) the most known multisensory phenomenon is the ventriloquism effect, the spatial location of the sound is where the location of correlating visual stimuli is situated. The McGurk effect (McGurk & MacDonald, 1976) is a known illustration of speech being a multimodal process. In haptics, surface and spatial information gathering is seen as a multimodal process combining touch, audio and vision (Klatzky & Lederman, 2003).

Most Virtual Environment applications have focused on visual perception, the reason is according to Bowman and colleagues that most immersive VEs are highly visual (Bowman et al., 2005). According to them an important aspect in VEs is the perception of depth in the environment, where different visual cues can help the subject to determine the depth of the scene. These cues are divided into monocular or binocular, static or dynamic.

When dealing with interaction and interfaces, words like natural interaction and intuitive designs are often mentioned. Oxford dictionary defines intuitive as: "*Of knowledge or mental perception: That consists in immediate apprehension, without the intervention of any reasoning process.*" meaning the cognitive process that is needed to be capable of performing a task without thinking about it.

Many cognitive theorists have tried to map the cognitive process, where one of the best known is the "*Modal Model of Memory*" suggested by Atkinson and Shiffrin in 1968. According to them there are three types of memory storage: *sensory, short-term memory and long-term memory.* The model suggests that all memory pass through short-term to a long-term store after a period of time. Miller's law suggests that the short time memory can store seven plus minus two units at the time (Miller, 1955). Long term memory on the other hand can store a vast amount of information in a long period of time (Ericsson & Kintsch, 1995).

Although the "*Modal Model of Memory*" theory has been criticized by later theorists for being a simplification of the mental processes, e.g. (Eysenck & Keane, 2005) and (Baddeley, 1997), it has been the basis for many cognitive theories.

According to Eysenck and Keane the key component of the working memory is the central executive and it has limited capacity, the other three are subsystems that are relative independent of each other. The assumption of working memory is:

- *If two tasks use the same component, they cannot be performed successfully together.*
- *If two tasks use different components, it should be possible to perform them as well together as separately*

(Eysenck & Keane, 2005)

Knowing these limitations of the working memory will aid to a better utilization of multimodal sensory elements in an interactive VE design.

5 Interaction techniques

The following chapter will introduce novel research within the area of ITs in VEs. The research is presented in the order using Bowman & Colleagues definition of interaction areas (Bowman et al., 2005). This definition consists of three main areas, which are defined as:

- *Navigation:* travel or view point motion control
- *Selection*: involves the specification of one or more virtual objects by the user for some purpose
- *Manipulation:* setting the position and or orientation of a virtual object

(Bowman, 1999).

Another element of VEs are User Interfaces (UI), such as menus, toolboxes, palettes etc. These are defined by Bowman as *System Control techniques*. Chen & Bowman refers to theses generic task levels as universal, meaning that they can be applied to any application without taking the context into account (Chen & Bowman, 2009).

Having these definitions in mind the next step is to examine what has previously been done in the area of ITs for VEs. The main focus will be to examine the ITs, how they were supported by a medium, and finally examine to which degree they are usable to the current investigation.

To ease the discussion of past research, an outline of research done in each of the task areas navigation, selection, manipulation and system operations, will be illustrated in tables. The information about each project we see relevant at this point is the input/output interface and the ITs used in order to perform each of the tasks.

5.1 Navigation

The articles mentioned in the following section and in Table 1 are a cross section of the most reoccurring ITs concerning navigating in VEs, which according to Suma et al. are: "*real walking, walking-in-place, gaze-directed, pointing-directed, torso-directed, and traditional desktop controls*" (Suma et al., 2010). Some projects use a single technique, while others choose to use a combination of different techniques.

The problem with navigation in VEs is the fact that one is confined to a real world physical space while travelling a possible endless virtual space. There have been many attempts to investigate this problem. Researchers have been trying to find out if one can map the virtual space to the limited physical space for navigation (Peck et al., 2010), others have constrained the physical boundaries, may it be the space or movement of the subject (Valkov et al., 2010), (Terziman et al., 2010), (Wendt et al., 2010), while others to a certain degree have implemented real life navigation from a physical space to a virtual one (Bruder et al., 2009), (Suma et al., 2010).

The question of which technique is best suited has been asked and to a certain degree answered. According to Suma et al. there is no significant difference in travel performance between real walking, gaze directed or torso directed navigation. What they were focusing on was the degree of information the user gained while navigating in VEs, but on the other hand real walking outperformed pointing direction techniques. This opens the discussion about to what degree the pointing direction technique is suited for navigation. Suma & colleagues and others have proven that one does not need real life navigation in order to gain information about the VE. Bowman pointed out that the user needs to have a fluent stream of information, in order not to be disoriented in the Virtual Environment (Bowman et al., 1997). He describes that actions like teleportation from one place to another will be more confusing for the user than having him travelling from one location to another via traditional travelling means, meaning linear travel like walking.

Input/output interface				Interaction technique	
References	**Output: Display**	**Input: Tracking**	**Tracking Hardware**	**Name of technique**	**Available Motions**
(Terziman et al., 2010)	HMD/ Screen	Head	Web camera	Shake-Your-Head	Advance speed, Jump, Crawl, Turn
(Wendt et al., 2010)	HMD	Shins	Eight-camera PhaseSpace Impulse optical motion Capture system	Walking-In-Place	Forward Walking (various speeds), Turn
(Peck et al., 2010)	HMD	Head	3rdTech HiBall 3000 optical tracking	Improved Redirection with Distractors	Forward/Backward/Left/Right motion (various speed), Rotated view.
(Valkov et al., 2010)	Screen	Hands, Feet	Balance Board[1], Transparent FTIR-based multi-touch surface	2D Interface	Forward Walking
(Suma et al., 2010)	HMD	Torso, Head, Hand	3rdTech Hiball 3100 optical tracking, Nintendo Wii Nunchuk[2]	Torso-/Point/ Gaze directed Navigation	Forward/Backward Walking, Rotation
(Bruder et al., 2009)	HMD	Head, Hands	InterSense InertiaCube2[3], Nintendo Wii remote controller	Real walking IVE	Forward walking

Table 1 - An overview of research done in the area of navigating in the VE. The figure is inspired by (Terziman et al., 2010).

There is a fair amount of research done in the area of navigating in VEs, and the outcome of this research seem to mount out in the fact that in order to design a navigation method that will give the user most information about the virtual environment that he/she is travelling in, the best approach would be to give the user a consistent fluent stream of information as he/she would

[1] http://www.nintendo.com/wii/console/accessories/balanceboard
[2] http://us.wii.com/hardware/
[3] http://www.intersense.com/pages/18/55/

have when travelling in a physical environment. The research shows further that one is not necessary bound to the physical real life exploration options, when travelling in VEs.

When claiming that the question of best suited navigation techniques is to a certain degree answered, is due to the fact that it depends on what one is aiming for: is it an attempt to find the most user friendly way of travelling? Or is it to find a way of navigating which affects the presence or engagement? These choices open up for another discussion which is not within the scope of this project.

5.2 Selection

	Input/output interface			Interaction technique	
References	**Output: Display**	**Input: Tracking**	**Tracking hardware**	**Name of technique**	**Available Motions**
(Dominjon et al., 2005)	Screen	Hands	N/A	"Bubble" technique	6 Degree Of Freedom
(Chen et al., 2004)	HMD/WWD	Hands	InterSence[1] IS900 hand tracker	HOMER	6 Degree Of Freedom
(Choumane et al., 2010)	Screen	Hands	DTrack device[2]	Trajectory and kinematic Patterns (cursor)	Circular movement
(Martens et al., 2004)	Personal Space Station	Head, Hands	Infrared stereo cameras	Pen selection, on a physical object	Natural hand movements
(Liang & Green, 1994)	Screen	Hands	Hand-held Isotrak sensor	Spotlight selection	Rotation around the X-axis
(Wyss et al., 2006)	Barco Baron display-type workbench with active stereo projection	Hands	N/A	iSith (Ray casting)	6 Degree Of Freedom
(Maier et al., 2010)	Screen	Hands	Marker-based optical tracking	Tracked Physical Object	Natural hand movements, Shake
(Pierce et al., 1997)	HMD	Hands	N/A	Head Crusher technique	Natural hand movements
(Poupyrev et al., 1996)	VR	Hands	Polhemus Fastrak 6 degree-of-freedom sensor	Go-Go technique	6 degree-of-freedom

Table 2 - An overview of research done in the area of close and out of range selection techniques in the VE.

Mine defines two main categories of selection techniques as: *local* and *at-a-distance* (Mine, 1995). Local, as the name suggests, concerns the process of selecting objects that are within the subject's reach. All of the mentioned articles in Table 2 have an element of local selection involved

[1] http://www.intersense.com/pages/20/14

[2] http://www.ar-tracking.com/Tracking-software-DTrack2.19+B6Jkw9dWtxdm5rZng_.0.html

in their research, but (Martens et al., 2004), (Maier et al., 2010) focus both on selecting objects within reach and on having a physical object as aid to the selection. Martens et al. claim that having physical objects as aid will support the natural interaction with the 3D world. These techniques have the limitation of the physical object, although one can argue that this object can serve as a container of virtual possibilities, such as a 2D widget interface in a VE.

According to Wingrave, Bowman, & Ramakrishnan (2002) there are in the area of selection techniques three main metaphoric naming approaches: *ray casting* (Dominjon et al., 2005) (Wyss et al., 2006) (Liang & Green, 1994) (Chen et al., 2004)*, occlusion* (Pierce et al., 1997) and *arm extension* (Poupyrev et al., 1996)**.** Linking these metaphors to the category of selection *at-a-distance*, it becomes a question of how well the user understands the metaphor of selecting objects that are far away and how well the feedback is designed. One can argue that having a ray as a pointing support to the process of selection can be helpful, while others will argue having a “bulls-eye” occlusion selection technique (Wingrave et al., 2002), helps the user to set the object of interest in focus. With arm extension selection techniques the user can to a higher extent relate to the action of selection, since it is to some point a more human like action.

Most of the research done in the area of object selection has focused on the visual feedback the user gets when performing an action and has focused on the hands as the primary source of selection. Another observation made is that most feedback used in selection techniques is visual feedback. This seems to be a great limitation when having the possibility of a multi-modal VE. It seems that sound has not been prioritized when dealing with selection techniques which could be subject for future research.

5.3 Manipulation

Bowman et al. (2005) defines manipulation as a set of tasks, where to a degree each manipulation contains a combination of the three basic tasks *Selection, Positioning and Rotation* (Bowman et al., 2005). Further Bowman et al. (2005) separates the manipulation ITs being either *isomorphic* or *non-isomorphic* view. Isomorphic view suggests that the physical world and virtual world’s actions are geometrically mapped one-to-one. A non-isometric view suggests a more open frame of mapping physical actions to virtual actions. Further Bowman classifies the techniques as series of metaphors. These classifications will assist in organizing the techniques presented in Table 3.

We can derive from Bowman’s definition of manipulation that selection techniques to a certain degree also can be considered as manipulation techniques. Some of the in the selection techniques section mentioned researchers operate within the isomorphic point of view, while (Poupyrev et al., 1996), (Chen et al., 2004) and (Pierce et al., 1997) have a more non-isomorphic approach towards manipulation. The selection part of the manipulation often has a isomorphic approach, other manipulation function often use a non-isomorphic method, as e.g. a *plane* based manipulation (Hoang et al., 2009), *Voodoo doll technique* (Pierce et al., 1999), *2D interface technique* (Igarashi et al., 2007), or *WIM* (Stoakley et al., 1995).

Input/output interface				Interaction technique	
References	**Output: Display**	**Input: Tracking**	**Tracking hardware**	**Name of technique**	**Available Motions**
(Stoakley et al., 1995)	HMD	Hands	Polhemus tracker sensor[1]	World In Miniature	Picking, pointing, rotation. 6 DOF
(Pierce et al., 1999)	HMD	Hands	Fakespace PinchGloves[2]	Voodoo Dolls	Pinch, 6 DOF
(Liu et al., 2010)	Screen	Hands	Polhemus FASTRAK[3], 6DOF, Ultrasound Logitech Head tracking	Pointing	6 DOF
(Hoang et al., 2009)	HMD	Hands	Pinch gloves	Image plane	2DOF
(Gutiérrez et al., 2005)	Screen	Hands, Head	Optical tracking, PDA interface	Pointing + semantic description	Pointing, facial gestures
(Igarashi et al., 2007)	Screen	Hands	Display-integrated tablet[4], Electric white-board, Mouse	2D interface	2DOF.

Table 3 - An outline of manipulation techniques.

There is a vast amount of literature concerning manipulation techniques that utilize hand tracking in some manner, while whole body movements are to a lesser degree employed in the design of manipulation techniques. This focus of interest has the effect that most metaphoric representations of manipulations are limited to tasks that can be subjected to hand labor.

5.4 System Control

Bowman et al. (2005) composed a taxonomy that sums up the different areas of system control, *Graphical menu* (Lindeman et al., 1999) (Pierce et al., 1999), *Voice command* (Lee & Billinghurst, 2008), *Gesture command* (White et al., 2009) and *Tools* (Forsberg et al., 2000).

Pierce et al. (1999) identified drawbacks of using 2D or 3D widgets in a VE, 2D consuming screen space, and a constant shift between 2D desktop and 3D world. 3D widgets have the issue of access, how to gain access to them, and the danger of cluttering up the 3D world. Voice commands have been examined, by e.g. (McTear, 2001). Lee & Billinghurst (2008) concluded in their research, that in order to use speech recognition key words are needed.

Gesture commands are often limited to the input technology used in the setup. Bowman et al. (2005) define three types of gesture input techniques; *glove-based recognition, camera-based recognition, surface-based recognition*.

[1] http://www.polhemus.com/
[2] http://www.inition.co.uk/inition/dispatcher.php?URL_=product_glove_fakespace_pinch&SubCatID_=26&model=products&action=get&tab=summary
[3] http://www.i-glassesstore.com/logtractracs.html
[4] http://www.mutoh.com/support/mature/mat_videotab.html

White & colleagues (2009) use a camera-based gesture recognition technique, which is based on a shake technique. They argue that this has an advantage over simple button press, by the fact that any object can be tracked and function as a menu controller.

Input/output interface				Interaction technique	
References	**Output: Display**	**Input: Tracking**	**Tracking hardware**	**Name of technique**	**Available Motions**
(Gerber & Bechmann, 2005)	Screen	Hand	Intersense IS900 Tracker	Spin Menu	6 DOF
(White et al., 2009)	See through Head-worn display	Hand	6DOF optical marker tracking using AR-Tag[1]	Shake menu	Shake
(Pierce et al., 1999)	Desktop Virtual Reality	Hand	Touch pad	3D Widget	2 DOF
(Lindeman et al., 1999)	HMD	Hand	Logitech ultrasonic tracker, Ascension Flock-of-Birds magnetic tracker[2]	2D Widget	2DOF
(Lee & Billinghurst, 2008)	HMD, Screen	Hand, Voice	BumbleBee2 tracker[3] camera, Microphone	Hand, Voice gesture	Natural hand interactions, speech commands
(Forsberg et al., 2000)	4 wall CAVE	Hand	Microphone, Pinch Gloves	Gestural and voice	6 DOF, pinch

Table 4 - An overview of different techniques within the area of system control.

Tools in VEs are defined by Bowman & colleagues (2005) as either physical or virtual. They argue that the use of familiar real-world devices will increase the 3D interaction usability. Forsberg & colleagues (2000) used a virtual tool belt as aid for interacting in the VE; the belt contained different 3D metaphorical widgets.

Design of graphical menus seems to be largely inspired by earlier desktop application designs, based on the WIMP metaphor. This is not directly a negative direction to take, most of the methods are tested and verified to a full extend, but we argue that 3D environments must and can offer more than just existing 2D interaction methods.

[1] http://www.artag.net/
[2] http://www.ascension-tech.com/realtime/RTflockofBIRDS.php
[3] http://www.ptgrey.com/products/bumblebee2/bumblebee2_stereo_camera.asp

6 VE input and output methods

Previous research of ITs has to a large degree focused on obtaining user input data by devices, such as e.g. pinch gloves and tracking devices which often are very expensive and exclusively used in lab settings. Our focus will be on the recent trend of affordable depth sensing cameras, starting with the possibilities these products can offer. The following section will cover the area of different frameworks that are used in user tracking. Finally the output devices will be covered, here the focus will be on 3D output source, since the goal of this study is to create IT for 3D Virtual Environments.

6.1 Input

Since our motivation is to create a VE that allows users to perform actions without wires attached to the user, the following section will cover optical tracking devices allowing such interaction.

Optical tracking utilizes the electromagnetic spectrum of light, as depicted in Figure 3.

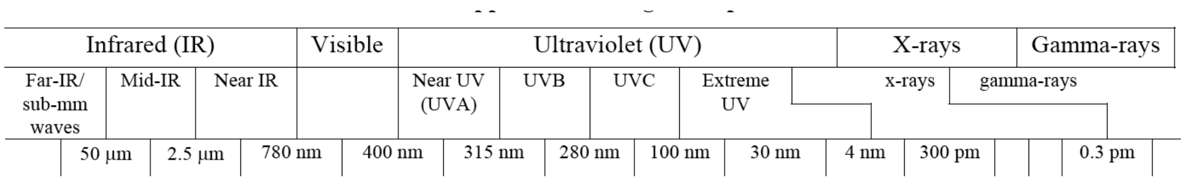


Figure 3 - The electromagnetic spectrum ranging from about 10^{12} Hz to more than 10^{22} Hz. Source of figure (Foxlin, 2002).

According to Foxlin (2002) common and inexpensive cameras based on CCD and CMOS driven sensors can be used to record visible and near Infrared (IR) light. Since most optical tracking systems up to date work in that area, due to complications toward reaching in the mid-IR and UV light (Foxlin, 2002), this may be considered as a relative cheep method of tracking by e.g. using multiple web-cameras.

There are different approaches towards optical tracking, these are beacon-tracking, optical TOF ranging and structured light. PlayStation Move uses the beacon-tracking, whereas the Kinect utilizes a structured light methodology (PrimeSense Ltd., 2010).

Beacon-tracking is where the subject wears an active or passive vision based tracker, and the system recognizes the position and orientation of the subject. There are two systems of beacon-tracking; outside-in and inside-out. Outside-in is better for positioning of the subject, while inside-out is better to determine the orientation of the subject. The main disadvantage of the optical system is occlusion of the tracked objects; that is when the line of sight is not clear between the sensor and the object that is being tracked.

The structured light approach, taken with i.a. the Kinect, has been to send out IR light in the scene and using a CMOS sensor to read the coded light back from the scene (PrimeSense Ltd., 2010). The result is a 3D representation of the scene, which then can be further processed by computer vision software.

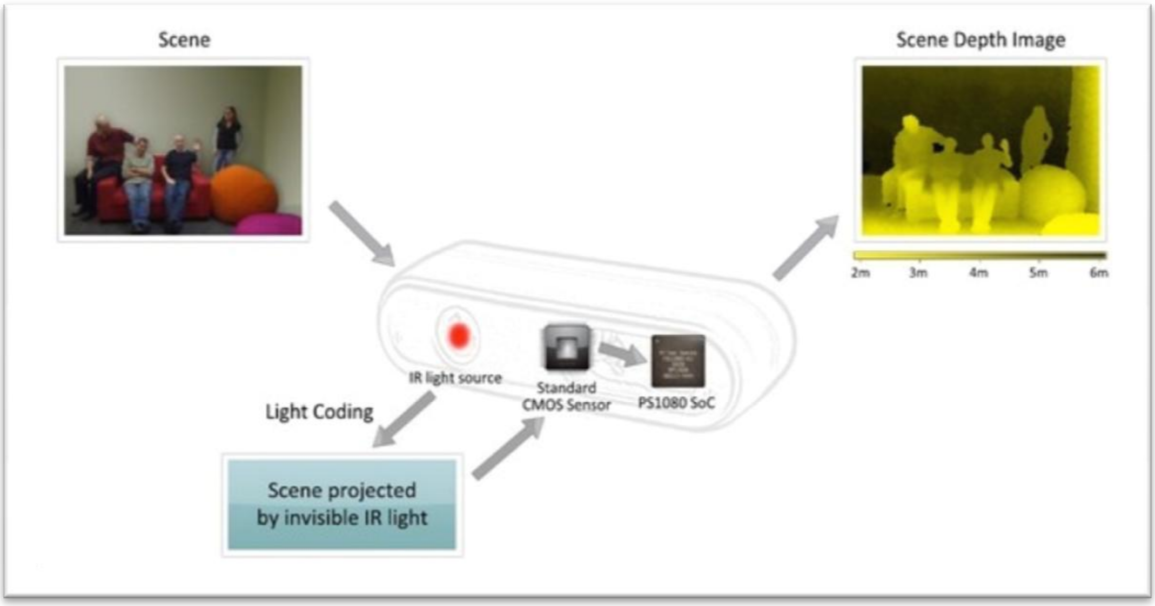


Figure 4 - The principle of how the Kinect is collecting 3D information about the scene, note that the picture is depicting a depth camera developed by PrimeSense. Source of picture: (PrimeSense Ltd., 2010).

The advantage of the structured light approach over the other optical tracking devices is that it does not require any form of physical addition on the subject (as e.g. a device in the hand or markers on the body).

For further details on tracking devices see (Bowman et al., 2005), (Welch & Foxlin, 2002) and (Foxlin, 2002).

6.2 Body part recognition

The choice of using the Kinect as an input source has the advantage of having access to the publicly available NITE tracking framework ((OpenNI, 2011) (PrimeSense, 2011)) which is capable of retrieving the depth information and from that recognizing body parts. Since the focus of our investigation is on the creation of ITs, the access to such a tracking framework was considered as an advantage that could not be overlooked.

According to Shottom & colleagues it is possible from a single input depth image to have a per-pixel body part distribution, which is color labeled in order to conduct weight estimation resulting in a proposition of each skeletons joint position, as seen in Figure 5. This information can be used in order to construct different interaction methods in a VE. See (Shotton et al., 2011) for more details concerning this tracking methodology.

This methodology will give the position of each major body part, but according to (OpenNI, 2010) individual finger information cannot be obtained; only the center of the palm. Hence the interaction techniques will be confined to body joints.

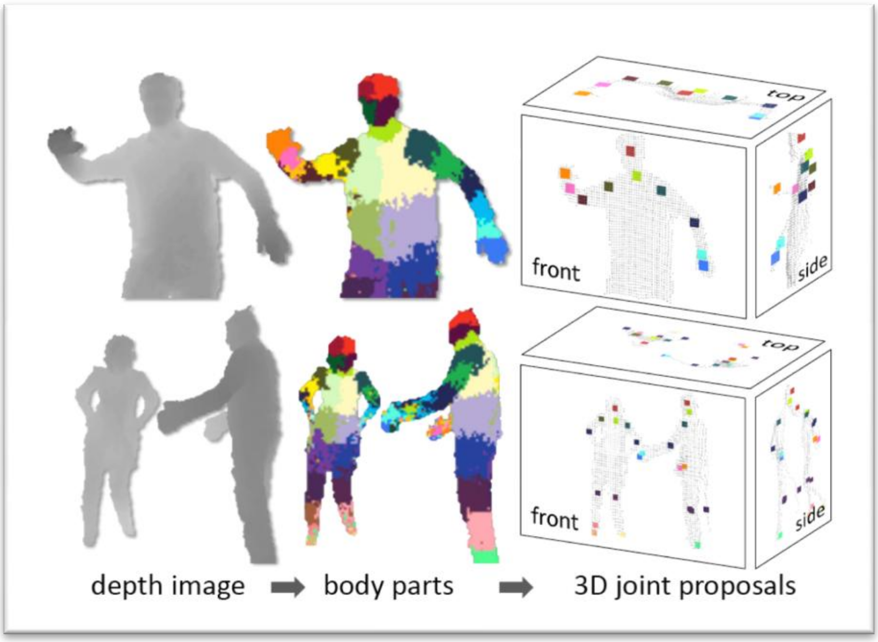


Figure 5 - Method of retrieving camera depth information and predicting 3d positions of body joints. Source of image: (Shotton et al., 2011)

6.3 Output

According to Foxlin (2002) in the field of 3D visual displays, there are two types of displays; Head-Mounted Displays (HMD) and Fixed-Surface Displays (FSD). Due to the motivation of having a VE that has no wires connected to the user, the choice of output was to use FSD.

Fixed-Surface Displays denote any stationary display station ranging from a desktop display to ImmersaDesk™, to a wall projection that can be up to a full Spatially Immersive Display (SID) e.g. CAVE™. These FSD can be equipped for stereoscopic 3D view, using shutter glasses, polarized glasses or auto-stereoscopic display techniques (Foxlin, 2002).

The choice of output display for a VE fell on using a wall projection with stereoscopic capabilities accompanied with shutter glasses. The reason was due to accessibility of materials.

7 Discussion

For the development of ITs *two* different approaches can be used as the bases for design, where most ITs use *metaphors or principles* to form the fundamental mental model of the technique, as for example the metaphors presented in this pre-analysis. The other approach is *systematic task decomposition*, where tasks are classified and divided into components, which serve as basic elements for constructing and combining new ITs, as used in the work of (Bowman et al., 2005).

Previous research in ITs has been focusing on the hands as a primary mean of *selecting* or *manipulating* objects in the virtual world, *system control* differs to a degree since there have also been implemented voice commands. In the area of *navigation*, researchers have implemented methods that involve more body parts, such as torso, feet and head.

In many research prototypes data gloves or 3D pointer devices are used for selection and manipulation tasks. Although many other examples can be found, there has only been little attention on using continuous input devices for all tasks. Most often a combination of devices is used for input, e.g. tracking devices together with data gloves or 3D pointer devices. In most cases prototypes use buttons, bend-sensors or other form of finger-controlled sensors to execute actions, such as confirming a selection. Only little attention has been paid on solely using tracking as input. We think that in most cases the combination of input devices was used to avoid dealing with the problems that come when relying on a single input device. We do recognize that using multiple input devices may give advantages for many tasks, but we believe that at least an equal usability can be achieved with a single input device, such as the Kinect. In this context it is interesting to note that one of the guidelines from the literature for choosing input devices is the following:

> *"It is often better to have a series of specialized devices that work well for a specific group of interaction techniques rather than one or two generic input devices for all of the techniques in 3D applications."*
> *(Bowman et al., 2005) - p. 128*

It is our intention to investigate the problems that come with using a single continuous input device and to find out if the Kinect together with the NITE framework qualifies as useable interaction device for VEs. One of the important problems to deal with in this case is how to *confirm or execute actions or selections* (coined "*confirmation of selection*" by (Bowman et al., 2005)), since the input is continuous and there is no distinct state indicating a confirmation.

8 Problem statement

8.1 Goals, Focus and delimitation

The intention in this thesis is to develop a theoretical model which we use as the bases for designing ITs. Also we want to present principles which unify the presented IBI and RBI concepts and are useful in the design of controller free interactions. We will limit this investigation on visual selection tasks as we see it as the core task for controlling 3D user interfaces. In this way a selection task precede almost all manipulation tasks and some navigation tasks, and is one of the fundamental interaction techniques reoccurring when designing interfaces. Driven by the motivation of developing a cable-free and low-cost VE interface, we will focus on body controlled selection interactions using optical tracking (Kinect) and stereoscopic screen based output for the implementation of ITs. Within this focus on selection we will exclude local selection, and focus on at-a-distance selection, since the local selection requires navigation.

With this work we wish to contribute with a framework and specific usability evaluated ITs which can be used by other developers of novel body controlled interfaces. From these ITs we also wish to generalize the experiences with developing and the results from evaluating them in form of a list of guidelines usable for body controlled interfaces.

The goals for this thesis can be summarized as the following:

- Development of a theoretical model
- Applying the model in the design of interaction techniques
- Evaluation of implemented interaction techniques
- Generalization in form of guidelines

8.2 Problem statement

Based on the investigation throughout the pre-analysis and the goals set for this thesis the following problem statement is formulated:

Which body controlled interaction techniques are best suited for single and multiple selection of screen content, using optical markerless motion capture (Kinect and NITE) and which design elements constitute the success of a technique?

This problem statement serves as the main focus of the ongoing investigation throughout the rest of this thesis. To clarify the meaning an explanation of the elements in the problem statement follows.

When writing "best suited" we refer to the usability of the interaction techniques. Since usability has multiple components, which are defined differently throughout the literature, the definition of Jacob Nielsen, who described usability based on five attributes, is used. These attributes are: learnability, efficiency, memorability, errors and satisfaction (Nielsen, 1993). Additionally the comfort attribute will be part of this definition, since comfort is important for body controlled interaction. A detailed description of these parts constituting to the usability is included in the next section, the methodology.

Another important factor to clarify at this point is that usability is mostly evaluated and understood in the context of a target group. In this way are the attributes weighted differently based on the intended user of a system. Since the problem targeted in this thesis deals with the fundamentals of selection techniques, and by this does not have a specific end product or target group, it was chosen to concentrate on novice users. This choice is based on the assumption that most people have no experience with body controlled interfaces and by this novice users may be the greatest part of the target group of future systems using the techniques developed in this project.

The last part of the problem statement: "which design elements constitute to the success of a technique" should be understood as the intended outcome of the evaluation, which are design guidelines intended to help designers to understand and develop future system using body controlled interaction techniques.

9 Methodology

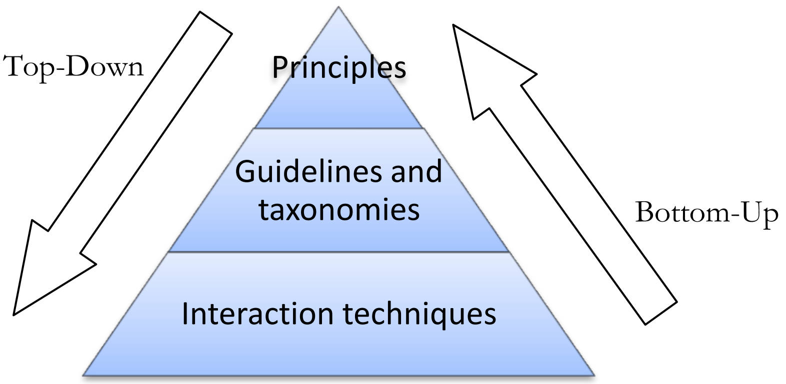


Figure 6 - Top-down vs bottom-up approach

9.1 Process

For the design and development of new ITs we will consider two principal different approaches, which we refer to as *top-down* or *bottom-up* approaches (Figure 6). In general the top-down approach starts with a broad perspective and from that perspective the more concrete design concepts are developed. In practice this could be the formulation of design-principles or metaphors from which ideas for ITs are developed. Often such approach is helpful for creating consistency in the design of the interface and the associated ITs. The disadvantage of such an approach is often the lack of formalism and structure (Bowman & Hodges, 1999), when compared to a bottom-up approach. The bottom-up approach tries to give structure and formalize in a systematic way all parts of a system. In this approach taxonomies are often used to create categories and to identify shortcomings in a design. Bowman and colleagues use such an approach throughout his work (Bowman, 1998) (Bowman & Hodges, 1999) (Bowman, 1999) etc., to formalize ITs for VEs. Using a similar approach building on the existing research and findings will be used as a starting point in the following analysis chapter. This will help us to identify various possibilities of ITs for the selection tasks. Since we also want to incorporate consistent metaphors in the design of selection techniques we will synthesize principles, based on the research presented in this pre-analysis. These principles will be used in the practical design of new ITs for selection tasks.

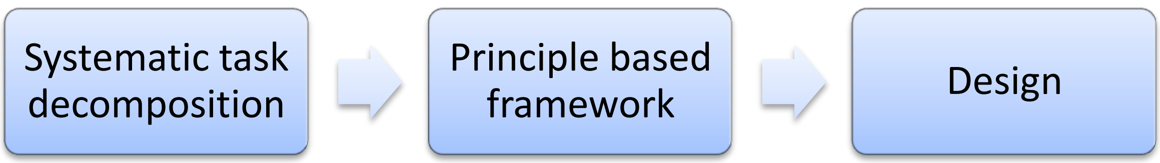


Figure 7 - Process structure analysis

Figure 7 shows an outline of the theoretical process for the analysis, which will be used as the bases for the design. The design process has the goal of designing ITs for selection, suited for various scenarios. Though the basis for this process is a theoretical developed framework, it is still a fundamentally creative process; ideas need to be developed, prototyped and tested. Our approach to this process will be based on Human-centered design (Maguire, 2001), depicted in Figure 8. The overall process is based on an *active involvement of users, appropriate allocation of function between user and system* and *iteration of design solutions* (Maguire, 2001). The active involvement of users will in general let us evaluate the usability of the system, where an early involvement can identify weaknesses early in the process and thus lead to avoiding errors. The appropriate allocation of function between users and system is especially important, as this is used to determine which aspects of the task should be handled by the user and which by the system (Maguire, 2001). The iteration as overall structure is essential for the process; it allows constant reevaluation and refinement of proposed design solutions. In general the design starts with developing a low fidelity prototype, using paper drawings and photos in order to allow an early identification of problems and solutions. Later stages of the design include high fidelity prototypes, with a proper implementation of a system focused on selection tasks in VEs.

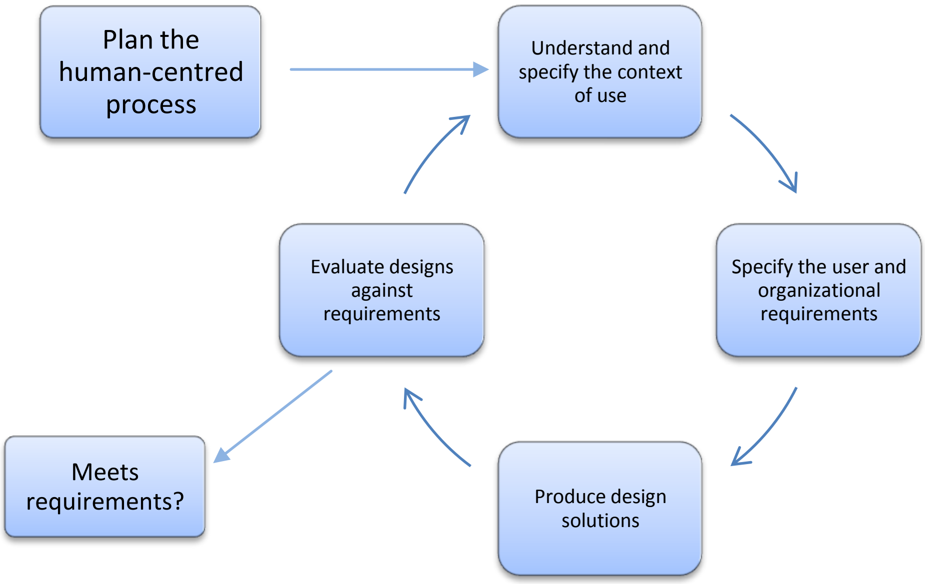


Figure 8 - Key human-centered design activities (from the ISO 13407 standard)

9.2 Evaluation criteria and metrics

With the goal of developing and verifying usable selection techniques, we need to define usability. Jakob Nielsen defined five usability attributes in his book "Usability Engineering" (Nielsen, 1993), which are:

- *Learnability*: The system should be easy to learn so that the user can rapidly start getting some work done with the system.
- *Efficiency*: The system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible.
- *Memorability*: The system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again.
- *Errors*: The system should have a low error rate, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them. Further, catastrophic errors must not occur.
- *Satisfaction*: The system should be pleasant to use, so that users are subjectively satisfied when using it; they like it.

(Nielsen, 1993) - p. 26

Other authors choose a more compact definition into three major categories: *effectiveness, efficiency,* and *satisfaction* (Frøkjær et al., 2000). Here the effectiveness refers to the accuracy and completeness with which a user reaches a certain goal. Metrics for effectiveness can include qualitative measures - meaning specific measures of the interaction between user and system - and error rates. Efficiency refers to the relation between the accuracy and completeness with which users reaches certain goals as well as the resources expended in achieving them. Metrics for efficiency include task completion times and learning time. Satisfaction refers to the comfort and subjective attitude of users towards the system. Metrics can include different rating scales of the user comfort or attitude, commonly evaluated through interviews or questionnaires. In VEs the satisfaction could also include measures of presence, for which a range of questionnaires have been developed.

Besides metrics for evaluating usability which are focused on the user, the system performance should also be evaluated. The system performance gives indications on how usable a system is, since a system achieving low performance may hinder the user interaction significantly. For system performance general metrics include, average frame rates, average latency - meaning input and output delay, variability in frame rate - identifying situations with low frame rates, network delay, and distortion - which in VEs for example can happen in graphical output using HMDs. All of these parameters can have an effect on task performance and usability.

9.3 Evaluation methods

With the iterative human-centered design used in this thesis practical work, a range of different methods for evaluating the different stages of prototypes will be used. Early prototypes are thus based on heuristic evaluations, whereas the later stages include formative and summative evaluations. In the following we will give a short introduction of these three evaluation forms, where more details can be read in the respective test chapter and the appendices.

Heuristic evaluations[1] are used on the early low fidelity prototypes. These refer to an evaluation by interface experts, based on a set of guidelines. The interface and ITs can be examined visually, via written descriptions, or through actual use, in order to determine if the set criteria are met (LaViola et al., 2009). In our work we used heuristic evaluations to limit the possibilities of the numerous ITs for selection. As a starting point we defined a model describing possible ITs for the selection task (section 10.2). In order to find out the most meaningful interactions described by this model, a low fidelity prototype based on drawings and photos served for evaluating these possible ITs (see appendix I.b). As we intend to implement a system supporting whole body interactions the most obvious method for an early evaluation was to act like using this system, without having an actual implementation. Such technique was for example used by Jeff Hawkins, who invented the PalmPilot, and carried around a block of wood in his everyday life which he used as a prop to imagine different situations of usage. The technique was later formalized and described as Playacting (Sato & Salvador, 1999). Results and more details on the used methodology can be found in the appendix, section I.

Formative evaluations[2] are used in the medium stages of development, where early system implementations are evaluated. In practice these informal evaluations serve for refining parts of the

[1] Heuristics in this context is understood as a helpful procedure for arriving at a solution but without a proof.
[2] Formative is in this context understood as informal and qualitative feedback from actual users.

implementation, design, or the metaphors on which the design was based (LaViola et al., 2009). In our work these evaluations involved observational and informal user studies using the implemented system, in order to identify usability issues with the implemented ITs. For the methodology observations of users interacting with the system and interviews was used. These included a "think out loud" approach, where the users commented on the implemented ITs while using them. Observations were used to find out and refine certain aspects of the implemented ITs. Results from these informal sessions and more details on the methodology can be found in the appendix, section II.

Summative evaluations[1] are used to compare and evaluate the various implemented techniques in the late stages of development (LaViola et al., 2009). This was carried out as a formal experiment using a structured task-based usability evaluation, including the different usability metrics presented in the previous section. This formal evaluation was based on a formal test design, with the independent variables being ITs and dependant variables the task-scenario, the environment and system. The methodology, ecological validity, results and a statistical analysis are included in the test chapter, where more details can be found.

[1] Summative is in this context understood as quantitative and comparative analysis providing proof of concepts.

PART II - Analysis

Based on the conclusions from the previous discussion on interaction techniques in VEs, the focus of this investigation will be on selection techniques. Hence the following chapter will first focus on a task decomposition of selection techniques, here the main inspiration will derive from Dough Bowman's taxonomy, and therefore a presentation of his taxonomy will first be given. From this foundation our contribution to a potential interaction selection model will be presented and discussed.

Furthermore the considerations made in the pre-analysis concerning design principles will form the bases of our effort towards a unified model of reality and imagination based principles.

Finally a requirement specification for selection techniques will be constructed from the knowledge gained in the pre-analysis and analysis, setting out the consideration needed for the design chapter.

10 Selection taxonomy

One of the key figures in the area of 3D user interfaces is Doug Bowman; he and his colleagues (2005) have investigated different existing interaction techniques and have decomposed them into a taxonomy - an approach that also will be addressed in the current context.

Taxonomies have originally been used to classify organisms, creating a structure where one can have a set of higher and lower levels of categories within a species (Encyclopædia Britannica, 2011). This notion has been adapted over to 3D User Interface by Bowman and colleagues (2005) who calls it *Generic technique-decomposition.*

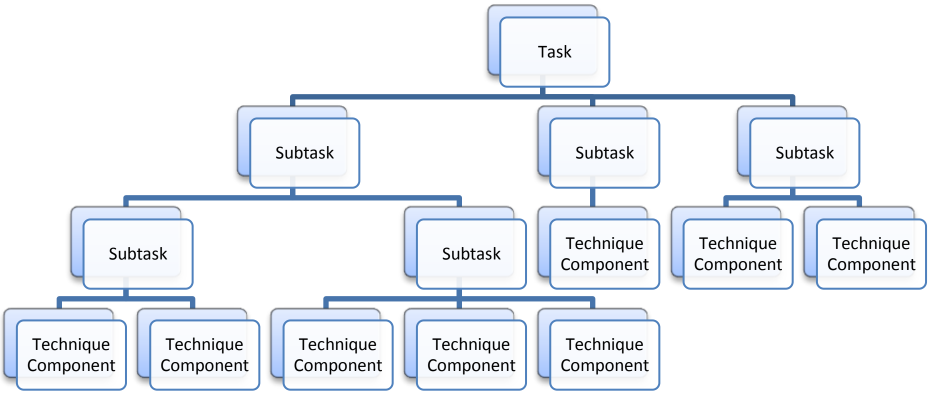


Figure 9 - Shows how each task can have a subtask, and each subtask can have another subtask of technique component. The figure derives from (Bowman et al., 2005, p.353).

Our approach is inspired from Bowman and colleagues (2005) and we make use of a taxonomy to define the selection techniques in order to fully understand all the elements that have to be taken into consideration when designing a selection technique for a VE.

10.1 Previous Taxonomies

Lindeman (1999) constructed a taxonomy of manipulation techniques. In the realm of manipulation there is also selection, meaning that a manipulation technique is a combination of selection and manipulation. Lindeman (1999) divides manipulation techniques in to two different categories, direct and indirect manipulation. His direct manipulation is determined of how closely user movements are mapped to objects movement, and indirect manipulation uses symbology or mediated tools for manipulation.

Lindeman's (1999) device oriented taxonomy focuses on input/output devices, forming the final three axes of this taxonomy which are: the *manipulation parameter* (the manipulation is direct or indirect), the *action type* (the action that is required to perform the technique is discrete or continuous), and lastly the *degrees of freedom* the user can use the technique to physically manipulate any given object.

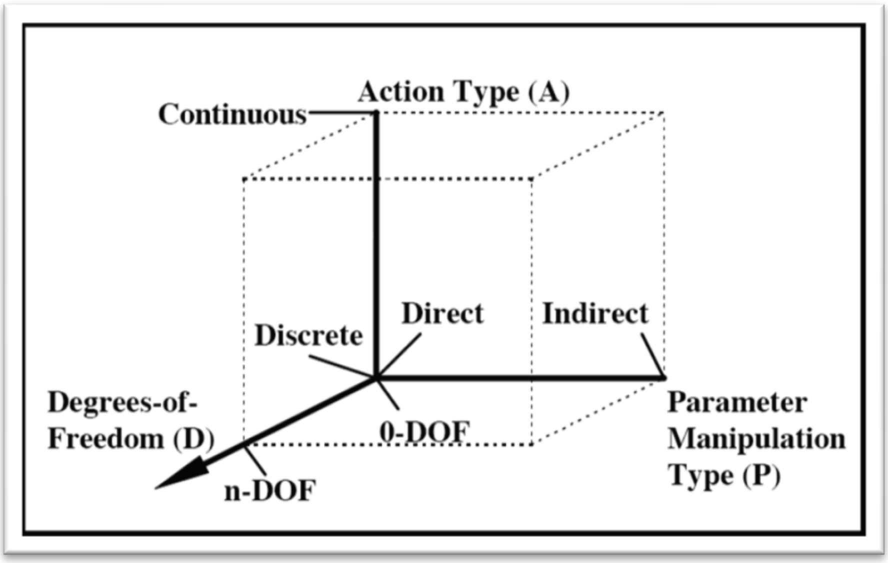


Figure 10 - Lindeman's IVE Interaction Taxonomy, having three axis, Parameter of Manipulation type "Direct - Indirect ", Degrees of Freedom "0 DOF - n DOF" and lastly Action Type "Discrete - Continuous". Source: (Lindeman, 1999)

Although Lindeman only focuses on manipulation techniques, his taxonomy points out some of the dimensions to be taken into account, when designing ITs.

Jian Chen and Doug Bowman (2009) have a different approach in decomposing interaction techniques into a series of tasks. As already introduced in the pre-analysis their description of ITs consist of four universal tasks "Navigation, Selection, Manipulation and System Control". This builds upon Bowman and colleagues' (2005) work within 3D user interfaces.

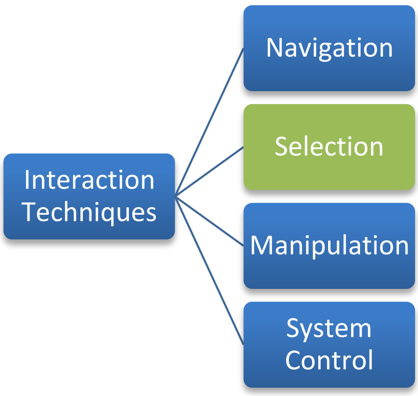


Figure 11 - The four universal tasks that according to Chen and Bowman are within the domain of interaction techniques. Selection will be the main focus of current investigation of interaction techniques. The figure derives from the article (Chen & Bowman, 2009).

Each of the interaction techniques subtasks has other subtasks. In order to construct a single interaction technique the lowest level of subtasks are combined.

Bowman (1999) constructed a structure to selection/manipulation techniques, applying a method of combining the lowest level subtasks in selection and manipulation, concluding that a total amount of plausible techniques were 4608, reducing the number with dependencies and constraints in the design space to 667. This example gives a good indication of the possible interaction techniques by combining the different low-level subtasks.

Since the focus of our work is on the task of selection, we need to fully understand selection tasks within VEs. In order to do so it is necessary to define and deconstruct the term selection. Doug Bowman and colleagues (2005) define selection, as "*Selection is the task of acquiring or identifying a particular object from the entire set of objects available… The real-world counterpart of the selection task is picking an object with a hand*" (Bowman et al., 2005, p.142) and from this definition they constructed a taxonomy of selection techniques.

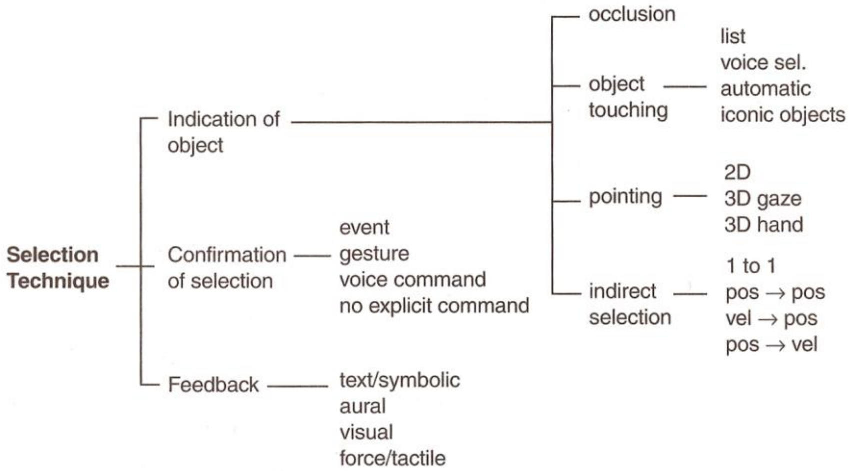


Figure 12 - Bowman and colleagues taxonomy of selection techniques, image source (Bowman et al., 2005, p.149).

Bowman & colleagues (2005) stated that selection techniques have three subtasks *indication of object, confirmation of selection* and *Feedback*. Indication of object has four subtasks and three of these subtasks have technique components. Confirmation of selection and feedback has each four technical components.

Lindeman and Bowman have two different approaches towards describing interaction techniques. Lindeman's focusing on the input/output devices to describe the techniques, the drawback of his taxonomy is that it gives a set of parameters to be taken into consideration in each technique, but it does not give a detailed description of each element in the technique, elements that can be implemented in a software environment.

Bowman and colleagues on the other hand have decomposed each interaction technique into task elements; this approach allows a good understanding of the technique and assists when implementing the technique into a software environment. Further on, this method supports the notion of reusing elements from one technique to another technique that may be needed in a specific situation or application, an important aspect that is often seen in software developments.

Although agreeing with Bowman and colleagues task oriented approach, our critique is that their model is only based on the known techniques applied into the taxonomy, resulting in a taxonomy that becomes a static model for such ITs. Thus the development of novel techniques from this taxonomy only happens by combining existing techniques.

Bowman & colleagues selection taxonomy does not pay much attention to design principles; we argue that this is an area that should be paid attention to when defining selection techniques.

They touch upon the subject when they define manipulation techniques, mentioning isomorphism in the regards of having real and magical actions corresponding to the users actions in the VE. They correlate this subject to mapping of the users physical actions to system feedback.

We agree that this is a part of the answer, but we see these mappings as a product of a set of design principles. We argue that having an interaction taxonomy with design principles as a point of origin will leave a model that will be more generic and resistant for technical developments within the field of VE. The correlation between these two will be described in details in the principles for VE & interaction techniques section.

Furthermore the results of their decomposition are lacking a more detailed description of the system responses to the users actions. Bowman and colleagues (2005) state that interaction in a selection task should "*provide the user with the means to indicate an object to select and confirm the selection; and provide …feedback while performing the task*" (Bowman et al., 2005). It becomes clear from this statement that in order for a selection task to be possible, there are two main actors in the scenario: first the user that is indicating and confirming the selection and secondly the system is providing the user with feedback on his actions; hence there is an input/output relationship between the user and system. Bowman and colleagues (2005) taxonomy is lacking a more detailed description of these actors, e.g. what are the constraints and feedback options of both parties? These missing links will be addressed in our taxonomy that will be further described in the following section.

10.2 Proposed taxonomy

The proposed taxonomy takes its foundation in Bowman & colleagues selection task taxonomy, assisted by our investigation of current selection techniques and selection task decomposition of selection techniques. Results from this approach indicated that a selection technique was a product of a more complex structure than earlier researchers have demonstrated, a structure that will be scrutinized in the following section.

We have reached the conclusion that the structure of a selection technique is as following: the lowest level is human physical and VE hardware (software) input to the system, this aspect has been introduced in the pre-analysis, and therefore not repeated in this chapter. Next level is where the main tasks of selection technique *Indication, Confirmation and feedback* take place. Higher levels are interaction techniques and finally principles of design. Further description of the different levels and relational description between the different levels will be addressed in the following sections.

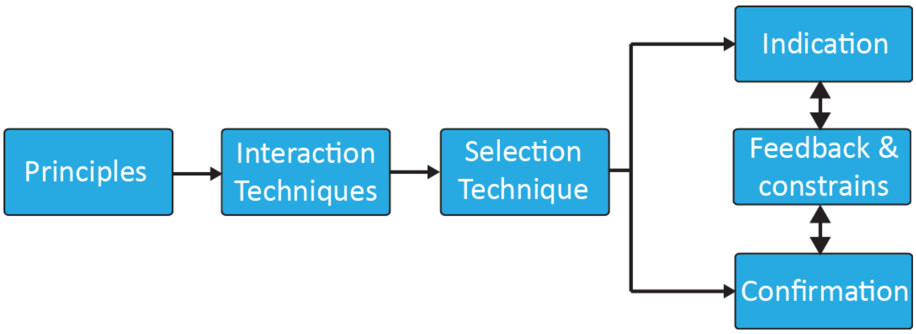


Figure 13 - The figure depicts how selection techniques can be described.

The description of the different elements that are situated in correlation with the selection techniques will be divided *low level* (e.g. indication in selection technique) and *high level* (e.g. interaction techniques) related subjects. Lastly a final relation description of how all the elements are related to each other.

Selection technique

Our proposed taxonomy is task oriented. It differs from Bowman and colleagues' in the respect that we clearly acknowledge the actors included in the different selection tasks in a Virtual Environment, whereas Bowman (1999) when defining selection "*Selection involves the specification of one or more virtual objects by the user for some purpose*" seems to have neglected the system that responds to the users actions and provides the user with proper feedback. Bowman (1999) mentions feedback as a subcomponent of selection, but regards it purely as an interaction issue, which does not correspond to the actual users' goal. We do not agree with this statement, building on the Human-Computer Interaction (HCI) notion having a system of user and computer actions. We are bound to the fact that each users action has an impact on the system and the system has an impact on the user, a symbiosis that needs to be taken into account and cannot be regarded as purely feedback of actions; thus the computer system that the VE is a part of needs to be considered when describing selection techniques.

The consequence of this notion can be seen in how the subcategories of selection techniques are related to each other. The users *indication* and *confirmation* are closely connected to the system *VE Properties*. Furthermore *indications* and *confirmations* subcategories are dependent on user and computer system constraints.

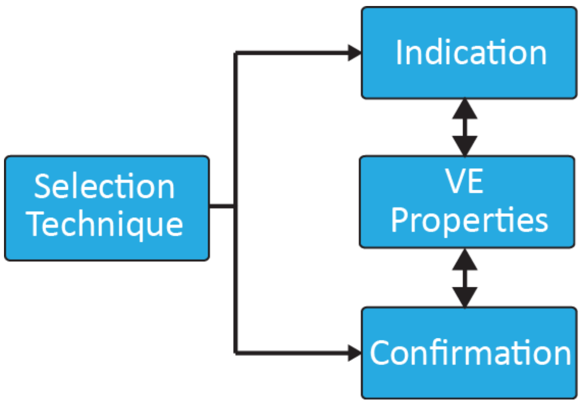


Figure 14 - The sub categories of selection technique and how they are related to each other.

Indication has two subcategories, *direct* and *indirect* selection; further details can be seen in Figure 14. We define direct selection as the task where the user needs to touch the object in order to select it. The technical component of direct selection is *object touch*, which should be considered in regard to the users' virtual attributes. If the users are able to virtually touch the object, we regard it as direct selection. An example can be the "GoGo technique" that could be considered as pointing gesture from a programming perspective - an indirect selection, but the main difference here is that the ray (which in the GoGo technique is represented by a virtual arm) is used to touch the object directly, opposite to an indirect pointing technique. In this context the virtual arm is considered as an extension of the users arm. Thus the direct selection also requires no further confirmation action but selects objects directly upon touch.

There are two technique components and one subtask in indirect touch. These are *Head Up Display (HUD) pointing* and *Gesture*. The technique component HUD is defined as a mediated method of choosing an object, here selection is possible through lists or iconic representation of the object.

Gesture can be divided into two technique components. First is posture, a static gesture that does not change position over time. An example can be pointing and the main difference between pointing in a direct and indirect selection is the feedback and constraints relation between the user and system. Dynamic gesture is considered as a posture that has a dynamic movement over time. One example of such dynamic gesture is lasso tool selection in Photoshop[1], where the user is drawing a line that is circling all the objects he wants selected.

Pointing can normally be categorized as a static gesture, but due to the vast amount of implementations versions of this technique component, it will remain as independent category. An example of pointing technique is the ray casting technique.

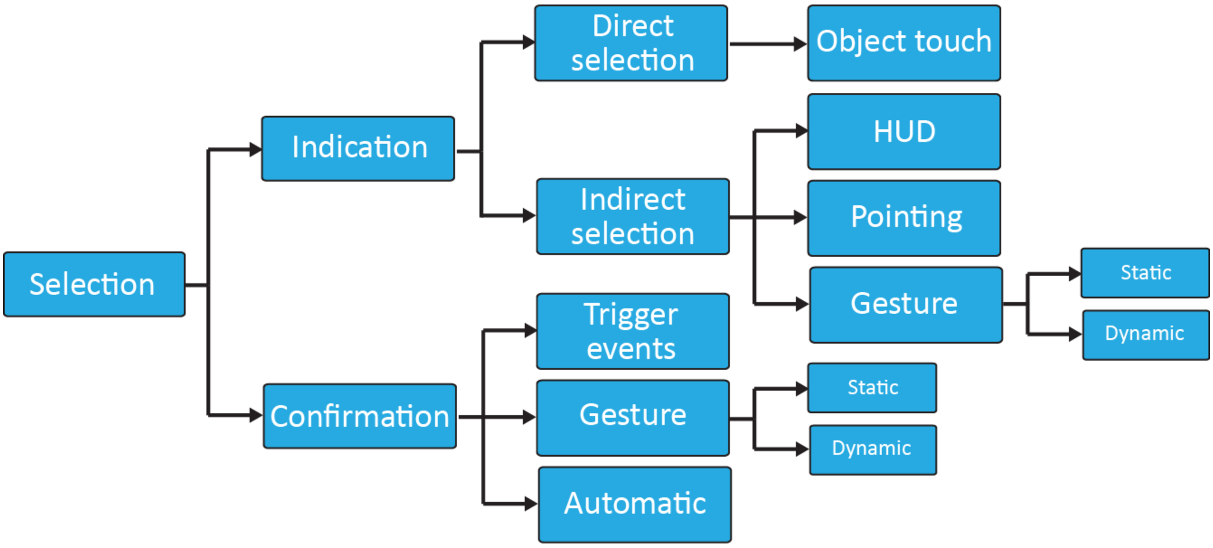


Figure 15 - Shows our proposed taxonomy of selection techniques.

The task of confirmation consists of three technique components: *Trigger event, Gesture and Automatic.*

Trigger event is to be considered as the discrete action taken to confirm the selection. One example can be the push of a mouse button. Gesture of confirmation has the same properties, as in indication, hence is the division of static and dynamic gesture; these can also be regarded as discrete and continuous actions. Automatic confirmation is either to be considered as a direct confirmation from object touch or the confirmation through a time threshold, where the user needs to indicate an object for a certain amount of time in order to select it.

As stated earlier a selection technique is a relationship between Indication, Confirmation and VE Properties. Feedback is a part of VE Properties and is regarded as the communication link between the user and computer system. A link which according to Donald Norman (1990) can be considered to be a part of design principles. Recognizing the individual designers preferences when dealing with feedback, our taxonomy of feedback will only give subtasks of the subject.

Feedback has four subtasks: *Text/Symbolic, Aural, Visual* and *Force/Tactile feedback*. Most of the subtasks are self-explanatory by their names. Aural is audio feedback the user gets when confirming or indicating a selection, such as for example found in Windows when double clicking on a folder. Visual feedback can be used in different aspects, such as a guiding tool for the user to

---

[1] A image editing software application from Adobe, http://www.adobe.com/products/photoshop/photoshop/whatsnew/

navigate in the scene with his selection tool; one example can be the ray extending from the user hand when using the ray casting selection technique. Force/Tactile feedback is the physical feedback the user gets when interacting with the system; one example can be the Nintendo Wii remote controller that is vibrating when the user is performing various tasks. Text/symbolic subtask could be regarded as a visual technical component, but we regard it as an independent subtask due to the semantically content of this task. One example of textual feedback can be a list produced to identify objects in the scene.

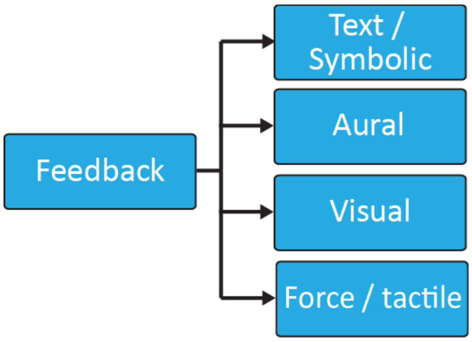


Figure 16 - The different feedback options available for ITs

System constrains are also considered as a part of VE Properties. System constrains finds it roots in Donald Normans design principles (Norman, 1990). Constrains is to be regarded as a guiding tool the system gives to the user in order for the user to more easily understand the system and to be more capable to perform actions that are a part of the designed system. System constrains have three sub-elements: *Motion, Visual* and *Aural*.

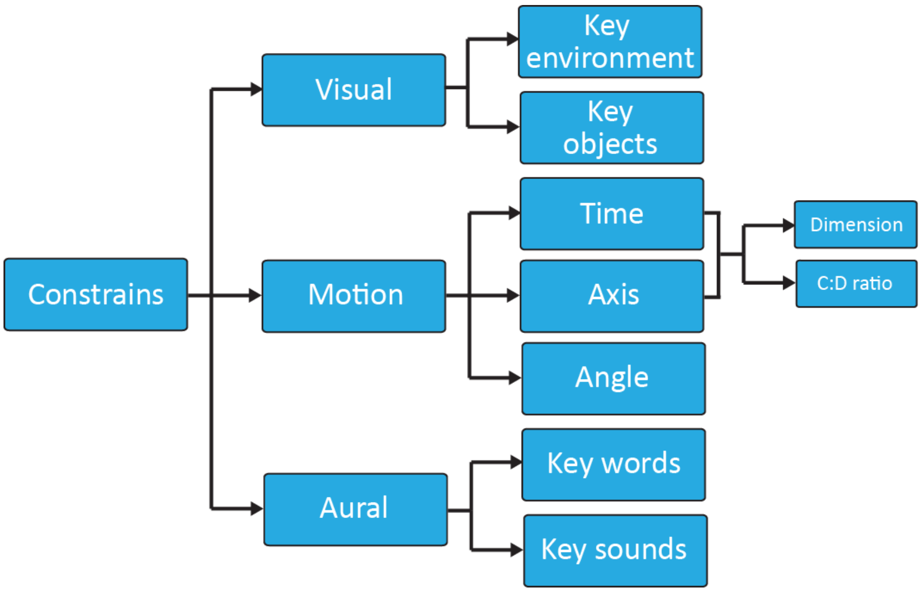


Figure 17 - Depicts the constrains that can be added to the interaction technique

Motion has three sub-elements: *Time, Axis* and *Angle*. Time is considered as sub-element of motion, due to the factor of movement, hence time can be a constrain factor in determining velocity of a movement. Another time related issue is when the user is confirming his action; time can be a determining factor in regards to respond of action.

Axis can be defined by its technical components *dimension* and *control display ratio*. In an axis point of view, a dimension is the describing factor of the virtual environment space the user is operating in. The number of dimensions describes the minimum number of coordinates that are needed to describe a point within the Virtual Environment. One example can be three-dimensional virtual environments - here a point can be described by the xyz coordinates. Control display ratio is how user actions in one dimension are mapped over the virtual environments dimensions. An example of such mapping can be the users three-dimensional actions are mapped to a two dimensional environment.

Angle has two sub-categories: *dimension* and *control display ratio*. Dimension is describing to which degree it is possible to rotate a point on each of the axis. Three-dimensional spaces have the three axis that a point can rotate around. Control display ratio is how the users rotational actions can be constrained by the system; one example can be that the users 360-degree movement on the x-axis is mapped to have a span of 180 degree in the VE.

Aural and Visual constraints are regarded to be the systems internal infrastructure of audio signals. An unconstrained audio environment within VE can be considered as a simulation of the real world. But due to the complexity of such a system, a united understanding of system commands across all cultures is unlikely to be achieved. Therefore to guide the user to understand the system, aural constraints are divided into keywords and key-sounds, while visual constraints are divided into key environments and objects.

Having described how all the sub tasks in a selection technique, it should be clear that in order to create a selection technique, elements in the indication, confirmation and feedback/constraints are needed. Furthermore the selection technique is also dependent on the hardware and user input to the system. The overall relationship is described in Figure 18.

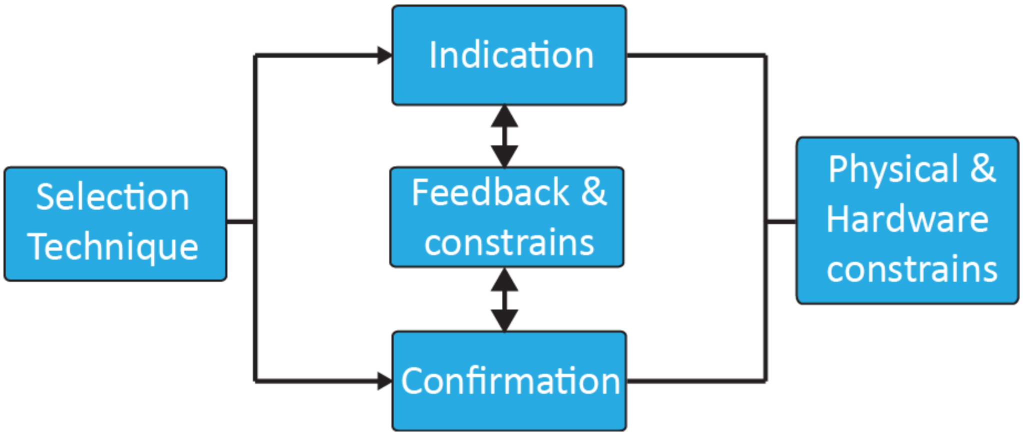


Figure 18 - The relationship between the elements that constitute a selection technique

Further consideration are needed when creating a selection technique, and that is design principles, a subject that has been touched upon in this section but will be further described in the following section.

11 Principles for VEs & Interactions

In this section we will present our idea based on a combined model of reality and imagination based principles. This model was derived from the principles described in the pre-analysis, section 2.

The goal of our model is to describe the principles that are particularly suited for body controlled interaction in VEs or more generally in 3D interfaces. As already described in the pre-analysis there are two opposing set of principles: reality based (RBI) and imagination based (IBI) principles.

The source of our model is the RBI framework (Jacob et al., 2008), which is described through the four main categories: *naive physics, body awareness, environment awareness* and *social awareness* (see pre-analysis section 2 for a full description). From these RBI principles we derived the imagination based principles (IBI), which fits to each of the RBI categories and can be considered as an opposition to the RBI principles. Our proposed model can be seen in Figure 19. In the following paragraphs we will explain this model and give some practical examples on how to utilize such principles in the design of ITs and VEs.

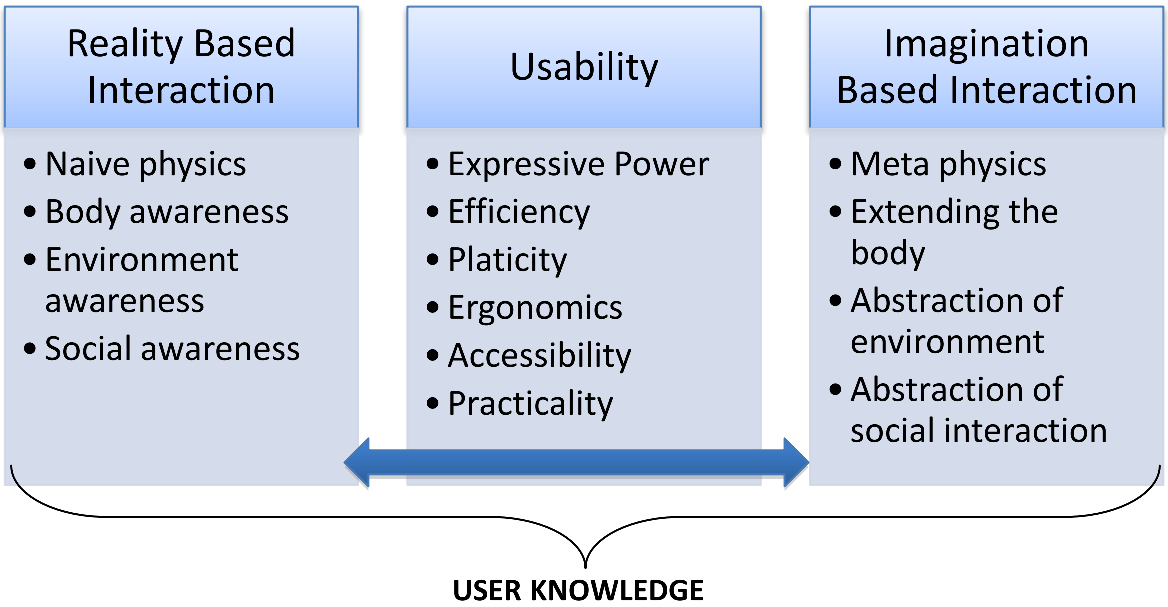


Figure 19 - The relationship between reality and imagination based principles and their relation to usability

All principles in our model utilize the users pre-existing knowledge. The RBI principles build on users everyday knowledge and it can be considered as expectations towards the system of what the user normally would encounter in the real world. The IBI principles on the other hand build upon the users' knowledge from various other sources than the real world, such as movies, games, books, dreams or in general imagination. It is important to note that this knowledge is highly subjective and thus the utilization of principles should always be considered in a cultural context. The RBI principles may though provide a greater common cross-cultural understanding than the IBI principles, as they build on the more direct perceivable phenomena than principles rooted in imagination. We placed the usability between the IBI and the RBI principles, since we consider the usability as the tradeoff between the two opposing set of principles. In this context we want to highlight that basing a system solely and strictly on either of the two opposing principles may result in a loss of usability. This will be further explained throughout the following sections.

11.1 Naïve physics and Meta physics

Basing ITs on the naïve physics principle should be considered as mimicking the expected real world behavior. For example could an interaction with a virtual object result in a physical response of the object. Considering manipulation techniques, the object could for example be rotated by direct touch and should act under the influence of gravity, friction and the users input forces. For selection techniques the principle could be utilized by providing physical response as feedback on a selection action. For example could a push result in a physical corresponding action of the object. By this the direct selection, which we describe in our taxonomy, is highly connected to the naïve physics principle, since it relates to the corresponding real world action of selecting objects which is carried out by simply grabbing or touching a real object in order to manipulate it.

The opposing principle to naïve physics is in our model considered as the principle of Meta physics. Under this category most of the principles as described by Alexander Kulik (Kulik, 2009) can be found. These included: *suspension of naïve physics, scaling of geometry and motion, automation, magic spells,* and *mode changes.* A more detailed explanation with examples can be found in the pre-analysis section 2.2.

11.2 Body awareness and extending the body

The principle of body awareness refers to familiarity and understanding that users have of their own bodies (see pre-analysis section 2.1). Drawing its strength from proprioception, the principle is especially used in many of the ITs found in VEs. For selection techniques, the principle is present by using arms and hands to point at objects, in contrast to the mediated (through the mouse) style of interaction found in the WIMP paradigm. The principle can be supported by providing the user with feedback of his/her body, which in VEs often is utilized through displaying an avatar of the user. The use of full-body motion tracking further supports this concept, where users' body motion can be directly translated into the system. We consider the RBI principle of body awareness as building strongly on user expectations of his/hers own body and its virtual counterpart. Utilizing this principle to its full extend would be to represent the user one to one in the VE, meaning that the avatar by which the user is represented actually matches the user.

The principle of body awareness can be bent and extended, and the user may accept a variety of representations to be him/her-self. Thus the opposing principle to body awareness can be considered as an abstraction of the more realistic representation of the user. By this the principle of "extending the body" builds on the concepts found in ecological perception and embodiment. As almost anything is possible, in a VE the principle can take various forms, were simple forms of utilizing the principle may be to extend the arm of a user to allow him/her to reach a distant object. More abstract forms could even be to replace the users' representation entirely, which for example could allow the user to "be" an Orc, a Worm, a Tiger or almost any other object thinkable. By such the principle still draws its strength from proprioception, but translates it to an entirely different representation. Here it is especially important to map the input so that the user still can make sense of body actions and the corresponding actions (feedback) of the representation. Practical considerations of the body awareness and extending of the body principle should always include mapping and feedback considerations. For example the users hand position can be directly mapped to the position of a cursor for selecting objects. In such example the users hand could be represented by a cursor, which can be represented through the classical mouse courser icon or through a more matching representation of a "hand" icon. Also in many situations it may not be necessary to actually represent the user through an avatar, but the body parts used for an interaction should always be taken into consideration when choosing feedback, so the user can easily relate his actions to what is going on in the system.

11.3 Environmental awareness and abstraction of environment

The environmental awareness principle builds on the concept that humans are aware of their presence in an environment, surrounded by objects and landscape. The environment provides the user with clues, which facilitate the human's sense of orientation and spatial understanding. In this context the VE provides the user with a range of clues, which in turn help the user to navigate and understand the spatiality of the environment. Besides depth clues, such as shadow, lighting and horizon (see pre-analysis section 2.1 for details) the principle also regards the virtual objects in the VE. Here a realistic environment can build on the users pre-existing knowledge of the real world and thus provide better understanding of the users orientation and spatial understanding. On the other hand computer graphics allow for abstraction of this environment, where the user could be situated in virtually every thinkable environment. Such abstract environment can still utilize depth clues to provide the user with at least a minimum understanding of his/her orientation.

11.4 Social awareness and abstraction of social interaction

The principle of social awareness describes the social interaction and a user's general awareness of the presence of others (as described in pre-analysis section 2.1). The principle includes verbal and non-verbal communication, the ability to exchange physical objects with others, and the ability of collaboration (Jacob et al., 2008). Depending on the technology, the communication in a system can be direct co-located or mediated, and it can also include verbal and non-verbal communication. In a system using full-body input, the mediated communication can even be through body language of the users' avatars. Our model includes the principle of abstraction of this social interaction on the opposite side of social awareness. In VEs where the user's avatar can have virtually any form or character, we think that the communication will reflect upon that changed representation. In that sense users could communicate and interact through more abstract means, whereas we think of the RBI principle of social awareness as the more direct way of communication - through language, words, body language - mimicking the social interaction found in everyday life. For the design of ITs, this principle is reflected in the choice of how users should interact with each other, and also interact with computer characters (NPCs - non-player characters).

11.5 The usability tradeoff

Our model should not strictly be understood as a scale going from reality to imagination, but more as a description of two extremes where many in-betweens may exist. Also we think that every choice of a set of principles should always include usability considerations. In this way basing the system on e.g. the strict RBI principles may introduce a lot of usability issues. For example could a technique for navigating through a VE, based on the RBI principle - mimic the real world - so that in practice the users navigate by walking or running. This form of navigation can be very unpractical, thus the usability of the navigation method may be very ineffective. On the other side basing the navigation method on the strict IBI principle may also lead to issues in the usability. Examples for this have been studied, where users were confused when the navigation technique directly teleported them to their designated target location (Bowman, 1999). In general the designer should always have the usability in mind when designing ITs.

12 Requirement specification

The intention with our taxonomy and the described principles in this analysis chapter is to be used in the practical design of selection techniques. As already described in the problem statement and methodology (section 8 & 9) our goal is to implement and evaluate a set of ITs. Since there are many possibilities for creating techniques we define a set of requirements for the design and implementation of these techniques. This is also helpful to limit the amount of techniques and to focus on the design.

- The selection techniques should reflect all major aspects of our taxonomy, which are:
    - A technique using direct selection – by object touch;
    - A technique using indirect selection – by a HUD element;
    - A technique using indirect selection – by a pointing method;
    - A technique using indirect selection – by gestures.
- Principles should be applied in the design of the techniques whenever possible.
- The selection techniques should be usable for both single and multi-selection scenarios.

Requirements to the implemented techniques are that they meet a set of usability criteria in order to actually be usable in an evaluation context. The general usability criteria to the techniques are:

- The techniques should be easy to learn.
- The techniques should be easy to remember.
- The techniques should be effective to use.
- Using the techniques should possibly not result in making selection errors.
- The techniques should be pleasant to use and comfortable.

With the goal to develop techniques for 3D interfaces and VEs it is required to design and implement a 3D environment. As we do not work with a specific context or a specific focus with the environment, the requirements to it are broad. The goal of the environment is in general to support depth perception and to allow easy selection of objects placed in this environment. With our focus on interaction and the evaluation of these, we set out to keep this environment simple. This has the advantages of not requiring an extensive amount of cognitive resources from the user and also the intention was to keep the environment as the dependent variable in the test design. The requirements to the environment are the following:

- The environment should support depth perception.
- The environment should be simple to not confuse the user.
- The objects which users are able to select should be easy identifiable.

Besides the above described requirements to the design, another set of requirements to the hardware and implementation are described. These requirements are based on the input and output method used in the prototype – and describe the more general requirements to the prototype developed during this thesis. Since we are using optical markerless motion capture with the Kinect camera as input, requirements to the prototype are that:

- The input delay (response time of input or ping) should be kept as small as possible.
    - At least below 500ms.
- The ITs should be designed to work well with the method of tracking.

The input delay is important to keep low since high response time could cause users to be confused. Executing a gesture or an action should in this way always have an immediate result in the system.

Requirements to the prototypes output method are:

- Providing an adequate real-time frame rate (~30+FPS).
- Supporting a stereoscopic screen-display method.

PART III - Design

The structure of the design process used for creating selection ITs was based on an iterative process, as described in the methodology (section 9). In this process we used several creative methods to create the ITs. For the initial ideas brainstorming was used, whereas later concepts were refined through playacting and drawings. Throughout this process we created concepts for around 12 different techniques. Some of the techniques we include are already existing techniques or based on them, where others are new techniques. In the iterative process we cut down the number of techniques and evaluated the most usable and feasible for implementation. The outcome was 5 techniques which were implemented and later tested in a user study. All of the techniques were created in the scope of our taxonomy and principles were applied in most of them. In this chapter the theoretical design of these five techniques are described, along with a range of design considerations discovered when creating the concepts of these techniques.

13 Design framework

With the described taxonomy selection techniques were separated into *indication* and *confirmation* actions. The goal of the techniques was to cover each of the elements found in indication: *object touch, HUD, pointing* and *gesture* (Figure 20). The confirmation actions were designed separately and were assigned to the different techniques based on a usability evaluation with real users. For the confirmation actions we excluded trigger events, since they would require a discrete input device such as a button.

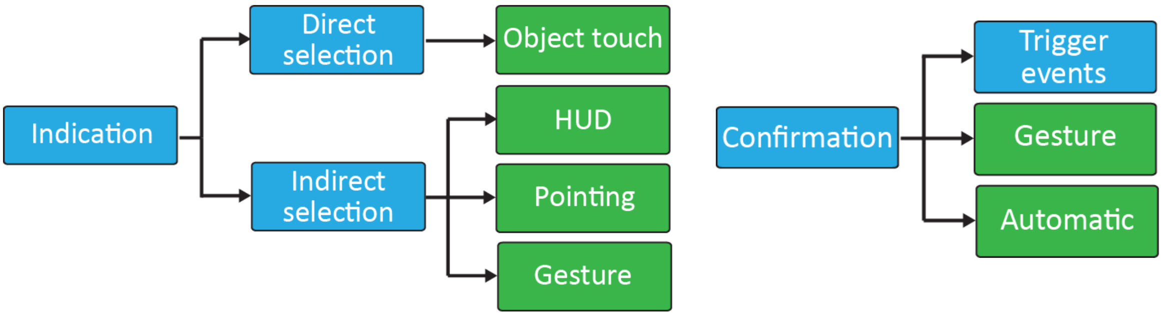


Figure 20 - Targeted indication and confirmation actions

For each of the indication methods constraints can be added, in order to guide the user into action. Also each IT require some form of feedback. In our design we included only visual feedback, knowing that the use of multi-model feedback possibly might enhance the overall usability.

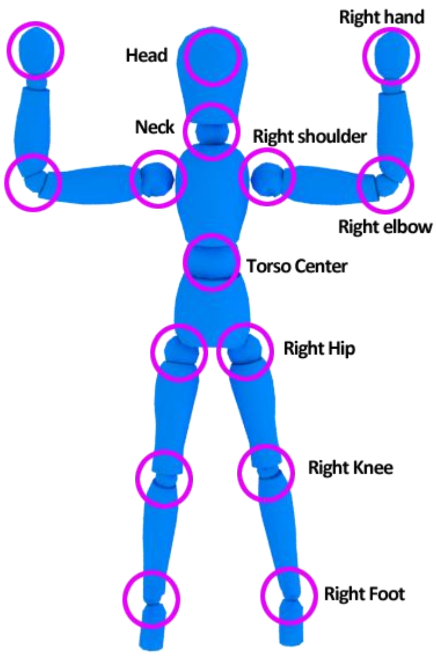


Figure 21 - Input joints available through the NITE skeleton tracking framework

13.1 Input and output

With the motivation of designing for controller free VE interaction, we optimized the ITs for screen display and for the input we limited the design space to the available skeleton tracking method (see pre-analysis section 6.1 and

Figure 21) for the Kinect camera. It is important to notice that a better tracking method could include tracking on more detailed joint parts, such as finger position or hand rotation etc., which in turn would give more possibilities in the design of interactions. Nevertheless are the most important body parts included in the tracking, mainly the hands and arms as well as the head position and rotation. Throughout the design cycles we tried out different body parts for indication actions, such as e.g. the head rotation. We came to the conclusion that using other body parts than the hands and arms (e.g. the head rotation) was not very usable and lacked of accuracy for fine tuned selection tasks. The different techniques described throughout this chapter therefore mainly use the arm position, rotation and hand position for interactions.

14 Confirmation actions

The confirmation actions are thought of as the actions users need to perform in order to confirm the indication of an object to consequently select it. This is similar to the actions users perform in desktop computers, where indication is carried out by placing the mouse curser on top of an object (often icons) and the confirmation is carried out by pushing the mouse button.

The two categories for confirmation actions are gesture and automatic. With the automatic actions, either direct object touch or confirmation through a time threshold is considered. The gesture confirmation however can include numerous possible actions; in the following a short explanation of the designed and implemented actions as well as some other possibilities for gestural confirmation is given.

We divided confirmation gestures in two rough categories: simple and advanced gestures. The simple gestures include actions like raising a hand, pushing, pulling and swiping. The advanced gestures include more complex actions like e.g. drawing a question mark, a circle or a confirmation sign in the air.

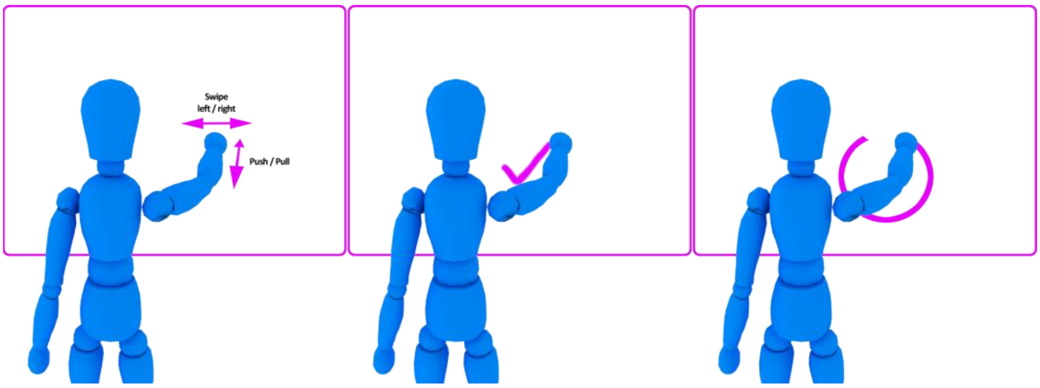


Figure 22 - Confirmation actions

The swipe gestures are well known from touch enabled mobile phones, which are defined by a sideways movement of a finger. In our case the swipe gesture is carried out by moving the hand to the right or to the left from its original position.

The push gesture is carried out by moving the hand forward. During the design we tried out different sizes of motion required for this gesture, and most users preferred to have only a small movement or tilt of the hand as the push gesture. Bigger motions of for example moving the whole arm forward showed to be not feasible in connection with most indication actions. The pull gesture is similar to the push gesture carried out by moving the hand backwards.

The advanced gestures that were discussed during the design of techniques were all based on a motion trajectory or path, meaning that these are actions where the user should draw some kind of symbol in the air or do a defined kind of motion. The motions that were considered for confirming a selection were to draw a check sign and circling the selection. These were however disregarded because they showed to not work well together with the indication actions.

15 Selection techniques

In the following sub-sections the design concerning the five techniques that were developed, implemented and evaluated in this thesis practical work are explained. The structure of each technique is divided into indication and confirmation actions. For each technique a short explanation and discussion of the feedback, constrains and the principles utilized is described.

15.1 Pointing techniques

Two pointing techniques were developed, one similar to the ray cast technique and the other resembles a mouse like behavior. We will refer to the ray casting technique as: "**pointing technique**" and the technique resembling mouse behavior will be referred to as the "**mouse technique**".

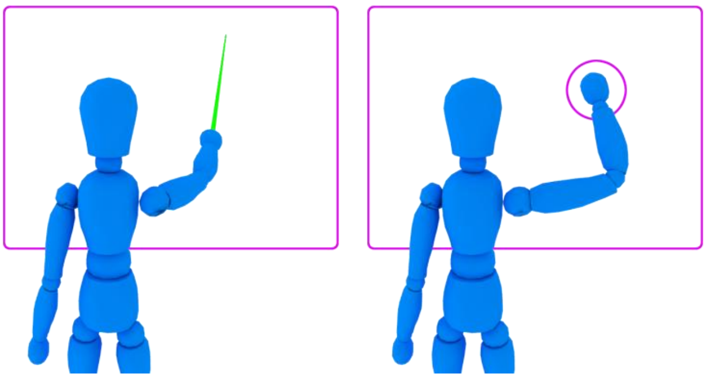


Figure 23 - Left shows concept of pointing technique - right shows concept of the mouse technique

Methods of indication

The pointing technique uses a ray that extends the users arm to indicate objects. This ray checks with the scene if objects are intersected. In this way the user can stand and remotely point at distant objects. The technique does not require the user to navigate to an object and can therefore be very effectively used for selecting objects.

The mouse technique uses a more direct mapping, where the hand position is mapped to the screen. A cursor is used in order to show this hand position (marked with a circle in Figure 23 right). When the hand is moved outside the screen area the cursor will still be shown on the side where the hand left the screen area, much like in most operating system.

Methods of confirmation

For the pointing technique the timed automatic confirmation was assigned. The reason of this choice was that when indicating distant objects it requires holding the arm steady to not point at a different object, so that the timed confirmation is useful since it does not require an additional hand or arm motion.

For the mouse technique the push action as confirmation was assigned. The reason behind this choice was that users could relate to this push action as clicking a mouse, so that the technique becomes more intuitive and easier to remember.

Feedback

Feedback for the pointing technique included a visual ray from the users hand position in the direction of pointing. The confirmation feedback indicates how long the user is supposed to hold his hand in order to confirm, by showing a small time clock with the time passed and remaining time left in a circle. For the mouse technique the choice was to only show feedback in form of a cursor. This was designed to resemble a hand, so the user easier could form a mental model of the technique.

Constrains

For the pointing technique no constrains were added. However, the possibility of adding motion constrains may become helpful in order to support selection of very distant objects. As the ray extends the arm, small arm motions become magnified when the distance to objects increases. For very distant objects it may thus be necessary to support the stability of indicating objects by adding C:D gain (control - display ratio) constrains or to simply filter the arm motion.

For the mouse technique C:D gain constrains were added. This resembles the functionality in most operating systems, where users can adjust the speed of the mouse cursor. The C:D constrain for the mouse techniques means that changing the ratio results in a changed mapping of real hand position to the corresponding cursor position. Increasing the ratio results in a smaller area of real hand movement mapped to the corresponding cursor, thus the cursor moves faster when moving the hand. Decreasing the ratio results in a larger area of real hand movement mapped to the cursor position, thus the curser moves slower and the hand motion for controlling it spans over a larger area. Adjusting the C:D ratio thus allows the user to either perform smaller fine tuned cursor movements or bigger coarse cursor movements.

Principles

The principles utilized in the pointing technique were mainly focused on the usability. Although it can be argued that the ray extending the hand somehow represents a magical element (IBI principles). This principle could further be highlighted by adding visual effects to this ray - as e.g. changing the feedback for timed confirmation to a magic spell with sparks emanating through the ray when pointing at an object, provoking imaginative associations to the magical domain in the user.

The mouse technique strongly builds on the user knowledge from using WIMP like operating systems. The principles in our case were focused on the usability. However the technique may be good in supporting RBI principles through the utilization of affordances. As the action used for confirming with the mouse technique was a push gesture, this action could further be enhanced by using principles and affordances. The RBI principles could e.g. be to have a physical response of the confirmed objects; in the case of a push the objects could utilize naïve physics principle by responding to a push gesture in a physical manner. Similarly confirmation actions for the mouse technique could utilize the physical principles in the affordances, so that the objects afford a certain confirmation action. This could e.g. be done through the visual design of the selection objects, where a pulling gesture could be afforded through objects with handles, pushing gestures through buttons, swipe gestures by sliders, etc. Such affordances are often utilized in operating systems were buttons are designed in a semi three-dimensional fashion (2.5D) and pushing the button result in a pseudo physical response of this button.

15.2 Object touch technique

For the direct selection of objects through touch the concept was to extend the users arm to directly touch objects. Since we did not include navigation techniques, this was the option to directly touch objects (at-a-distance). The other case of object touch would be a technique were the user navigates through the scene and simply touch objects directly (local), resembling the real world behavior where desired objects are interacted with directly. The technique of extending the users arm was originally developed under the name "**GoGo technique**" which it will be referred to in the rest of this thesis.

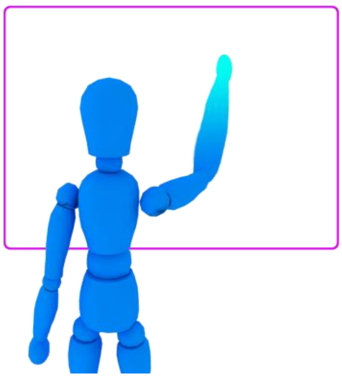


Figure 24 - Concept for the GoGo technique

Method of indication

Similar to the ray casting technique the GoGo technique extends the users arm, but instead of a ray - the arms representation is extended or contracted by a non-linear function. In this way the user can control the length of the arm by either moving it forward or backwards.

Method of confirmation

The method of confirmation for the GoGo technique is automatic. The objects will be selected when touched with the virtual hand placed at the end of the extended virtual arm. In other VE interactions this technique is often used in conjunction with manipulation techniques, so that the user can directly manipulate the object upon touch. As our focus was set on the selection part the choice was to only select objects upon touch.

Feedback

The visual feedback for the virtual arm should possibly resemble the look of a real arm, so that users easily can create associations with their arm movement and the virtual representation. However, it can be argued that this visual representation must not necessarily represent a real arm; as long as the user can associate his arm motions to the visual representation more simplistic or advanced visual representations may be possible. This will be further discussed in the principle paragraph.

Constrains

Adding constrains as e.g. the C:D ratio gain adjustment to the arm can be beneficial when high accuracy is required. In our case we replaced the non-linear function for extending and contracting the arm by a function which extended or contracted the arm directly to the object of interest (similar to the HOMER technique). This has the benefits of providing higher efficiency, since the user must not wait till the arm has the intended length. The desired length of the arm can be calculated by finding the object nearest to the pointing direction and setting the length matched to that objects position.

Principles

The principles applied in the GoGo technique are imaginative. Extending the arm is strongly inspired by the principle we described as “extending the body” and it builds on the embodiment approach. Touching the objects directly however can be classified as a reality principle, since it resembles the actions performed in a real world situation. Furthermore the imaginative principles could be stronger emphasized by changing the visual representation of the virtual arm. Since the name of the technique is inspired by the animated television series of “Inspector Gadget”, the imaginative principles could e.g. be further supported by resembling the gadgets the inspector uses in the series, which look like a telescope arm (Figure 25). Pseudo physical principles could be supported by adding a physical response to the object upon touch.

Figure 25 - Inspector Gadgets telescope arm (Image source: http://www.tons-of-toys.com/inspector-gadget-2_i6136)

15.3 HUD technique

For the HUD techniques selection of objects is performed indirectly through either a form of menu / dropdown list or through the selection of object representations such as icons, small copies of the objects or similar. Our choice was to base the HUD technique on an effective menu system. This system was based on a circular menu, we named it accordingly: "**circle-menu**".

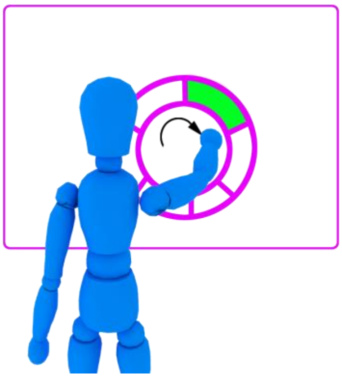


Figure 26 - Concept for the circle-menu technique

Method of indication

The circle-menu consists of multiple circular shapes representing different levels of objects. The intention is that lower levels can be used to selected categories of objects whereas the higher levels can select single objects. The lower levels are placed near the center of the circle and higher levels grow outwards. In this way users can access categories of objects by moving their hand in a circular motion around the center of the circles. Moving the hand outwards will select and show the content of a category. This content can then further be navigated by circular hand motions to select single objects. The intention with this layout is that users can open the menu, placed directly at their hand, then make small motions to select objects. By this the distance between indicating objects is kept on a minimum which should make this form of selection very effective in conjunction with Fitts' Law (MacKenzie, 1992).

Method of confirmation

In order to select the objects indicated on the circle-menu, the user needs to perform a push-gesture. The menu can be opened and closed by swipe gestures. In the different design stages we also tried several other options of opening this menu, e.g. by automatically opening the menu when raising the hand, as well as confirmation of selection by a distance threshold to the center. Although these confirmation showed to be equally effective we settled on the push and swipe options for the final design of the circle-menu.

Feedback

The choice of feedback for HUD selection is a difficult. Many examples build on either textual or symbolic feedback. For example do most drop-down menus use text representation of the objects which can be selected; in the context of 3D modeling applications this is often used to select objects which have been given a name (e.g. Box001 etc.). In a VE context different methods with representing a miniature version of the object has been used, as e.g. the voodoo doll or world in a miniature technique (see pre-analysis section 5). Symbolic feedback could be used by creating icons for the objects that can be selected. For our circle-menu we tried out different options. One used textual feedback in the form of numbers. These numbers were displayed on the objects in the scene and in the menu. By this users can look at the scene and find the objects number indented for selection, and then select the corresponding number in the menu. This kind of feedback can be used, but it showed also to cause small confusion when displaying the circle-menu as an overlay on the scene. Finally we settled on using color highlighting for the menu. In this way hovering over a certain item in the menu will highlight the corresponding object by color in the scene.

Constrains

Constrains utilized in the circle-menu are a product of the mapping. As the users hand position is mapped to selecting the objects corresponding to the angle of the hands XY position to the center of the menu, it is constrained to the plane (2D). The unconstrained version of the circle-menu could be a spherical menu where another dimension of control is added to the menu. In our case we used this circle-menu as a form of flat (2D) GUI element, also with the intention of representing a HUD element.

Principles

The principles targeted with the circle-menu are strongly focused on the usability aspects. Through the layout and the form of the menu the intention is to have a very effective and comfortable way of selecting screen content. The possibility of supporting imaginative or reality based principles exists, though not utilized in our case. The imaginative principles could for example be targeted by changing the form of feedback, to e.g. using the world in a miniature style or a voodoo doll like representation of the objects, in this way utilize meta-physical principles (scaling of geometry). Reality principles could be targeted by adding naïve physical response to elements in the menu.

15.4 Gesture technique

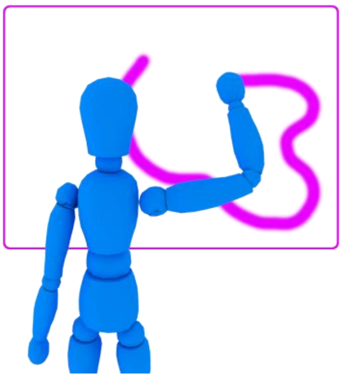


Figure 27 - Concept for the lasso technique

For the gesture technique we developed a technique which resembles the behavior of lasso-selection, often found in graphical editing software (e.g. Photoshop[1]), hence we gave it the name

[1] http://www.adobe.com/products/photoshop.html

"**lasso technique**". This technique was developed with the intention to be effectively for selecting multiple objects.

Method of indication

In the lasso technique the selection is indicated by drawing. Here the hand position is mapped to the screen and by moving the hand the selection is drawn. The concept of the lasso technique is that the objects inside this drawn selection are selected. In this way the user outlines or circles the selection.

Method of confirmation

For the lasso technique we had to define two methods of confirmation, one for starting the drawing of the selection and one for ending the drawing. Throughout the different design stages we tried out combinations of push and pull gestures for this. One way was to perform a push gesture to start the tool and then to perform a pull gesture to end the drawing and select the content within the drawn selection. The idea behind this was build on the concepts of throwing a lasso, where first the lasso is thrown around something (push) and afterwards it is pulled back. This concept was later changed to start and end the drawing both with the same push gesture, since many users were confused by the two different actions.

Feedback

Feedback necessary for the lasso technique is the drawing or path the user applies through hand motion. Consideration for the visual design of this path should include occlusion. In our case we settled on a semi-transparent path in order to not occlude the view of the scene by the drawing. We also experimented with different visual representations of the path, giving a more lasso-like feeling to the technique.

Constrains

For the lasso technique no direct constrains were added, however there are a number of possible constrains which could benefit and support this technique. One possibility is to constrain the indication (drawing) in some form. This could e.g. be to only allow the drawing of straight lines, as also found in some other lasso tools where line segments are used to indicate a selection. Another possibility is to constrain or smooth the motion of the drawn path by spline interpolation or similar polynomial functions.

Principles

The principles in the lasso-technique might be mostly influenced by usability aspects. However, we tried to utilize imaginative principles in our design, in the form of the feedback given after confirming a drawn selection. Here we added a magic-spell like visual effect to the confirmation in order to support imagination and to easier identify the techniques concept.

16 Virtual Environment

The design of the virtual environment had the goal of being used in the usability study. With this goal the design of the VE was intended to not place unnecessary cognitive load, nor confuse the user. By this the design of the VE was tried to be kept as minimalistic as possible, with the requirements of supporting depth perception. Our choice was to use a simple quadratic room as the environment. The depth perception was supported by adding parallel lines to the wall texture used as monocular cues for perspective and convergence. Shadows were also added, which serve as an important cue for depth perception. Through the placement of light and the used shading the depth perception was further enhanced. Another important feature for depth perception that was added was the use of motion parallax. This was done by mapping the virtual camera to the tracked head position of the user, thus giving motion parallax when the user moved.

Another important factor of the VE was to design the objects which users could select, through the different ITs. These objects were designed, similar to the environment, as cubes. This choice was made in relation to have easy identifiable objects. Since the goal of the environment was to be used in a test scenario, the choice of simplicity was more important than visual fidelity. In a real application this VE could be much more complex, an element which could add to the ecological validity of the test. Also the principles described could be fully utilized in the design of a complex world in conjunction with the techniques, which we see as an important point for future work.

17 Discussion of techniques

The design of the selection techniques was part of an iterative process based on the proposed taxonomy. The techniques therefore represent the two main categories of this taxonomy: direct selection and indirect selection with corresponding subcategories: object touch, HUD, pointing and gesture.

Although the taxonomy was used as the framework for designing the ITs it still is very broad - not suggesting particular details for the different techniques - leaving room for creativity.

We consider the proposed selection technique model as support tool to the creative process of creating techniques for VEs. This model could be further developed to give the designer suggestions toward which elements a selection technique should contain in order to fit a specific purpose. In our case a selection techniques that is usable and preferred in single and multiple selection tasks. These recommendations will be supported by research done in regard to that specific subject, meaning that any potential findings in our final evaluation can be used to further define the model in regards to design recommendations for general selection techniques in a VE.

It should be noted that the design of the selection techniques is in respect to the final summative evaluation; hence the design will contain elements that in normal situations would be regarded as bad design. One example is that the current design of the techniques does not contain an option for the user to revert from a wrong selection. The reason for this is due to the test design, where selection errors are recorded without being able to revert from them. Having theses design considerations in mind the next step will be to implement them.

PART IV - Implementation

Based on the design consideration done in previous chapter, the implementation will be presented in regards to how the selection techniques were implemented in order to be a part of a final evaluation. First an overview of the implementation structure will be presented, depicting the input sources, input data handling and output rendering. Then the OpenNI and NITE framework will be outlined in regards to the data provided and its usage for the selection techniques. Followed by the gesture section, which describes the confirmation actions used throughout the different selection techniques.

In the selection techniques sections, the implementation of previous design considerations in regards to selection techniques and user interaction with scene content will be described. The final implementation, containing considerations to how the 3D scene is organized and how the techniques are presented to the users during the evaluation will be described in the test implementation section. The output hardware will be described in the stereoscopic rendering section.

Finally a discussion section will reflect on all the decisions taken in respect to the implementation elements, detecting any potential weaknesses that should be taken into consideration when analyzing the evaluation results.

Code examples are given throughout this chapter, showing details of our implementation where relevant. All the code for this implementation can be found on the CD.

18 Structure

For the implementation Unity3D (Unity3D, 2011) was used. Unity3D is a game development tool which allows creation of graphical content, through an easy to use graphical interface. Though it is intended for the creation of games, its versatile possibilities of using imported .dll files and the access to all important parts of the engine, makes it an optimal tool also for the creation of novel interaction interfaces. Unity3D allows code access through C#, java and boo. For our implementation all code parts were written in C#.

The overall structure (see Figure 28) of the system uses the Kinect as input device, connected to a computer. The Kinect communicates with the OpenNI framework (OpenNI, 2011) through a customized driver, an unofficial open-source version[1], since the Kinect at the time this master thesis was written had no officially released drivers yet. The openNI framework gathers the data from the Kinect and communicates with the NITE framework. The NITE framework is responsible for the tracking; it uses an algorithm (Shotton et al., 2011) to find joint positions and rotations in the depth map. This NITE tracking is based on a C implementation and is used as imported .dll (dynamic linked library) in Unity3D. In Unity3D the data from tracking is normalized and used as input for a skeleton representation. From this skeleton the gesture detection and the interaction techniques take their data and allow interaction with the scene. The output from Unity3D is rendered on a Geforce graphics card running Nvidias 3D vision system, which automatically creates the stereoscopic images, based on the depth buffer. These are outputted on a wall projection and viewed through the shutter glasses.

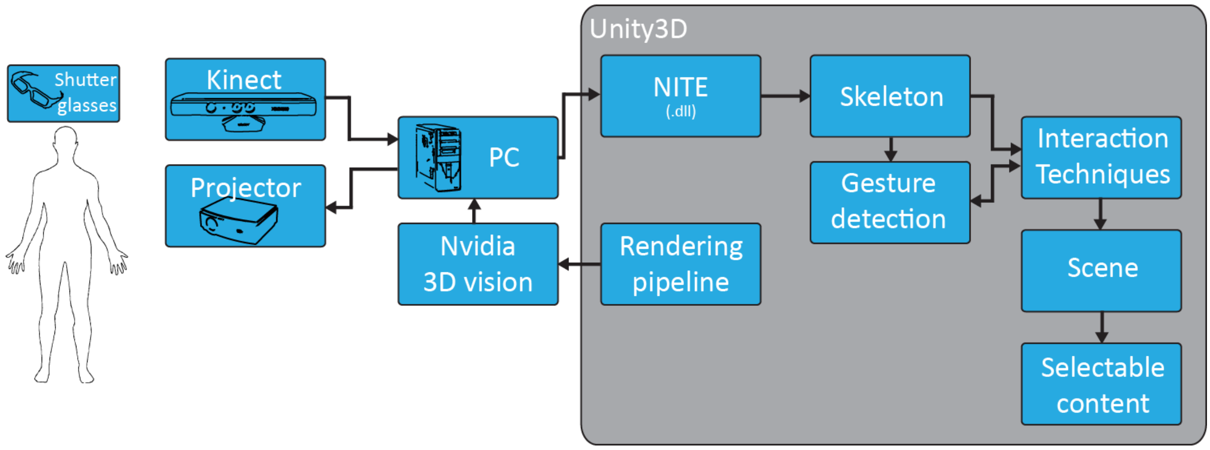


Figure 28 - The overall structure of the system

In the following sections the important parts of this system are described and discussed.

19 OpenNI and NITE (Input)

The input to our system happens through the OpenNI framework (OpenNI, 2011). This framework is a standard interface for 3D sensor data and it is open sourced. The purpose of this framework is to define data types like depth maps, color maps and a module that generates them from the hardware, for 3rd party developers. The NITE framework (PrimeSense, 2011), is a mid-

[1] github.com/avin2/SensorKinect

dleware component utilizing the data from the OpenNI framework in order to output tracked positions of the user. NITE also provides gesture detection, which however was not used in our project, since we created our own.

The implementation in Unity3D is based on a wrapper provided by OpenNI. This wrapper[1] has been written by Shlomo Zippel. For the purpose of this project a range of changes has been applied and functionality added to the wrapper.

The starting point for this wrapper is a static implementation of the external C functions provided by NITE, here is an example of one important function (in C#):

```
[DllImport("UnityInterface.dll")]
public static extern bool GetJointTransformation(uint userID, SkeletonJoint
joint, ref SkeletonJointTransformation pTransformation);
```

The wrapper handles in this way all functionality with the Kinect: it starts the camera, shuts it down, displays the depth map and most importantly tracks the user. We structured the functionality of the wrapper further by adding a general skeleton class (MasterUser), which holds all the tracked data points and added access functionality to this class.

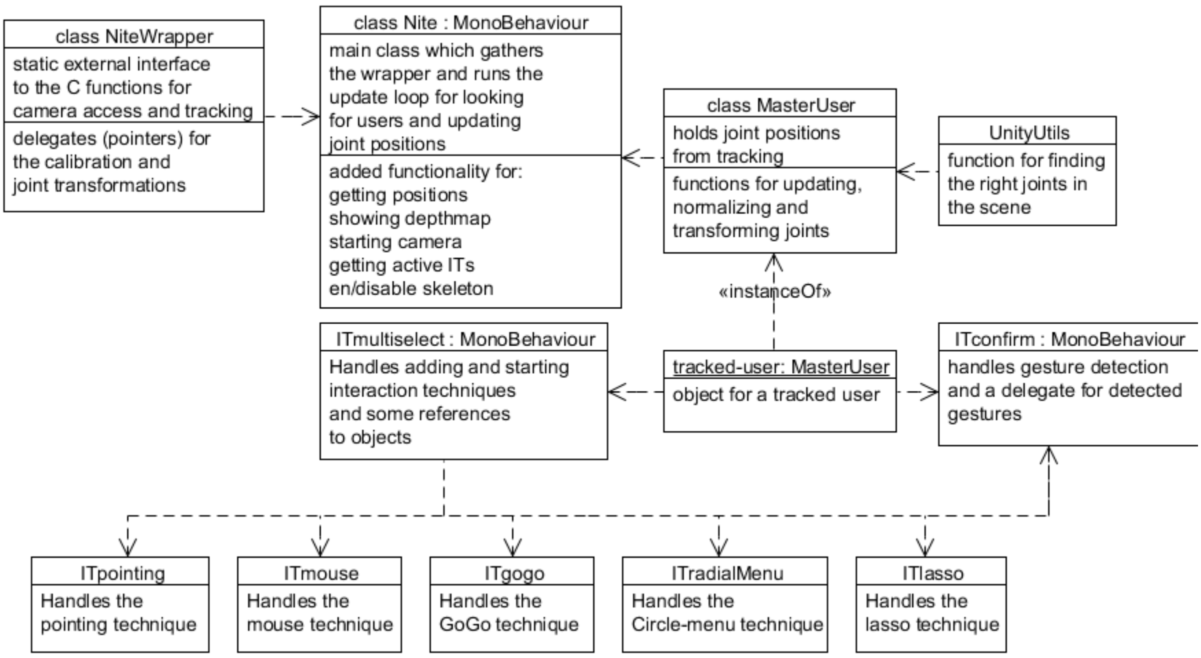


Figure 29 - Overview of the most important classes and their dependencies

Overall the functionality is laid out in the way that when the camera is running it starts looking for users and when it finds a user calibration can be started. The calibration requires the user to hold a “psi-pose” (Figure 30) for around 3seconds, after which the user is calibrated and tracked.

With a calibrated user, tracking is running, which updates all joint positions on every frame. From here the different ITs can be added in order to allow interaction with the scene content.

[1] https://github.com/OpenNI/UnityWrapper

Figure 30 - PSI-pose in order to start calibration

Since Unity3D has the feature of prefabs, which basically are reusable game objects, we added a skeleton prefab for visual representation and easy access to all joint transformations. This prefab is instantiated for every tracked user and also servers for camera control and for managing the different ITs (Figure 31).

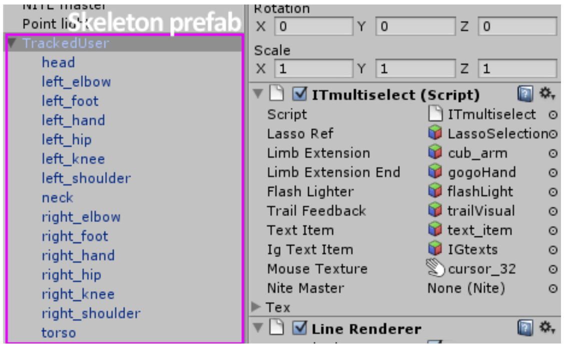


Figure 31 - Skeleton prefab inside Unity3D

Each tracked limb has a Transform object in this prefab; the transform is used in Unity3D to hold position and rotation information. These positions and rotations are updated from the Nite class, within the MasterUser object. The available joints, tracked by the external NITE functions, are: *head, neck, torso, left and right - elbow, shoulder, hip* and *knee*. The hand and feet positions are not directly tracked and need to be calculated as an extension of the respective elbow and knee joints. The position and rotation data is updated for every tracked (object of MasterUser) user inside the MasterUser class. Here the function UpdateAvatar is responsible for updating all joint (or bone) information. Each joint is updated in the function TransformBone, which gets the original transformation from the Nite wrapper and normalizes the data.

```
void TransformBone(uint userId, NiteWrapper.SkeletonJoint joint, Transform dest)
{
         NiteWrapper.SkeletonJointTransformation trans = new NiteWrap-
         per.SkeletonJointTransformation();
         NiteWrapper.GetJointTransformation(userId, joint, ref trans);
```

Since Unity3D handles the z-coordinates opposite from NITE the rotation information is rotated and stored as a Quaternion.

```
Vector3 worldZVec = new Vector3(-trans.ori.m02, -trans.ori.m12, trans.ori.m22);
Vector3 worldYVec = new Vector3(trans.ori.m01, trans.ori.m11, -trans.ori.m21);
Quaternion jointRotation = Quaternion.LookRotation(worldZVec, worldYVec);
```

The position information from NITE is represented in real world metrics (cm), where the position xyz (0, 0, 0) represents the position of the Kinect camera. The maximum depth is set to 10m or 1000cm and the data obtained for positions is normalized to Unity3D coordinate space by dividing by 1000. Furthermore we added linear interpolation for both rotation and position information in order to slightly smooth the data between old and new positions and rotations.

```
dest.rotation = Quaternion.Slerp(dest.rotation, newRotation, Time.deltaTime *
20);

dest.position = Vector3.Slerp(dest.position, new Vector3(trans.pos.x/1000,
trans.pos.y/1000-1, -trans.pos.z/1000), Time.deltaTime * 20);
```

In this way all joint transforms are updated directly by setting the position and rotation. Another way of updating these joint positions would be to use a hierarchical structured skeleton which is updated through the rotation information (forward kinematics). This approach however is more inaccurate, as the rotation information obtained from the NITE framework in general is more inaccurate than the position information. Furthermore the data obtained from the Kinect may become unstable in some scenarios where the tracking algorithm has problems with segmentation and identification. These known issues are:

- Arm tracking is less stable when close to other body parts, especially the torso.
- Leg tracking is in general slightly unstable and noisy
- Pose tracking becomes unstable when the head is not visible (e.g. when occluded by the hand)
- Arms and legs in extremely stretched positions might be lost by the tracker
- Very fast motions may cause tracking failure
- In some cases tracking might be bad, where re-calibration helps to resolve the problem

(PrimeSense, 2011)

The accuracy and instability of the tracking is one of the major concerns for the usability of the ITs. We tried to eliminate unstable and noisy tracking whenever possible by smoothing the data. NITE provides functionality for this directly by the following function:

```
[DllImport("UnityInterface.dll")]
public static extern void SetSkeletonSmoothing(double factor);

NiteWrapper.SetSkeletonSmoothing(0.6);
```

The default smoothing is 0.6, increasing this to a maximum of 1 removes a lot of noise on the cost of latency. In our case we used an increased smoothing on the pointing and the GoGo technique, since these use rotation and position information and therefore are especially prone to errors. Increasing the smoothing for all ITs showed however to be too expensive on the latency side, where the normal smoothing has approximately ~200ms response time (this depends on our implementation) the increased smoothing may increase the response time to around 1second. Since we wanted the system to be as responsive as possible we settled on the default smoothing in most cases, which in turn may be slightly noisy and could possibly cause errors in the accuracy of the selection techniques.

20 Gesture detection

Detecting gestures is one of the very important features of the implementation. The gestures serve as confirmation methods, since no buttons are used. The problem with detecting gestures in continuous input is the differentiation between normal movement and gesture. In that way the implementation should make sure that only deliberately by the user executed gestures are detected and normal hand and arm movements are not. In our implementation we used a simple approach for detecting the push, pull and swipe gesture, which is based on the velocity. Though there might be possible better approaches, the velocity based approach still showed to be adequate for being used in our prototype.

For the push, pull and swipe gestures the hand position is used as input. The hand position comes as a 3D point and for the first steps of our simple gesture detection the velocity of it is calculated as: $\overline{v} = \frac{\Delta x}{\Delta t}$. :

```
private void calcVelocity()
{
        lastTime = currentTime;
        currentTime = Time.time;
        float dt = currentTime - lastTime; //delta time

        if(dt > 0)
        {
                float dx_r = lastPosition_r.x - rightHand.position.x;
                float dy_r = lastPosition_r.y - rightHand.position.y;
                float dz_r = lastPosition_r.z - rightHand.position.z;
                velocity_r.x = dx_r / dt; // delta x/delta time = velocity x
                velocity_r.y = dy_r / dt; // - || -
                velocity_r.z = dz_r / dt; // - || -
```

From the velocity thresholds are used to further evaluate gestures. Furthermore we included a few assumptions which drastically improve the gesture detection. The first assumption is that a users hand is raised above their hips. The second assumption is that after performing a gesture it will take some time (~1sec. or more) to execute another gesture. In this way for e.g. the push gesture the z-velocity (depth velocity) is used, when over a threshold and the hand is raised above the hips the gesture is detected and further gesture detection is prevented for around one second. Another assumption included to prevent gesture detection errors was that gestures are only performed when not moving; this was implemented by checking the velocity of the root transform of the skeleton (which is the torso).

```
string gestCandidate = "none";
if(checkVel.x > checkVel.y && checkVel.x > checkVel.z)
{
        //this would be a swipe (left right)
        if(velocity_r.x < -swipeVelocityThresh)
                gestCandidate = "swipe_right";
        if(velocity_r.x > swipeVelocityThresh)
                gestCandidate = "swipe_left";
}
```

Other simple gestures used in the implementation were raised arm triggers, which simply are detected by a distance threshold (e.g. hand position in relation to hips).

```
private bool bothHandsRaised()
{
	if(Mathf.Abs(rightHand.position.y - rHip.position.y) > 0.6f &&
	Mathf.Abs(leftHand.position.y - lHip.position.y) > 0.6f)
	{
		if(!bothArmsUp)
		{
			bothArmsUp = true;
			return true;
		}else{
			return false;
		}
	}else{
		bothArmsUp = false;
		return false;
	}
}
```

In the implementation the different ITs like mouse, pointing etc. communicate with the gesture detection by an event system using delegates. In this way the gesture detection will send out a pull event, when a pull gesture was detected, which then is received in the particular IT implementations.

```
public delegate void ChangedEventHandler(object sender, ITeventArgs ie);

//called when a gesture was detected
itArgs.hasChanged = true;
OnChanged(itArgs);

//EVENT HANDLER
protected virtual void OnChanged(ITeventArgs ie)
{
	if(Changed != null)
		Changed(this, ie);
}
```

21 Selection techniques

In the following the implementation for the different selection techniques as explained in the design chapter is presented. The implementation of these techniques concern two aspects, one is the indication of objects and the other is the interaction with scene content. The confirmation of selection is also covered, but uses in all cases the described gesture detection implementation.

All techniques were structured in a way that it is possible to change the input limbs. By this it is possible to assign the techniques to a specific tracked body part, as e.g. the left or the right hand, but also to the head etc. This was used for the easy creation of a left and a right handed version of all techniques in order to support the different handedness of the users.

21.1 Pointing technique

The final implementation of the pointing technique used the arm rotation and hand position for ray casting into the scene. The confirmation method used was timed confirmation. By this the user indicates scene objects by pointing at them with his hand/arm and selects these objects by waiting 1 second. Visual feedback was added for the ray and the time needed waiting in order to confirm.

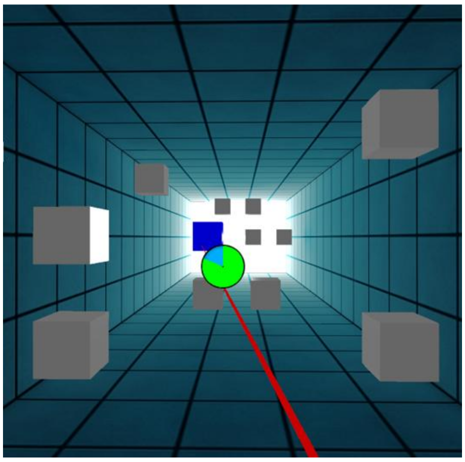


Figure 32 - Implemented pointing technique as seen from user perspective

Indication of objects

For the pointing technique a simple ray cast is used as the starting point. This ray is sent out from the hand position in its forward direction. Unity3D allows easy integration of such functionality, where a ray cast simply can be checked with the content it intersects.

```
public void pointing()
{
        RaycastHit hit = new RaycastHit();//handled by Unity's physics engine
        Ray pointRay = new Ray();
        pointRay.origin = limb.position;
        pointRay.direction = itConfirm.getDirection(limb);
        if (Physics.Raycast(pointRay, out hit, Mathf.Infinity))
        {
```

When the ray intersects with selectable objects in the scene a timer is started. This timer updates as long as the ray intersects the same object and resets when a new or no object is indicated. In this way the user needs to indicate the object for a certain amount of time (1 second) in order to select it.

Feedback

A simple red line was used to represent the ray (Figure 32) originating from the virtual hand position in the pointing direction. The hand position was visually represented by a magenta colored sphere. Visual feedback for the remaining time until confirmation was implemented by an animated tiled texture (Figure 33).

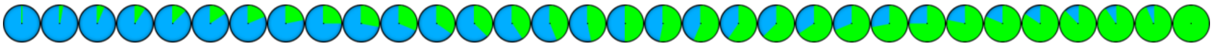


Figure 33 - Tiled texture of the time till confirmation feedback

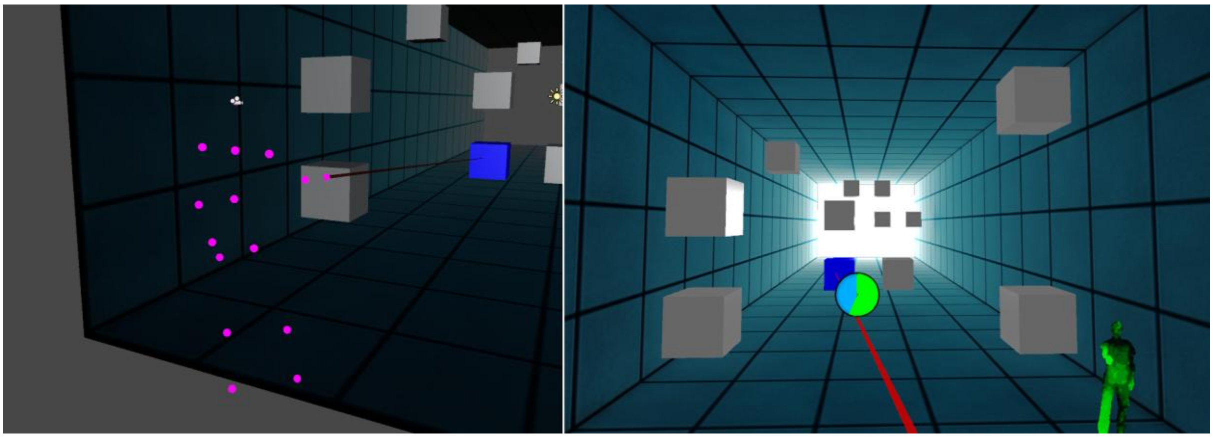


Figure 34 - Scene view of pointing technique with camera perspective (depth map overlaid)

Discussion

The pointing technique is simple and easy to implement. The limitations of this technique are that it allows only selecting one object at a time, and with the chosen confirmation method the user needs to wait for each confirmation. By this it can be argued that the efficiency of this technique is not the best. However since it only requires the user to point at objects, it should be easy to learn and requires no prior knowledge. With this it can be argued that it is an optimal technique for beginners.

Throughout the implementations different iterations we tried around with different smoothing settings for this technique as well as different confirmation methods. Since the technique relies on the rotation of the arm optimal results were achieved when users stretched the arm out. Further accuracy was provided by an increased smoothing. Altogether the implementation still showed accuracy problems for selecting very small screen objects, which are problematic due to the general noisy signal from tracking.

21.2 Mouse technique

The final implementation of the mouse technique mapped the users hand position to the screen by transforming the tracked hands position to screen space. From this position intersection with scene objects was checked by ray cast, similar to the pointing technique. The confirmation method used in this implementation was the push gesture. Visual feedback to the position of the users hand was added in form of a hand cursor. Further functionality to the mouse technique was a changeable C:D ratio, which however was not activated in the final evaluation, since it showed to confuse users.

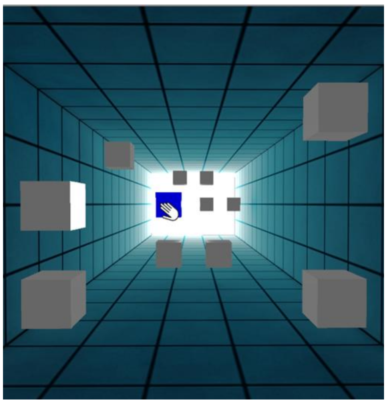


Figure 35 - Implemented mouse technique as seen from user perspective

Indication of objects

As the starting point of the mouse technique the users hand position is translated to screen space. The obtained position is then mapped through a custom function (worldToMouseCursor), which modifies the C:D ratio and keeps the hand position inside the screen space even when moved outside. In this way when a user moves his hand outside the mapped area the cursor will still remain on the screen, just like the mouse cursors in e.g. Windows, which always will stay at a screen side.

```
private void mousePoint()
{
        RaycastHit hit = new RaycastHit();
        //taking the limb to screen space
        actual_sp = Camera.main.WorldToScreenPoint(limb.position);
        cursor_sp = worldToMouseCursor(actual_sp);
```

From the transformed position a ray cast determines the intersection with scene content, which allows users to select objects when placing the mouse cursor over one such object.

```
private Vector3 worldToMouseCursor(Vector3 input)
{
        Vector3 temp = new Vector3(input.x * cd_ratio, input.y * cd_ratio, 0.5f);
        //actual position larger as width (flipped y)
        if(temp.x > screen_width)
               temp.x = screen_width;
        if(temp.x < 0)
               temp.x = 0;
        if(temp.y > screen_height)
               temp.y = screen_height;
        if(temp.y < 0)
               temp.y = 0;
        return temp;
}
```

The cursors position is displayed directly at the screen position. Figure 36 shows a visual representation of how this mapping is implemented.

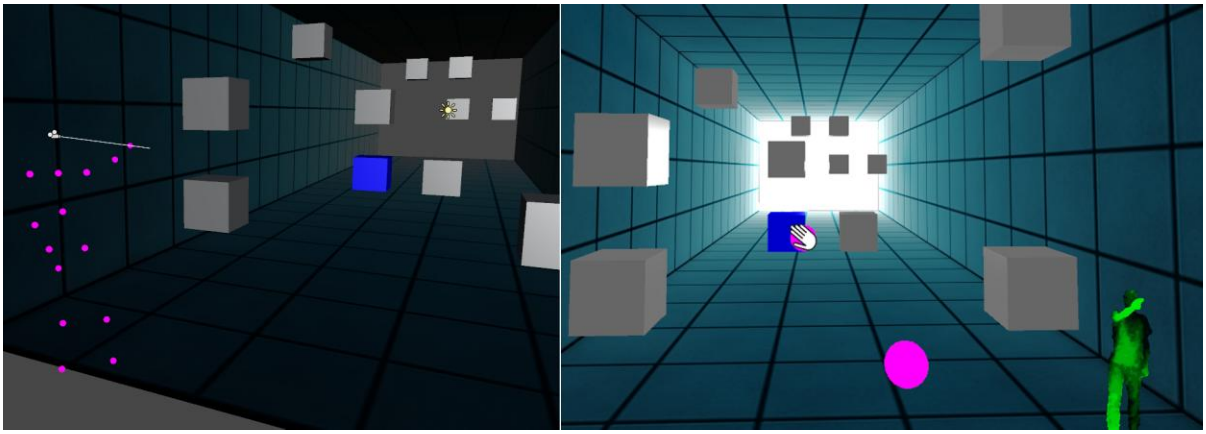


Figure 36 - Scene view of the mouse technique

Feedback

As explained above a cursor was used to indicate the users hand position on screen. This cursor was designed as a hand (Figure 37), in order to allow easier identification of which body part is used for this technique. Further feedback was not added, although we discussed adding feedback for the push confirmation gesture in form of a changed cursor.

Figure 37 - Hand cursor

Discussion

The intention with the mouse technique was to resemble the behavior of a mouse as used in most operating systems. The limitations of the technique are similar to the pointing technique that it only allows to select single objects. However the possibility of adding additional functionality exists to extend this technique also to select multiple objects. This could e.g. be done by adding a selection square, like used in most operating systems when pressing and dragging the mouse.

For the C:D ratio functionality we tried out different methods of changing this dynamically on runtime, throughout the implementations iterations. Here we added swipe gesture to increase and decrease this ratio. This however showed to confuse users and was not practical. Still the C:D ratio functionality may be very interesting especially if a high accuracy is required, as e.g. in the case of selecting small screen objects.

Problems with tracking inaccuracy in this technique are found when the user holds his/her hand in front of the head. Also since we did not add further smoothing to the technique the selection of small objects may become problematic. The chosen confirmation method of pushing further adds to problems when selecting small objects, since it requires the user to move the same hand which is used for indicating the object. It was tried to avoid this problem by setting the threshold for the push gesture to a low value, so that it only required the user to tip the hand slightly forward in order to confirm his/her selection.

21.3 GoGo technique

The GoGo technique builds on the concept of extending and contracting the virtual arm. Objects are selected by directly touching them with the tip of this virtual arm. The arm is contracted and extended by using the pull and push gestures. The virtual arm orientation uses similar to the pointing technique, the users arm rotation. By this the arm extends from the users real arm position to the virtual representation on screen. This virtual representation was displayed by a simple box with a sphere on the tip.

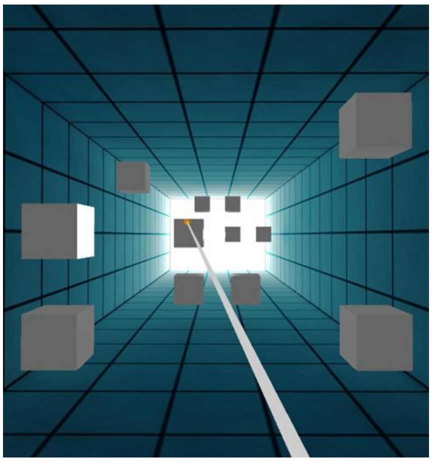


Figure 38 - Implemented GoGo technique as seen from user perspective

Indication of objects

Since the objects are selected directly on touch this technique uses an automatic confirmation action. The method of pointing and extending / contracting the arm is explained in this subsection.

The virtual arm is handled by scaling the representation, which as explained was displayed as a box. This was implemented by mapping the orientation of the virtual arm to the elbow limb. The length of this arm was set by the user. The plan as explained in the design chapter was to add a non-linear function to change the length of this arm. This was however changed in the final implementation, as a pre-evaluation showed that users had severe difficulties with getting the right length of the arm and by this made the selection of objects cumbersome. The non-linear function was therefore replaced to set the length of the arm to three pre-set levels of depth, matching the three depth levels of selectable objects used in the final scene (which will be explained at a later point). Although this changed the intended design, we still think that the implementation represents the overall concept of extending and contracting a virtual arm. Furthermore this concept was highlighted by visually animating (interpolating) the length of the arm when the user performed a pull or a push gesture.

```
private void gogoIT()
{
        Ray pointRay = new Ray();
        pointRay.origin = main_limb.position;
        pointRay.direction = itConfirm.getDirection(main_limb);
        //target distance
        if(cur_index_distance == 1)
                target_distance = dist1;
        if(cur_index_distance == 2)
                target_distance = dist2;
        if(cur_index_distance == 3)
                target_distance = dist3;
        //Lerping the extendend arm and taking base rotation
        current_distance = Mathf.Lerp(current_distance, target_distance,
        Time.deltaTime * transition_speed);
        ext_limb.transform.localScale = new Vector3(current_distance, 0.03f,
        0.03f);
        ext_limb.transform.position = main_limb.position + (itConfirm.getDirection
            (main_limb)* (current_distance/2));
        ext_limb.transform.rotation = main_limb.rotation * initial_rotation;
        end_limb.transform.position = main_limb.position + (itConfirm
            .getDirection(main_limb) * current_distance);
}
```

The selection of objects in this technique is determined by checking an intersection of the tip of the arm (represented by the sphere) with the scene content. This is in Unity3D handled by the physics engine and straight forward to implement.

```
void OnTriggerEnter(Collider other) {
        if(other.tag == "Player")
                selected = true;
}
```

Feedback

The virtual arm was as explained above represented by a simple box with a sphere on top. This simple representation showed to be enough for understanding the technique. However, it can be argued that a more advanced representation visually resembling a real arm may add to the ease of understanding this technique. Another especially important factor for this technique showed to be the ability to perceive the arms length, which was possible through the use of stereoscopic rendering. Without this the technique becomes almost impossible to use as it is crucial to see how long the arm is.

Discussion

The GoGo technique was changed from its original design; the changes should though result in an easier handling of the technique compared to the original design. Since we set the distance to the three different corresponding distances of the scene objects it must be argued how this distance could be implemented in a dynamic situation, where distances to objects are not known before hand. For this a ray cast could be used which finds the objects intersecting with the pointing direction. From this the distance could be stored dynamically, where the user navigates to the found distance by the push gesture if the distance extends the current distance or by the pull gesture if it is before the current distance.

The GoGo technique has the same inaccuracies as the pointing technique, since it depends on the rotation. Again the smoothing was increased in order to improve the accuracy. Selecting small objects may still require very fine changes of the users arm, possibly resulting in difficulties.

The efficiency of the GoGo technique can be argued. Since the technique requires changing the length of the arm and afterwards touching the objects there may be situations where this technique is unpractical. One example may be if many scene objects are place right to each other; in order to touch an object surrounded by others the user must first navigate the arm through one of the surrounding objects, which may cause him to select an unwanted object. In this case the length of the arm must be changed to a different than the object and then extended to hit the surrounded object, which in turn results in much more actions required to select a single object than most other techniques. Such situation may be more effective when using a non-linear function for changing the arms length mapped to the hands position.

21.4 Circle-menu technique

The circle-menu technique is based on an indirect selection of objects through the use of a menu item. This menu is displayed as an overlay on top of the scene (Figure 39). Objects are indicated by moving the hand around a fixed point in a circular motion. The confirmation method used for this technique was a push gesture. The menu could be opened and closed by a right and a left swipe gesture. Overall the implementation of this technique was far more complex then the above described techniques, also since we chose to implement a dynamic menu which can be adapted to the number and category of selectable objects contained in the scene.

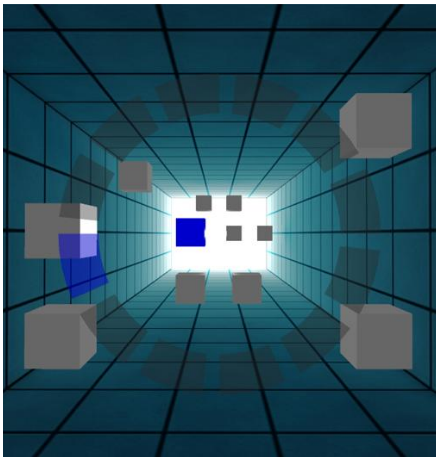


Figure 39 - Implemented circle-menu technique as seen from user perspective

Indication of objects

For the indication of objects the angle between the users hand position relative to a fixed point (the users neck) was used. From this angle the corresponding menu item on the circular menu was found.

```
private float getAngle()
{
        float dx = limb1.position.x - initialHandPosi.x;
        float dy = limb1.position.y - initialHandPosi.y;
        float angle = Mathf.Atan2(dy, dx) * Mathf.Rad2Deg;
        if(angle < 0)
                return 360+angle;
        return angle;
}
```

The intended design for this circle-menu was that different levels of categories could be represented. By this users should be able to select multiple objects on the main category (which we called "base" in the implementation) and single objects by navigating inside the base items. This was however changed during iteration of the implementation, since only one category of selectable objects was used in the final implementation. However, the implementation still allow this functionality which in the final implementation was switched off, resulting in only one level of selectable objects (as seen in Figure 39).

The selectable objects can in this way be indicated by placing the hand in the direction of a menu item. The menus origin or center point was placed relative to the users' neck, so that selecting an item in this menu, matched the real hand position relative to the neck. Furthermore the menu items indicated in the menu resulted in an indication of the objects in the actual scene (colored blue in as seen in Figure 39). These menu items or sub-categories were mapped to the actual scene objects in a circular fashion, so that when e.g. indicating a menu item to the right of the center the actual scene object to the right was also indicated.

The different categories of items are stored in arrays, which can be changed to match the actual amount of objects in the scene. This dynamic implementation added to the complexity of the implementation, since the visual representation of the menu needed to reflect this changeable array. By this the implementation of the visuals for the circle menu were based on a dynamic mesh generation, from code, adapting to the amount of scene objects defined.

The dynamic generation of the menu was implemented in the way that the each object contained in the scene was represented by a pie of the circle, so that when e.g. having 10 objects the circle was divided into 10 categories. This menu was initialized through the code found in the Appendix section IV.a). The functions in this code calculate each start and end angle of each pie, where one pie represents a selectable object (Figure 40). The dynamic mesh generation happens in another class (object pieMesh in the code) through the function drawPie as shown on the next page:

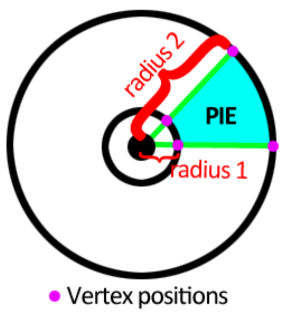


Figure 40 - Calculation of one pie representing a game object

```
private Vector3[] calcVerts()
{
	//each step has 2 verts - stepsize for each degree
	int vertexCount = (sliceTo - sliceFrom) * 2;

	vertexCount = vertexCount / stepSize;
	if(vertexCount%2 != 0)
		vertexCount += 1;

	Vector3[] verts = new Vector3[vertexCount];
	//x = center.x + radius * cos(angle)
	//y = center.y + radius * sin(angle)
	//vertex counter
	int j = 0;
	for(int i=sliceFrom; i<sliceTo; i += stepSize)
	{
		float theta = Mathf.Deg2Rad * i;
		//Vertex1 (inner vertex)
		if(j < vertexCount)
		{
			verts[j].x = center.x + radius1 * Mathf.Cos(theta);
			verts[j].y = center.y + radius1 * Mathf.Sin(theta);
			verts[j].z = center.z; //the depth without offset
		}
		//Vertex2 (outer vertex)
		if(j+1 < vertexCount)
		{
			verts[j+1].x = center.x + radius2 * Mathf.Cos(theta);
			verts[j+1].y = center.y + radius2 * Mathf.Sin(theta);
			verts[j+1].z = center.z; //the depth without offset
		}
		//increment counter
		j += 2;
	}
	return verts;
}
```

The mesh created by these functions can be seen in the figure below (left).

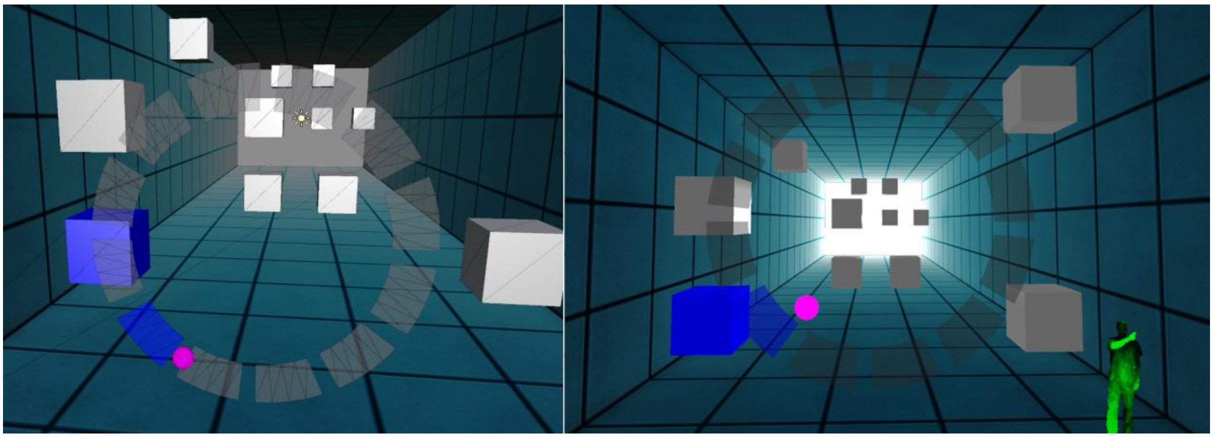


Figure 41 - Scene view of the circle-menu technique

Feedback

The circle-menu was displayed on top of the scene in a semi-transparent fashion. By this users were still able to look at the scene content while navigating the menu. The menu-items were colored by gray color when not indicated or selected. Indicated items were colored blue, along with the corresponding object in the scene. Selected items were colored green on the menu with the corresponding scene items. This color coding was used throughout the different techniques as also described in the test implementation - section 22.

Discussion

One of the major benefits of this technique is the radial layout of the menu. Through this layout the distance to select different objects are kept at the minimal distance possible, which in turn should make the technique very effective for selecting objects.

The accuracy of the technique does only depend on the hand and neck position, which makes it more stable than the rotation information used in the pointing technique. We experimented with different amount of menu items, where adding many items (20+) makes it more difficult to select one single item. With smaller amounts the technique should be easy to use.

21.5 Lasso technique

The final implementation of the lasso technique records the users hand motion when activated, and selects the content inside this drawn selection when confirmed. For both activation and confirmation of this recording the push gesture was used. A green line indicates the recorded hand motion, and the objects within this line are selected upon confirmation (Figure 42). The implementation of this technique was especially difficult since it required a method of selecting the boxes inside the drawn selection.

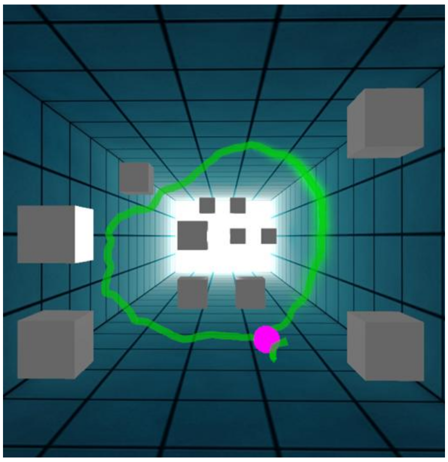


Figure 42 - Implemented lasso technique as seen from user perspective

Indication of objects

As a starting point the lasso technique records the users hand position when started and stores it in a dynamic array.

```
//start tricker
if(activationTrigger)
{
        //record positions
        trail.enabled = true;
        Vector3 curPos = lassoLimb1.position;
        lassoPath.Add(curPos);
        lassoRecording = true;
        itConfirm.drawLine(lassoPath.ToArray());
//end tricker
```

While drawing the selection, feedback is shown in form of a line rendering of the whole array. When the user finishes his selection drawing by confirming with another push gesture the content within this drawn path needs to be selected. This was however not a trivial problem, since the recorded path needs to be transformed through the cameras viewing matrix and objects within the volume defined by the actual recorded path and the projected coordinates need to be found. Since Unity3D has an easy accessible collider intersection implementation based on the physics engine, the workaround for this problem utilized in our implementation was to generate a mesh from the above described path and to use this mesh to find colliding objects, as seen in Figure 43. This mesh was send in the direction of the drawn indication by a constant force after confirming, which added a magical inspired visual feedback to the lasso tool.

```
for(int i=0; i < selectedPath.Length; i++)
{
        Vector3 screenPoint = Camera.main.WorldToViewportPoint(selectedPath[i]);
        Ray camRay = Camera.main.ViewportPointToRay(screenPoint);
        worldPath[i] = camRay.GetPoint(30.0f); //30 units depth along ray
        centerOfMassPro += worldPath[i];
}
centerOfMassPro /= selectedPath.Length;
//generate the mesh vertices
Vector3[] verts = generateVertices(selectedPath, worldPath);
int[] triangles = calcTris(verts);
```

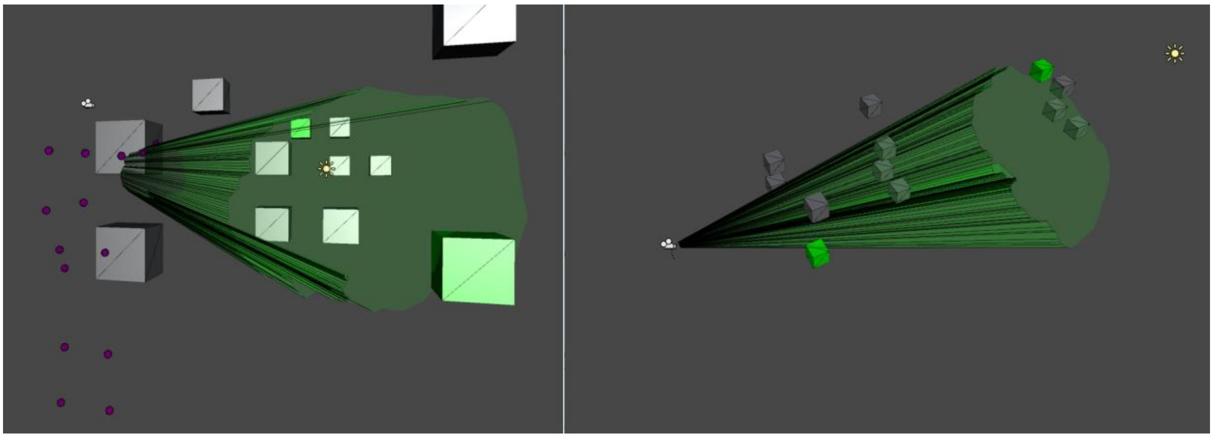


Figure 43 - Scene view of the mesh created for selection from the lasso technique

Feedback

The feedback of the lasso technique was shown in form of green semi transparent line while drawing. Additional feedback happened upon confirmation, which showed the selection mechanism also as a semi transparent green mesh (Figure 43), as also described above. Additional feedback activated was a cyan colored sphere which indicated the users hand position.

Discussion

The intended use of the lasso technique is for the selection of multiple objects, although it also can be used for the selection of single objects. Using it for selecting single objects may however be ineffective since the technique requires an action for start and end as well as drawing of the selection.

The accuracy of the technique is slightly problematic since the recorded hand position depends on the tracking fidelity, which may at times make it hard to draw an accurate selection due to tracking errors. Possible enhancements to this technique may therefore be to constrain the free hand drawing to for example the positioning of line segments (as explained in the design chapter).

22 Test implementation

For the final evaluation of the interaction techniques a test environment was implemented. This consisted of an introduction to the test, with explanations of the tasks, a tutorial stage and a task stage for every IT. For both the tutorial stage and the task stage the same VE scene was used. In the following a brief explanation of the used VE is giving followed by an explanation of the test bed.

22.1 Virtual environment

As explained in the design chapter the layout of the VE had the purpose of the supporting the selection task by enhancing the depth perception of the scene. The VE consisted of 2 parts, a static room or hallway, and twelve boxes serving as the items for selection.

The room uses baked textures, which were made in 3DsMax[1], in order to pertain higher detailed shading and soft shadow details than achievable in a real time engine. This was also due to the fact that the free version of Unity3D was used for implementation, which does not support real time shadows.

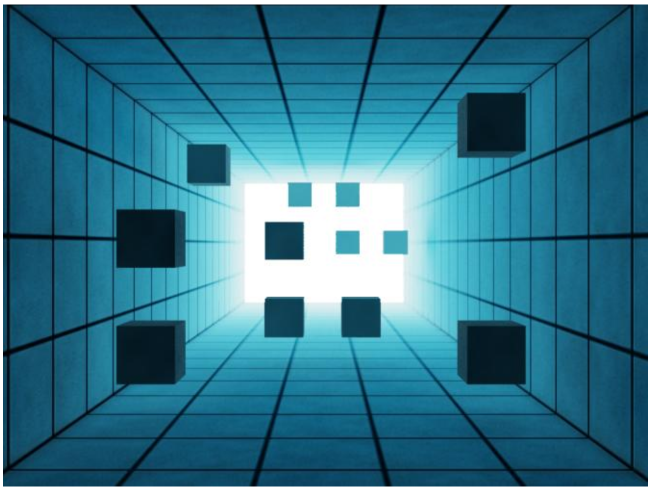


Figure 44 - Rendering of the VE test scene

The boxes were placed as seen in Figure 44, with equal spacing in between from an XY perspective but with three different levels of depth. All boxes were colored gray, when no interaction took place. In interaction with the different ITs, the coloring of these boxes was changed, as explained in the next section.

22.2 Test instructions, randomization and data logging

The introductions to the test were presented to the user as a graphical user interface (GUI) element in the form of simple textures containing explanations texts together with simple iconic

[1] http://usa.autodesk.com/3ds-max/

pictures. The first part introduced the user to what is happening during the test and what he/she needs to do, where the later explanations explained how to use each IT. The introductions are included in the appendix chapter. Furthermore the introductions given were divided among a range of slides, which could be navigated by raising the left, right and both hands. Here we used a simple mapping for the navigation of the different explanation boxes, raising the left hand would take the user to the last explanation box, raising the right hand would take the user to the next explanation box and raising both hands was used when the user did finish reading the explanation texts. These actions were explained to the user, and small icons indicated them on the explanation boxes (as seen in Figure 45).

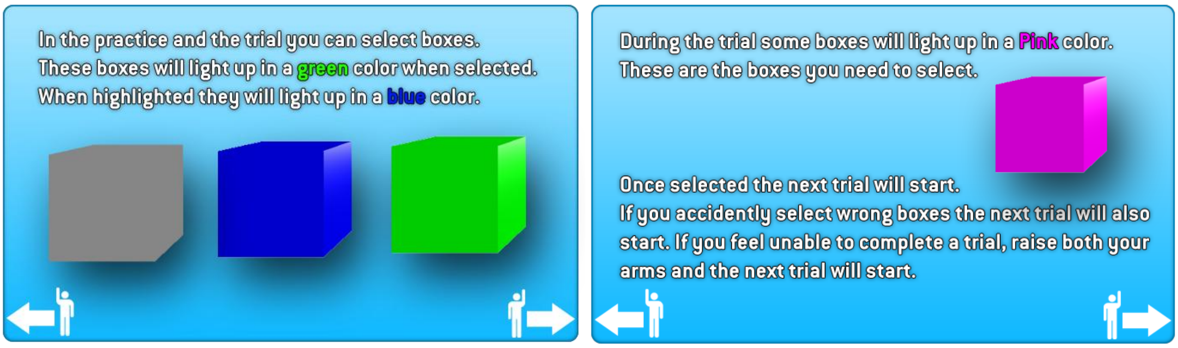


Figure 45 - Two of the explanation boxes as used in the test implementation

As mentioned we used different colored boxes to give feedback on the selection to the user. The colors used were a blue color when indicating a box, a green color for a selected (confirmed) box, and the pink color we used to show which boxes the user needs to select during the tasks. These colors can also be seen in the figure above.

For the evaluation the order of techniques and the order of tasks were randomized through code. 6 different tasks were given in random order to the user, repeated for each technique. More details on this randomized design are included in the test chapter.

Another requirement for the implementation was that all data is logged. For this purpose a file handling system was implemented, which stored the random order of techniques, and the random order for the tasks into a text file. In this text log file the completion times and the selection errors were also recorded. An example showing a fragment of such a log file can be found in the appendix at IV.b).

After the test all text log files were gathered into an excel xml file by a small program we wrote for this purpose. This program reads all data entries from the log files, sorts the data into correct order and stores it into the excel file. The complete code and the program are included on the CD accompanying this report (It is written in C# as a WPF application).

23 Stereoscopic rendering

The stereoscopic rendering used for displaying the implementation was based on Nvidias 3D Vision (Nvidia, 2011). We used a single wall-projection for displaying the VE together with Nvidia 3D Vision shutter glasses. These glasses are synchronized with the display using an infrared transmitter, which was mounted on top of the projection screen. The advantage of using this technique was that no manual rendering of two images had to be programmed. All of the stereoscopic rendering parts are handles by the soft and hardware. The disadvantage of this technique is that it is limited to Nvidia GPUs / Geforce graphic cards.

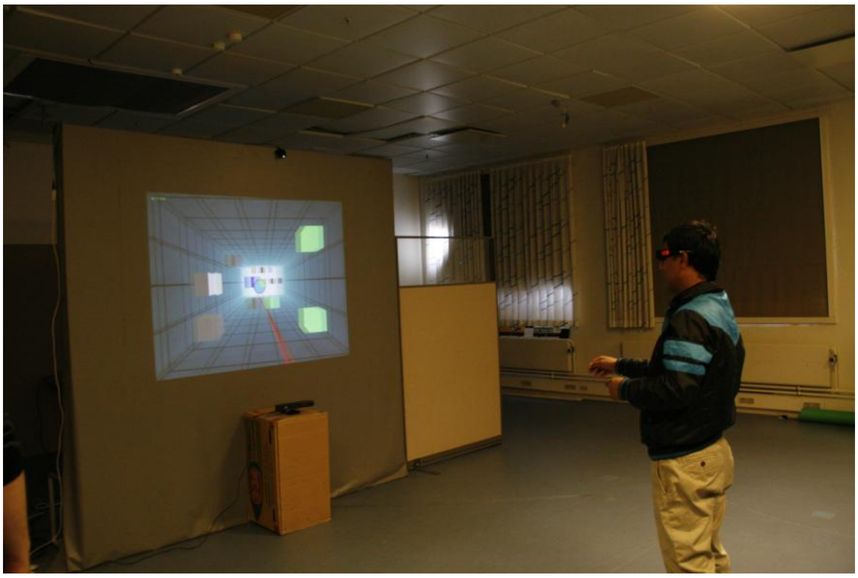


Figure 46 - Final setup of implementation with back projection and stereoscopic rendering

The depth was manually adjusted to match the camera position best. The figure above shows the overall setup, using back projection. The projected image shows how the stereoscopic rendering looks like when not wearing shutter glasses.

24 Discussion

The implementation of the selection techniques utilized the OpenNI and NITE framework. Unity3D game engine then optimized the tracker data towards a creation of selection techniques, and finally rendered a stereoscopic VE.

The use of these tools had an influence on the final prototypes of the five selection techniques. First the choice of the gesture detection algorithm, using velocity as a mean of defining swipe, push and pull actions, had the known problem of differencing "normal" movements from gestures, possibly resulting in a higher amount of errors than intended.

Furthermore the OpenNI/NITE framework has known weaknesses, e.g. when the subject is having his arms close to the body the tracking can be lost; this had an influence on the implementation of the different techniques, but was considered acceptable.

Another known issue when using continues signal input is noise in the signal. The NITE framework offered noise reduction, the price was latency. This issue was carefully investigated and an acceptable compromise “precision vs. response time” solution was tailored to each technique.

We are expecting that the different techniques will have advantages and disadvantages:

The pointing technique may be easy to learn, precise, but is probably not effective in multiple selection tasks.

The mouse technique may have the same limitations as the pointing technique. Furthermore having a push confirmation in order to select a box can result in problems, since it requires subjects to move their hands when confirming the selection and thus the indication can be lost. The solution that was implemented is a push threshold set to a minimum, to allow users to perform a minimal action to confirm the selection. Another drawback is that the mapping in some situations resulted in that the user is occluding the view with his/her hand, blocking the visibility of the projection.

The circle-menus radial design possibly makes it an effective selection tool when having a small amount of objects, but when the amount of objects rise it becomes more difficult to select the objects, due to spacing of the menu elements.

The lasso technique is possibly effective to make multiple selections, but may not be effective to make single selection, due to the actions that are needed to perform a selection.

The GoGo technique has three pre-defined lengths of the arm, corresponding to the positions of the boxes. The techniques design may provide an advantage of being easy to understand through the object touch confirmation. The drawback can be that the users are selecting boxes by mistake when navigating the arm.

Taking all consideration into account we consider the implementation of the selection techniques followed the intended design and therefore are ready to be evaluated on the intended target group.

PART V - Test

The iterative procedure used in the work of this thesis was evaluated through different tests. The first test was a heuristic evaluation of possible concepts for selection techniques constructed from our theoretical model, as explained in the methodology section 9.3. The second test was a formative usability evaluation, which had the goal of identifying potential weaknesses and flaws in the selection techniques design and implementation. The experience and results from the two first tests build the basis for the procedure and methodology of the final test. Detailed description of the first two pre-evaluation can be found in the appendix in section I and II.

The final test, described in this chapter, was a summative evaluation. Its goal was to answer the stated problem, by conducting a comparative evaluation of all implemented selection techniques, investigating which of these techniques are considered best suited for single and multiple selections.

25 Methodology

25.1 Goal

The goal of the final evaluation was to answer the proposed problem statement: "Which body controlled interaction techniques are best suited for single and multiple selection of screen content, using optical markerless motion capture (Kinect and NITE) and which design elements constitute the success of a technique?". In order to answer this statement, a hypothesis for the ideal selection technique for single and multiple selections of objects is required. The elements that will constitute a best-suited selection technique are based on Jacob Nielsen's definition of usability (Nielsen, 1993), these are: *Learnability, efficiency, memorability, errors* and *satisfaction*. Additionally a comfort rating was added to the satisfaction, as also described in the pre-analysis section 8.2.

Constructed from this definition of usability, the hypothesis for the ideal selection technique in a Virtual Environment is:

The ideal selection technique is the one that can be easiest learned, most effective, easy to remember, make least mistakes with, and being most satisfactory to use.

Recognizing the type of users when measuring the usability is important. The demands to the system may thus be different between novice users or experts in the field. According to Nielsen (1993) the subject of *learnability* often is referred to the novice user of a system; specifically how long time it takes the user to learn to handle the application. *Efficiency* refers to the expert users steady state of performance when the learning curve has flattened out. *Memorability* is directed to the casual user of the system, being able to return to the system and still remember how to control it. *Errors* and *satisfaction* of the system involve all user groups, since these can be important factors for e.g. both experts and novice users.

The intended target group for our implementation was users that are novice to body controlled interactions, since it was our assumption that a future VE would target these users.

Having these variables to define the quality of the system, and a target group, the next step is to measure these variables, an issue that will be addressed in next section.

25.2 Method of measurement

The data collection done in order to get a valid result was through data logging; getting quantitative data by recording the time and error rate each subject had on a technique, and a questionnaire; retrieving quantifiable qualitative data from the users concerning each technique by using likert-type scales.

In order to conduct an accurate evaluation of the techniques, the type of variables that are involved will first be described.

Independent and dependent variables

The independent variables were the five different techniques. The dependent variables were the tasks, meaning that each of the techniques had the same set of tasks, although presented in a random order. Hence the results of how the users did on the tasks would be a reflection of the techniques ability of handling the tasks.

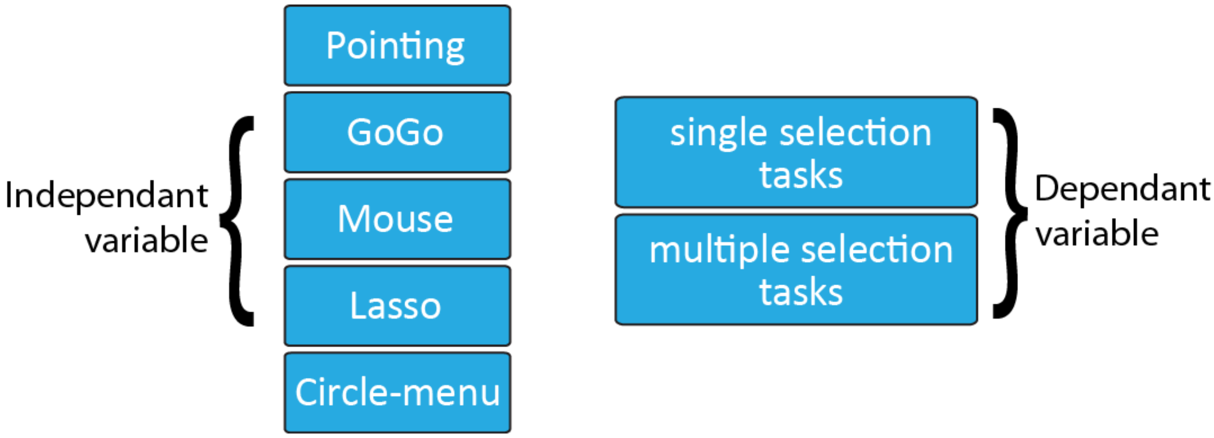


Figure 47 - The type of variables that were involved in the estimation of selection techniques. The techniques were considered as independent variables, the dependent variables were the tasks.

Further details on each of the measurement methods will be described in the following sections.

Data collection

According to Tullis & Albert (2008) the following is needed in order to get valid results from an evaluation: an understanding of the type of data, which metrics to apply and how to apply the correct method of statistical test to the data collected from a test.

Tullis & Albert (2008) defined four types of data involved in usability metrics, having corres-ponding metrics and statistical procedures.

Data Type	Common Metrics	Statistical Procedures
Nominal (categories)	Task success(binary) errors (binary), top-2-box scores	Frequencies, crosstabs, Chi-square
Ordinal (ranks)	Severity ratings, rankings (designs)	Frequencies, crosstabs, chi-square, Wilcoxon rank sum tests, Spear-man ranks correlation
Interval	Likert scale data, SUS scores	All descriptive statistics, t-tests, ANOVA's, correlation, regression analysis
Ratio	Completion time, time (visual atten-tion), average task success (aggre-gated)	All descriptive statistics (including geometric means), t-tests, ANO-VA's, correlation, regression analy-sis

Table 5 - Tullis & Albert's description of data types, metrics and statistical procedure to perform to get valid data. The figure is an inspiration from (Tullis & Albert, 2008, p.23).

For the evaluation the question was which of the metrics could be applied and which type of data could be used in order to measure the elements that were present in the variables of interest. As stated earlier the elements in the selection technique (independent variable) were: Learnability, Efficiency, Memorability, Errors and Satisfaction. A description of how each of these elements were assessed follows.

Data logging

In order to get quantifiable results concerning the elements in the independent variable (selection technique), data logging was needed. It is considered that by recording the time the subjects use in the tutorial to learn the techniques, will give an indication towards the techniques learnability. Furthermore the time the subjects use on each of the tasks will be an indication of how effective the technique is. The output data that would describe learnability and efficiency would be ratio-based data, as described in Table 5.

The error rate can be registered by counting the amount of erroneous selections the subjects are performing on each of the task for each technique; the output data type will be nominal. These data loggings are in our case conducted by the computer, as explained in the implementation chapter.

Questionnaire design

Finally in order to get qualitative data concerning all involved elements in a manner that will ensure results that can be quantifiable, a combination of nominal and interval data can be collected in a questionnaire. The metrics used to produce the interval data were Likert scale statements, an example of such statement used in our questionnaire can be seen below.

Choose one response to following statement "I feel that this technique allowed me to select objects very efficiently"

Strongly Agree	Agree	Neither agree or disagree	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

The metric that could be used to produce nominal data concerning memorability is to set up a condition that can be confirmed by failure or success. One example could be to ask the subjects to remember the techniques that they encountered.

Furthermore, ordinal data was produced by the statements defining the target groups demographic relations. An example from our questionnaire is shown below:

How much experience do you have with one of the following types of computer interfaces?

	Never used	1 - 6 months	6 - 12 months	1 - 3 years	3+ years
Windows/Mac/Linux	☐	☐	☐	☐	☐

Further details on how the data logging, observations and questionnaire were utilized in the final evaluation will be included in the procedure chapter.

The data collection techniques, using data logging and subject self report questionnaire will produce the all types of data. In order to ensure valid estimation of the data, a clear understating of how to test these results in a statistical manner is needed. This issue will be addressed in the results chapter.

25.3 Validity of collected data

Issues concerning the validity of the collected data are going to be addressed in the following sections.

Sampling

The sampling of subjects is considered as a vital part of the evaluations validity. Part of the sampling is the methodology of how subjects are collected to the evaluation. Tullis & Albert (2008) introduced four methods: "*Random sampling, Systematic sampling, Stratified sampling and Sample of convenience*". We considered the sample of convenience to be a method that was sufficient to ensure that the sample of subjects would be a representation of the intended target group. The recruitment of test subjects was conducted at Aalborg University Copenhagen. The majority of subjects were 4th to 10th semester students attending the Medialogy program.

According to Tullis & Albert (2008) the sampling size is dependent on two factors: the goal of the study, where they recommend a small group of subjects in the earlier stages of usability testing, 3-4 subjects. In the later stages greater numbers of subjects are needed.

The tolerance of errors when dealing with tasks completion is dependent on how well the results can be statistically valid, since the independent variable (selection techniques) is measured on the dependent elements (tasks); meaning that a higher number of task completions will result in a more robust evaluation. Tullis & Albert (2008) suggested a table of confidence interval:

Number Successful	Number of Participants	Lower 95% Confidence	Higher 95% Confidence
4	5	36%	98%
8	10	48%	95%
16	20	58%	95%
24	30	62%	91%
40	50	67%	89%
80	100	71%	86%

Table 6 - Tullis & Alberts table of confidence interval, the interval of confidence is dependent on the number of participants (Tullis & Albert, 2008, p.18).

With these considerations in mind the goal was to have at least 30 participants. This number should then ensure that at least 62% (lower 95% confidence) and up to 91% (Higher 95% confidence) of the subjects would complete the given tasks. These figures are a mean of completion rate for the large public assuming an equal distribution of novice and expert users within these numbers. But only testing on novice subjects, the number of errors will be higher and the opposite for only testing expert subjects. We considered this as an acceptable confidence to conduct statistical testing of the collected data.

Closely related to the sampling of subjects is the issue of how the collected data is used for comparing each participant. Nielsen (1993) defines two kinds of studies, first is *Within-subject* study, a comparison of different data for each participant, e.g. different designs of a product. The latter is a *Between-Subjects* study, comparing results for different participants, e.g. novice versus expert users. Tullis & Albert (2008) suggest a third method a *Mixed-design*, using both approaches in the same usability test. Their argument for doing so is elimination of separate studies and being time saving. Since the goal of our study was to find the most appropriate selection technique design for single and multiple selections, a within-subject study was chosen to be the preferred methodology.

This within-subject study has the advantage of not having to take into consideration the difference across groups. The disadvantage is the "carryover effects" if using a fixed order of tasks or techniques. These carryover effects are that subjects will be biased by the tasks or techniques encountered first and subjects will have an advantage in every following task or technique. One solution for this is to counterbalance the tasks and technique order, meaning that the order can be changed for each individual subject. The counterbalancing used in the final evaluation was randomized automatically by the implementation.

Biasing of results

Tullis & Albert (2008) presents different elements that can bias the end results and these are "*Participants, Tasks, Method, Artifact, Environment and Moderator*". These issues were addressed in the evaluation of the selection techniques.

Participants (subject's status during the test)

Most of the participants that were selected to the evaluation were Medialogy students, although on different semesters, these can be considered as experts within the area of computer systems and have fairly good technical background knowledge. But they were novices in the area of Virtual Environments and this will reflect on the results.

The method of selecting subjects was to address the subjects and ask if they were interested to participate in an evaluation. If reluctant the subjects would not be chosen, since their lack of motivation for participating in the evaluation would compromise the final results.

Tasks (the tasks design)

The design of the tasks should be constructed in such a manner that the subject is asked to select a number of boxes. The subjects should not be informed if they made a mistakes or succeeded with the tasks. The reason for this is to ensure that the subjects are concentrating on selecting the boxes and not how well they are performing.

Method (the test evaluation methodology design)

The chosen evaluation methodology should consider the subjects fatigue limit, meaning that the total time of the evaluation should not be too long, since this can have an impact on the subjects motivation to further participate in the evaluation and it will therefore influence the results.

Artifact (products state of completion at the evaluation)

The goal of the final evaluation should be to have techniques that are as much as possible in their final stage and fully functional. Since the goal of the evaluation was to compare five different selection techniques, the operational state of these should be on an equal level. Otherwise one more functional technique would have an advantage over a lesser functional technique.

Environment (test environment design to the evaluation)

The test environment should be balanced between an ecological valid and a laboratory setting so that it still can be used to observe the subjects behavior in a manner that these do not feel intimidated.

Moderators (status of the moderator during the test)

The role of the moderator is to be objective towards the test results, and not to bias the test subjects. Nielsen (1993) claims that the moderators experience will have an influence on the results.

The above mentioned considerations had the results in focus. The ethical issues considered by having live subjects evaluating our implementation will be addressed in the next section.

25.4 Ethical considerations

Ethical considerations must be done when conducting usability evaluations with persons.

Burmeister (2000) addresses this issue; he identified five elements that need to be taken into account when dealing with informed consent in usability testing "*Minimal risk, Information, Comprehension, Voluntariness and Participant's rights*".

Minimal risk is coined toward the subject's physical and mental health; the usability procedure should not challenge this. The procedure level of discomfort should not be greater than those encountered in daily life.

Information concerning what the user will encounter during the evaluation should be given at the start of the procedure, together with the opportunity to ask questions and the option of withdrawal from the test at any time.

The comprehensions refer to the consequences of participating in the evaluation as e.g. understanding the consequences when giving permission of video documentation. When the subjects are fully aware of all the elements that are in the test and agreed to participate, the evaluation can be conducted.

A subject's voluntariness can be challenged on many occasions, depending on the relationship between the subject and the testing facilitator, e.g. if the subject is offered financial benefit as reward for the participation. There can be a situation where the facilitator has authority over the subjects, such as employer to employee relationship. Therefore the subject should be given the opportunity to leave the test at any point in time.

Finally the rights of the subjects should be respected, and should be shielded for the general public also when presenting the results from the evaluation, may it be observations or other means of documentation. Hence anonymity of the subjects should be respected.

26 Procedure

Having described the theoretical considerations made before the final evaluation, this section will describe how these considerations were practically executed. The structure of this section will be introduced by an overall picture of how the evaluation was conducted, and then each of the steps will be described in further details.

The first step was to introduce the subject to the evaluation, having him signing a disclosure document. Secondly the subject was asked to take on the Nvidia 3D shutter glasses and pay attention to the screen. The test started by giving an explanation of what the subjects would encounter during the test and afterwards the first randomly assigned selection technique was introduced. For each technique a tutorial stage allowed the subjects to try and learn the technique for as long as they liked. After subjects had finished the tutorial they performed six randomly assigned tasks. This procedure was repeated for all selection techniques. When the subject had tried out all the techniques, he/she was asked to fill in an online questionnaire and was then excused from the evaluation.

26.1 Test environment

The test environment was constructed in a manner that it supported the technical consideration that must be made when dealing with a projected 3D environment. The subject would enter a darken room containing the VE installation; there were no other visible signs of that the subject was monitored by the investigators. The moderator was sitting on the subjects left side and the controller was sitting behind the projection screen, not visible to the subject.

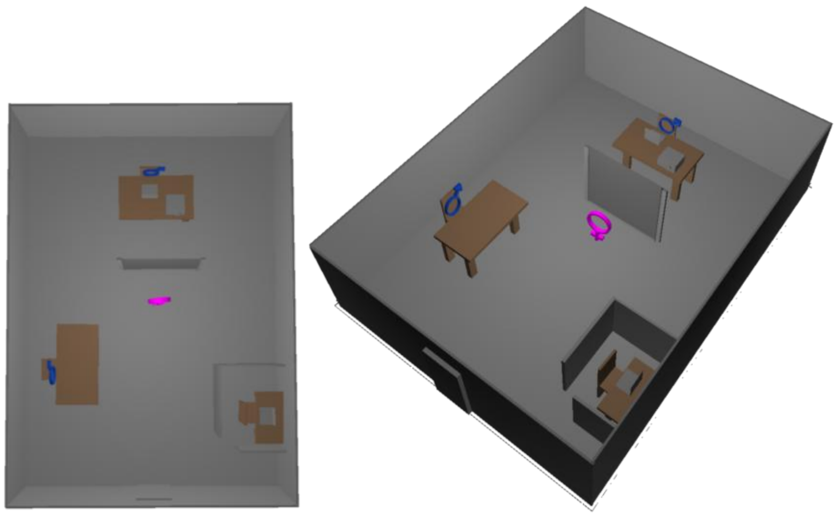


Figure 48 depict a top view and side view of the evaluation setup.

After the subject had tried out all the techniques, he was directed to a section where he could sit undisturbed and fill in the questionnaire.

The hardware and software used in the evaluation consisted of

- Unity 3.3[1], software game engine for running the implementation
- IntelPentium QuadCore 2.4 GHz, 4GB ram, Geforce8800GTS, PC running Windows 7
- Nvidia 3D vision™[2], consisting of active shutter glasses and 3D Vision USB controller/ IR emitter, allowing the subject to see the rendered scene in 3D
- InFocus™ DepthQ - DQ3120 stereoscopic 3D projector. 120Hz, 1024x768 resolution

26.2 Test introductions

Each test participant was provided with a set of screen instructions during the test procedure. This was done in order to ensure that all test participants would receive the same set of information, while not biasing them towards any particular response (Rademacher, 2002). In order to avoid the problem of what in social studies is referred to as "problem of demand characteristics", the instructions presented were designed carefully. This problem refers to the subjects desire to give results that they perceive as helpful towards the moderators research (Denzin, 2009). Denzin (2009) suggests an approach that is balancing between an explicit and unclear set of instructions, meaning that the user should know what to perform, but is to a certain degree not able to guess the purpose of the evaluation.

The following box contains the instructions the subject received before the first techniques was introduced.

Test instructions

In this test you will be introduced to five different techniques, which allow you to select objects, by using your hand and arms as controls. Each technique will be explained to you, after which you can try out and practice with the technique for as long as you like. After practicing with a technique you are asked to complete a small trial. In the practice and the trial you can select boxes. These boxes will light up in a green color when selected. When highlighted they will light up in a blue color. During the trial some boxes will light up in a Pink color. These are the boxes you need to select. Once selected the next trial will start. If you accidently select wrong boxes the next trial will also start. If you feel unable to complete a trial, raise both your arms and the next trial will start. If anything is unclear to you during the test, feel free to ask. You may withdraw from this test at any time! Now to start with the first technique raise both your arms.

For each of the techniques the subjects received an explanation on how to control it, which can be found in the appendix, section III.c).

[1] More information can be found on: http://unity3d.com/unity/

[2] More information can be found on: http://www.nvidia.com/object/product_GeForce_3D_VisionKit_us.html

26.3 Task design

The following sections describes: the details of the construction of tasks and relation to the data collection methodology, why this approach was taken in order to obtain data concerning the variables in the hypothesis, which subjects were chosen and how the evaluation environment was constructed in order to ensure a non-biasing environment.

Placement of boxes

The first issue concerning the tasks was the placement of the boxes that should be selected by the user. In order to ensure a full enquire of the selection techniques the boxes were placed in a level design structure. The structure of the boxes was as following: Foreground (F), Middle ground (M) and Background (B).

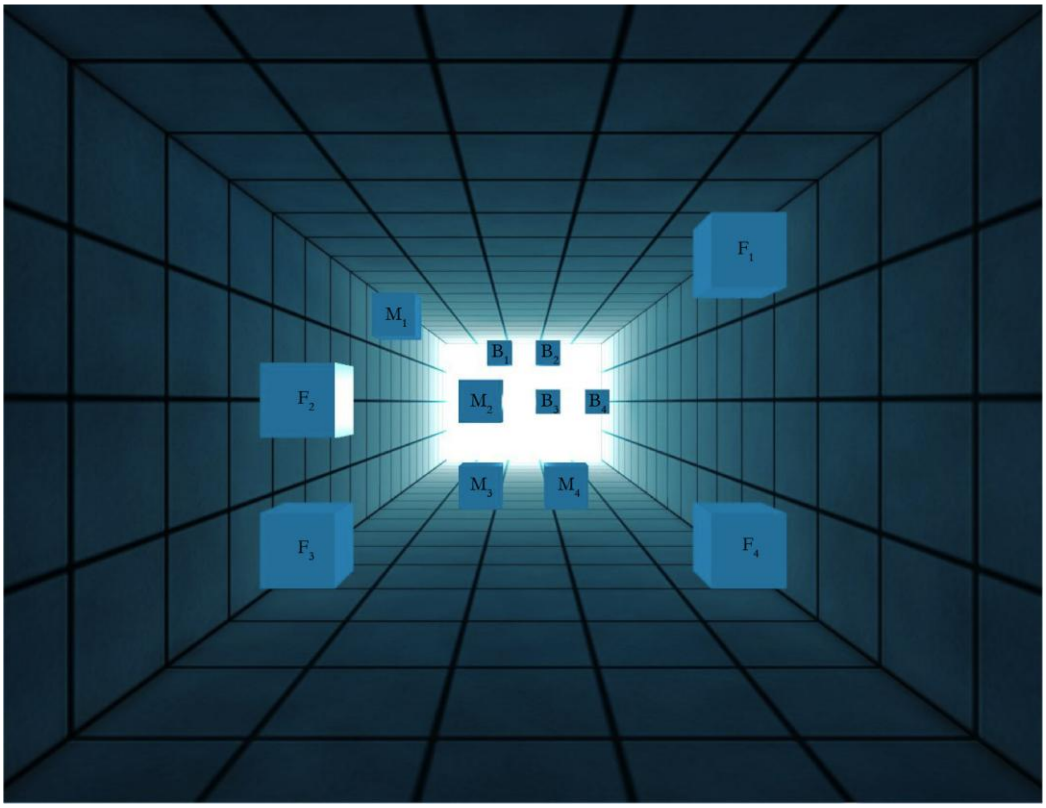


Figure 49 Depicts the placement of boxes and their corresponding identification; Foreground (F), Middle ground (M) and Background (B) and numbering lowest to highest from left to right.

This structure should prove the selection techniques ability to operate in a 3D virtual environment. The placement of the boxes was ordered in such a manner to ensure as little occlusion of boxes as possible, no matter where the subject was positioned. Further the three different depth levels should prove the techniques ability to function accurately, which we regarded important to prove their ecological validity towards being usable in other applications.

Randomization of techniques and tasks

The tasks were constructed in order to represent the situations the user could be in while using a selection tool. Furthermore the tasks were constructed in such a manner to ensure that the data extracted from the procedure was as little as possible biased by the order the selection techniques were presented to the test subjects.

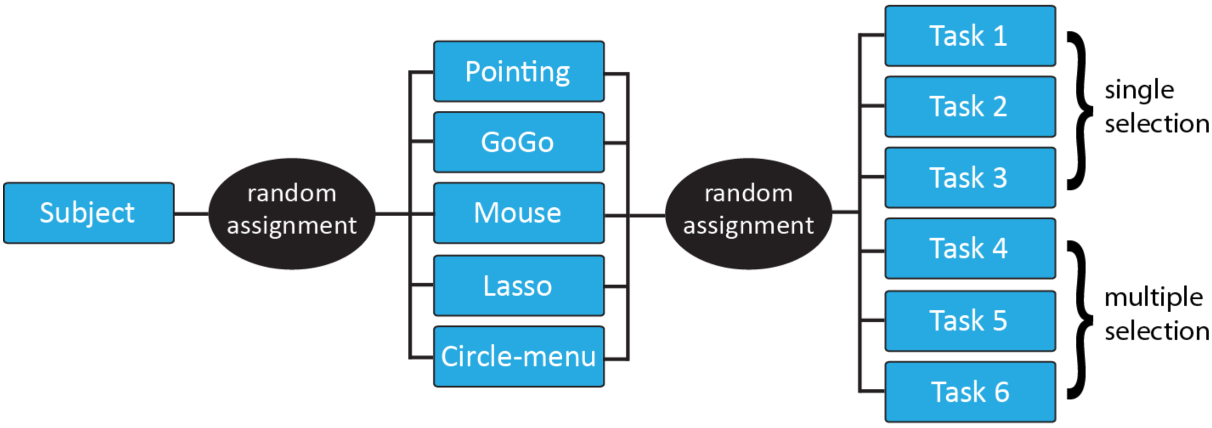


Figure 50 - Depicts how the user will get a technique in a random order with corresponding tasks presented in a random order.

In order to avoid selection techniques carry-over effect, which can take place when there is a fixed presentation order, the selection techniques were presented in a random order. Hence the user would have a 1/5 possibility of starting with the same technique as the previous user. This random order can therefore reduce the possibility of having carry-over effect among techniques. The same issue applies for the task presentation; each task was randomly presented to the test subject. The connection between techniques and tasks is depicted in Figure 50.

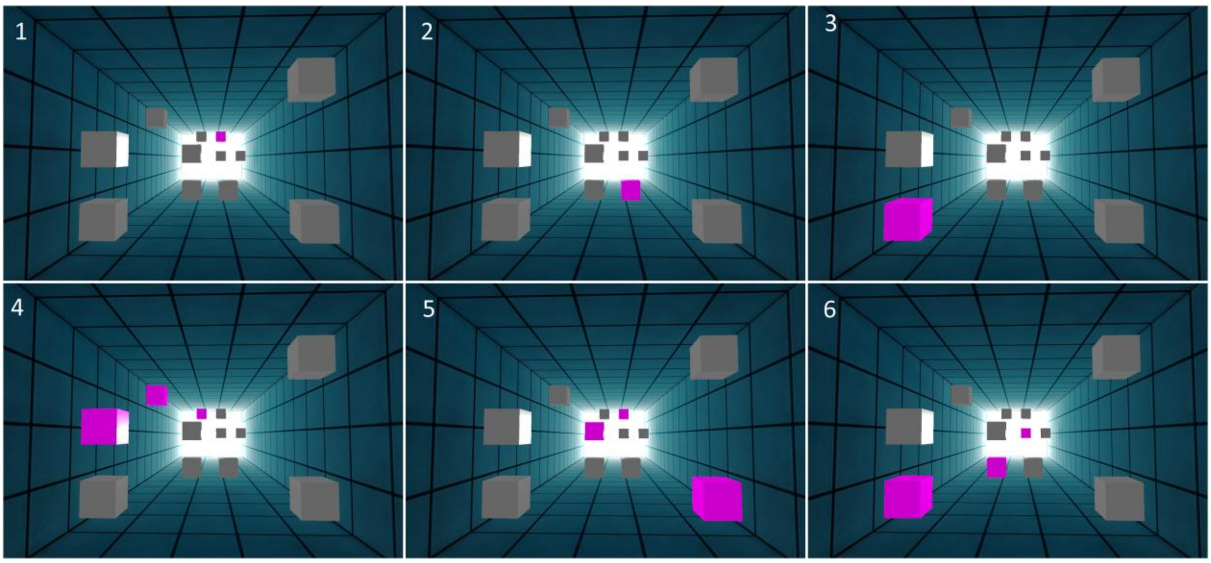


Figure 51 - Depicts the position of tasks in the final evaluation; they are in respective order; single selection tasks 1-3 and multiple selection tasks 4-6.

The combination of boxes in the tasks is constructed in such a manner that the user should not be able to predict a pattern, hence learn to know what the next combination of boxes is going to be. The constructed tasks should be a representation of the different situations a user could encounter in a real life VE application. Furthermore the numbers of tasks were carefully selected in order to balance the carryover factor (where the user learns the order). The chosen number of tasks was three single and three multiple selection tasks, as shown in Figure 51.

26.4 Questionnaire design

The choice of having the subjects filling in the questionnaire after finishing all the techniques was due to time concerns, having the subject re-calibrate the application after each technique questionnaire, would be too time consuming. The concern for the subject's memorability concerning each of the techniques was addressed by having pictures representing each of the techniques hanging in the self-report room. The whole questionnaire can be found in the appendix, section III.a).

The questionnaire was designed with Google spreadsheet and this choice was due to the flexibility of the tool and easy access of the data that it produced. When the questionnaire is filled online the data is saved in a file format at the user's choice, in our case Microsoft Excel format.

The questionnaire was divided into different parts. The first part had the purpose of gathering demographic information concerning the test subject. The second section was designed to gather information concerning the selection techniques the subject had experienced.

Demographic information

The purpose of the question asked in this section was to identify the test subject's demographic information, and experience with computer and VE systems.

Since the subjects' experience with VE and computer systems can have great effect on the results data concerning this issue was collected. Questions addressing the test subjects computer system experience were asked since this experience can have an impact on how well the subject can relate to the principles utilized in the implemented techniques.

How much experience do you have with one of the following types of computer interfaces?

	Never used	1 - 6 months	6 - 12 months	1 - 3 years	3+ years
Windows/Mac/Linux	☐	☐	☐	☐	☐

Furthermore, one goal was to determine the users previous experience with interfaces, since this could give an indication of how well the user is capable of adjusting to the techniques. It was also important to find out if any user had experience with the Kinect or similar tracking devices, in order to ensure that the subjects are part of the intended target group.

Which of the following input devices are you familiar with? (Check all that apply)

☐ Keyboard / mouse
☐ Joystick
☐ Touch Screen (iPhone / iPad / android / tabletops / etc…)
☐ Pen / Stylus / Drawing tablet
☐ 3D tracking devices (marker tracking, Kinect, other 3d trackers)

Selection technique information

The questions concerning the selection techniques were asked in a linear order from technique one to five. The questions were constructed with a Likert scale statement; these statements were derived from the usability elements "*Learnability, Efficiency, Memorability, Errors, Satisfaction*"

The reason for this approach was to have qualitative data that could be quantified over a chosen target group.

Choose one response to following statement "I feel that this technique allowed me to select objects very efficiently"

Strongly Agree	Agree	Neither agree or disagree	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

The results from the questionnaire would be assisting the observation and data logging that was performed during the evaluation, and from this data we would have a clear picture of what the subjects thought concerning the usability of the selection techniques, and prove or deny the stated hypothesis concerning an ideal single and multiple selection technique in a 3D Virtual Environment.

27 Test results

The final evaluation was conducted at Aalborg University Copenhagen in the period from the 2nd to the 4th of May 2011. The 33 participants in the final evaluation ranged in age from 21 to 39 years. There were 4 female and 29 males students we met in the department of Architecture, Design and Media Technology. Results of one of the participants were disregarded due to technical complications that occurred, hence the final number of participants were 28 males and 4 females.

Although Likert scale is technically termed ordinal data, Tullis & Albert argued that Likert scale can have interval data properties. In this way it is assumed that the distance between a 1 and a 2 on the Likert scale has the same distance as a 2 and a 3 on the same scale, an assumption that is regarded as degrees of intervalness. Furthermore the data in between two positions should also be appointed a meaning. Considering these assumptions the data obtained can be regarded as close enough to interval data to be treated as such (Tullis & Albert, 2008).

Agreeing with these assumptions we will treat the questionnaire data as interval data, likewise with the task results, analyzing each of the results with ANOVA and two-tailed paired t-test. We start with the assumption that the data collected was normally distributed, hence gave a general picture of how the targeted group felt towards the five different selection techniques.

27.1 Questionnaire results

Demographic data

All of the participants were daily using a computer and had over three years of experience in the subject. All participants were familiar with keyboard/mouse input devices and had used them on a daily basis. Most participants were familiar with joysticks (26 out of 32), but only one used it on a daily basis, three few times a week, nine few times a month and thirteen rarely or never.

Almost all participants (30 out of 32) were familiar with touch screen devices, where nineteen were using one on a daily basis, two few times a week, two few times a month and six rarely or never.

Three quarters (24 out of 32) of the participants were familiar with drawing tablets, one used it daily, one few times a week, eight few times a month, eleven rarely or never.

Most participants had no experience with 3D tracking devices (12 out of 32) where one participant had daily experience, one had few times a week and four participants used them a few times a month, six were rarely or never using 3d tracking devices.

A large amount of the participants (21 out of 32) were familiar with Wii/ PlayStation Move, none were using them daily, three were experiencing them on few times a week, and nine were using it rarely or never.

Some of the participants (6 out of 26) rated that they had experienced other devices, such as joypads, microphones, 3d connection joystick, PlayStation Eye toy and Xbox.

Most participants had experienced a stereoscopic display (30 out of 32) and approximately half of the evaluated subjects (14 out of 32) were familiar with 3D interaction techniques.

Post evaluation question concerning memorability, indicated that almost all subjects (31 out of 32) could recall all the techniques they had encountered, where one subject did not remember one of the techniques.

Usability data

The collected usability data reflects the participants rating towards five different statements. The following figures will depict the individual rating the questions addressed about the five selection techniques. The rating scale for how comfortable the techniques were considered ranged from normal (1) to extreme discomfort (3). The rating scale for the rest of the statements ranged from strongly disagree (1) to strongly agree (5).

Comfort

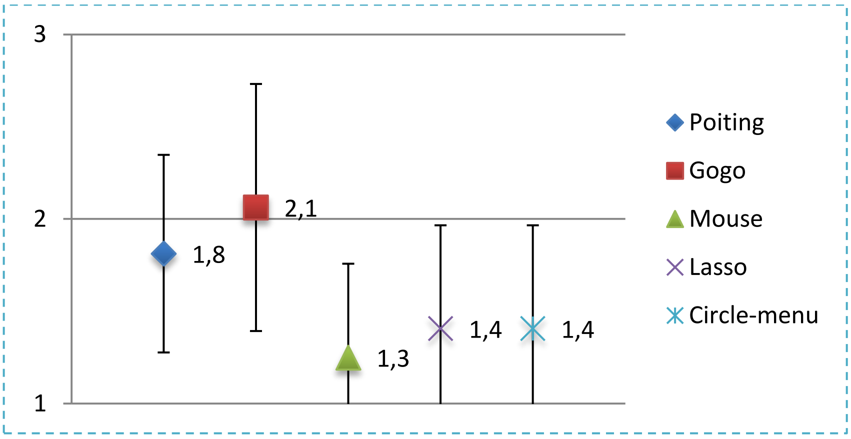


Figure 52 - Depicts the average rating for how comfortable is to use the technique. Error bars indicate ± standard deviation from the mean.

The average rating for Pointing for comfort was 1,8±0,5. The average rating for GoGo was 2,0±0,7. Mouse had an average of 1,3±0,5. Lasso and Circle-menu had an identical average of 1,4±0,6.

Learnability

Subjects were asked to rate how easy it was to use the selection techniques.

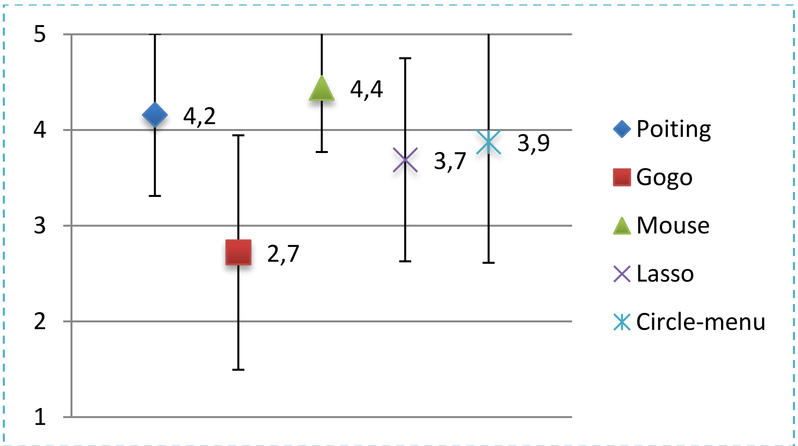


Figure 53 - Depicts the average rating for the statement “It was very easy to learn how to use this technique”. Error bars indicate ± standard deviation from the mean.

The participants rated Pointing with an average of 4,2±0,8. GoGo average rating was 2,7±1,2. Mouse had an average of 4,4±0,7. The rating average of Lasso was 3,7±1,1. Circle-menu had a rating average of 3,9±1,3.

Efficiency

The statement in this section is concerning the users subjective rating of the five different techniques efficiency.

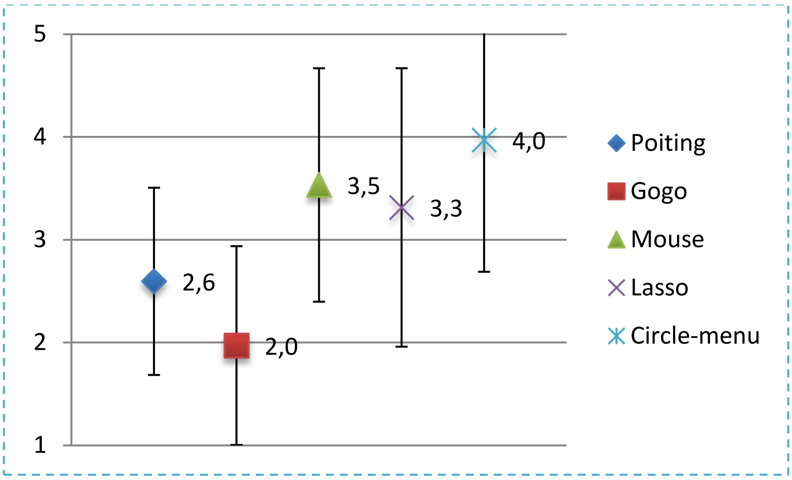


Figure 54 - Depicts the average rating for the statement “I feel that this technique allowed me to select objects very efficiently”. Error bars indicate ± standard deviation from the mean.

Participants average rating of Pointing was 2,6±0,9. The average rating for GoGo was 2,0±1,0. Mouse technique had an average rating of 3,5±1,1. Lasso had a rating average of 3,3±1,4. The rating average for Circle-menu was 4,0±1,3.

Errors

Subjects were asked to rate their error rate in relation to tasks and individual selection techniques.

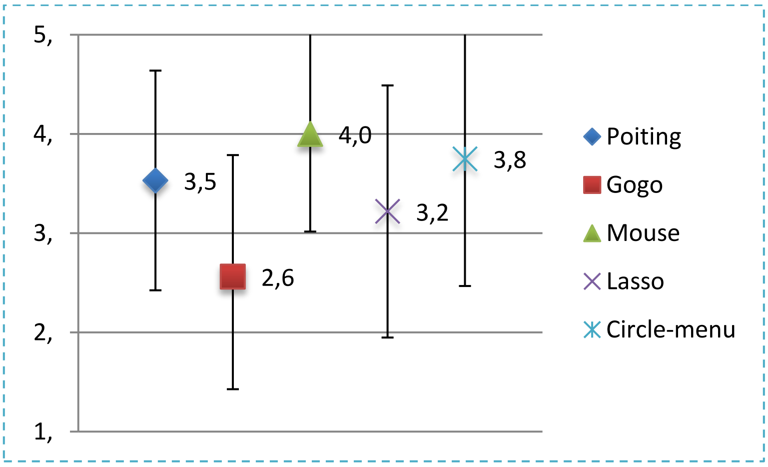


Figure 55 - Depicts the average rating for the statement “I felt I selected the objects I intended to select”. Error bars indicate ± standard deviation from the mean.

Pointing had a rating average of 3,5±1,1. The rating average for GoGo was 2,6±1,3. The Mouse technique had an average rating of 4,0±1,0. Lasso had a rating average of 3,2±1,3. Circle-menu had an average of 3,8±1,3.

Satisfaction

The last rating scale was directed to the participants feeling of how pleasant each of the techniques was to use.

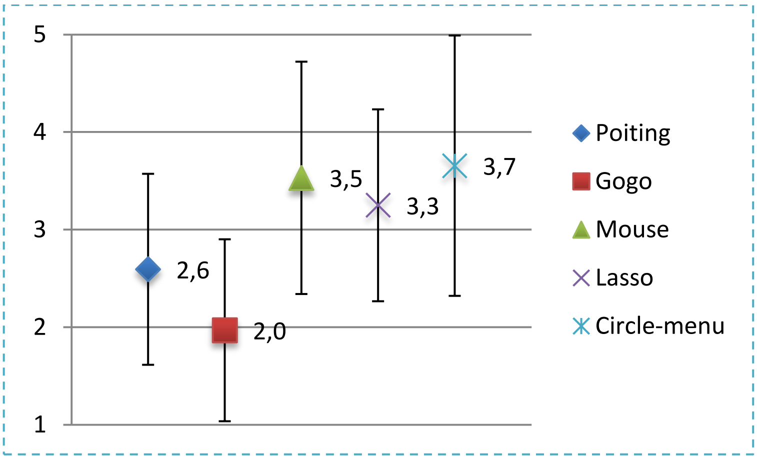


Figure 56 - Depicts the average rating for the statement "This technique is very pleasant to work with". Error bars indicate ± standard deviation from the mean.

The Pointing technique had a rating average of 2,6±1,0. GoGo had an average of 2,0±0,9. Mouse techniques' rating average was 3,5±1,2. Lasso had a rating average of 3,3±1,0, and the Circle-menu technique had a rating average of 3,7±1,3.

27.2 Analysis of questionnaire answers

In order to avoid type 1 and 2 errors the analyses of variance (ANOVA) was used. The type of ANOVA that is utilized in this investigation is two-factor without replication, assuming the null hypothesis that the users are the same and there is no significant difference (p>.05) between the selection techniques.

The ANOVA analysis indicated a variance in all stated questions. Question concerning Comfort shows a P-value for the users of 0,000630174, hence the null hypothesis is rejected (p<.05). The P-value for the techniques was 1,40777E-09, which is also a rejection of the null hypothesis of all techniques being evenly pleasant to work with.

The Learnability statement results express a P-value of 0,023755186 for the users, hereby rejecting the null hypothesis and a P-value of 9,55138E-10 for the learnability of selection techniques, also rejecting the null hypothesis.

The ANOVA of Efficiency statement results accept the null hypothesis of users being the same (P-value = 0,323040536). Furthermore the analysis rejected the null hypothesis for selection techniques being equally efficient of selecting boxes (P-value =1,79324E-10).

The analysis of Error results reveals a P-value of 0,994606269, herby accepting the null hypothesis of the users being identical. A P-value of 8,04666E-05 rejects the notion of the techniques being equally in conducting errors during the tasks.

The ANOVA of Satisfaction statements imply an acceptation of the users being identical (P-value = 0,104530461). Furthermore the analysis rejects the belief that all the selection techniques were equally satisfactory for the users (1,07624E-09).

Since the ANOVA indicates a difference between the techniques, paired t-tests can identify the location if the significant difference.

T-tests

The ANOVA identified significance between techniques in all respective areas: comfort, learnability, efficiency, errors and satisfaction. In order to gain more detailed information about which techniques are different from the others, a t-test is conducted on paired techniques.

T-Test
Two-Tail (paired)

	Comfort	Learnability	Efficiency	Errors	Satisfaction
P-G	0,018344605	1,40923E-06	0,003047309	0,001917196	0,004589908
P-M	4,31477E-05	0,173685947	0,001703034	0,104589327	0,001313783
P-L	0,004781027	0,033381779	0,005843412	0,352830363	0,007007962
P-C	0,001618269	0,141471908	6,17501E-06	0,499074778	0,00021657
G-M	1,47552E-05	2,02597E-08	1,24182E-05	7,20392E-05	7,35793E-06
G-L	9,28614E-05	0,00213766	2,5064E-05	0,046370297	1,05995E-06
G-C	0,000162405	0,000453541	1,05239E-07	0,000173631	9,30027E-06
M-L	0,201462121	0,00407969	0,526125955	0,016117808	0,348152855
M-C	0,133831032	0,04494024	0,194297817	0,44589593	0,657830059
L-C	1	0,447334525	0,034376436	0,090935463	0,135564703

Table 7 - Depicts if one of the techniques; Pointing (P), GoGO (G), Mouse (M), Lasso (L) and Circle-menu (C) is significant different (p<.05) from the other techniques in one and many of the respected categories; Comfort , Learnability, Efficiency, Errors and Satisfaction. The boxes that are highlighted with green indicate that there is a significant difference between two techniques.

T-test indicates that in the category of comfort by the Pointing technique is significant different (p<.05) than all the other selection techniques. GoGo is also significant different than the other techniques, while the Mouse-Lasso, Mouse-Circle-menu and Lasso-Circle-menu relations show no significant difference.

In the Learnability category Pointing is significant different from the GoGo and Lasso techniques, but not from Mouse and Circle-menu. GoGo technique is significant different from all the techniques. The Mouse techniques is significant different from most techniques, but not Pointing. Lasso is significant different from most, but not the Circle-menu. Finally the Circle-menu is significant different from Mouse and GoGo, but not from Lasso and Pointing techniques.

The relations between the different techniques in the Efficiency category indicate that Pointing is significant different from all the other techniques. GoGo is also significant different from the other techniques. The Mouse technique is significant different from Pointing and GoGo, but not from Lasso and Circle-menu. The Lasso technique is significant different from most of the techniques, but not from Mouse. Finally the Circle-menu differs from all but Mouse technique.

The Error category suggest that the following techniques are significant different from the others, Pointing is only significant different from the GoGo technique. The GoGo is significant different from all techniques. Mouse is significant different from GoGo and Lasso. The Lasso technique is significant different from GoGo and mouse. The Circle-menu differs from GoGo.

In the Satisfaction category Pointing and GoGo are significant different from all techniques. Mouse, Lasso and Circle menu are significant different from Pointing and GoGo, but are not significant different from each other.

More details concerning the questionnaire results can be found on the CD.

27.3 Task results

Learnability

The tutorial time used by the subjects can give an indication on how well the subject understood the technique.

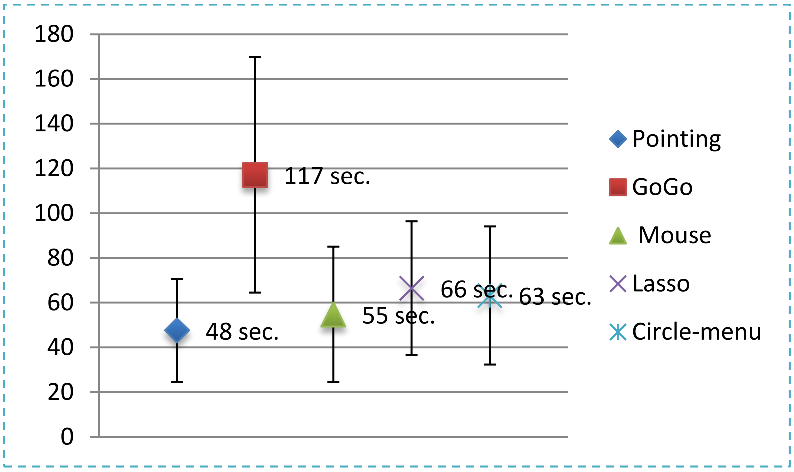


Figure 57 - Depicts the average tutorial time used in each of the technique. Error bars indicate ± standard deviation from the mean.

The participants spent a mean time of 48±23 seconds on the pointing technique. The average time spent on the GoGo technique was 117±53 seconds. Subjects were spending a mean tutorial time of 66±30 seconds on the Mouse technique. The Lasso techniques mean tutorial time was 66±30 seconds. Participants spent a mean value time of 63±31 seconds to learn the Circle-menu.

Efficiency

The completion time is calculated without fatal error task, other error tasks are included in the calculation of completion time for techniques in the categories single and multiple selection tasks.

Single selection tasks consisted of three individual one-box selection tasks, the mean completion value of theses task combined together constitutes the mean single task completion time.

The completion time is calculated in total and did not include fatal error tasks.

Single-selection

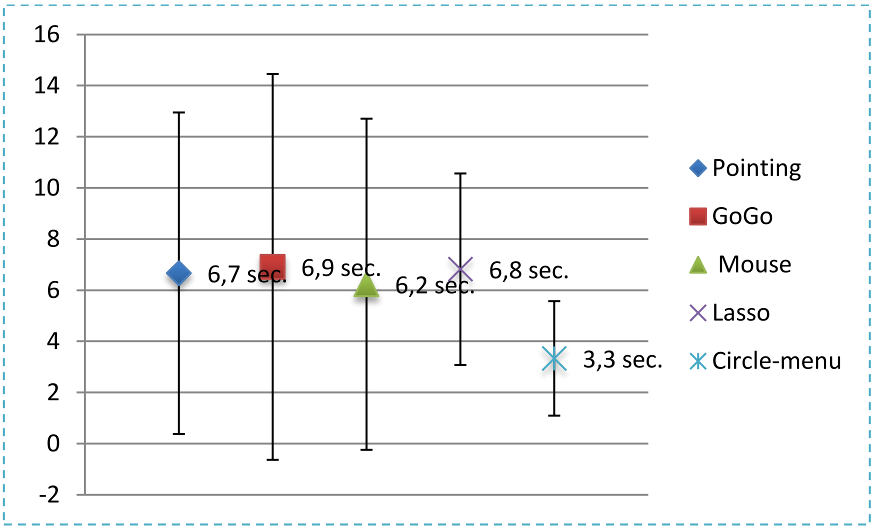


Figure 58 - Depicts the mean completion time for each of the five techniques in a single selection task. Error bars indicate ± standard deviation from the mean.

Participants were spending a mean time of 6,7±6,3 seconds to complete a single selection task with the Pointing technique. Completion time for the GoGo technique was 6,9±7,5 seconds. Using the Mouse technique the subjects were spending 6,2±6,5 seconds to complete a single selection task. The Lasso technique had a mean completion time of 6,8±3,7 seconds, and finally when the participants were using the Circle-menu they spent a mean time of 3,3±2,2 seconds to complete the single selection task.

Furthermore the participants were asked to conduct multiple selections with the techniques - selecting three boxes in each of the three tasks. The mean completion time was calculated in the same manner as in the single selection tasks.

Multiple-selection

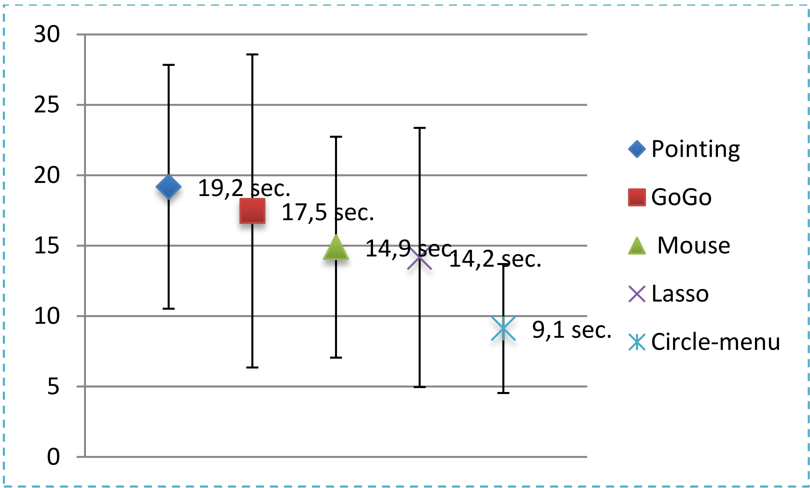


Figure 59 - Depicts the mean completion time for each of the five techniques in a multiple box selection task. Error bars indicate ± standard deviation from the mean.

While using the Pointing technique, the participants had a mean task completion time of 19,2±8,7 seconds. The GoGo technique had a mean completion time of 17,5±11,1 seconds.

Participants were completing the tasks with the Mouse technique with a mean time of 14,9±7,8 seconds. When using the Lasso technique, the participants spent a mean task completion time of 14,2±9,2 seconds. The Circle-menu had a mean completion time of 9,1±4,6 seconds.

Errors

Task error rate was taken into consideration in order to give a full performance description of all the evaluated selection techniques. The tasks error rate was divided up into single and multiple selections, in order to give a performance indication of each individual techniques capability of selecting boxes. Single and multiple box selection mean error rate was calculated in the same manner as in the completion time section. The error rate will be presented as frequencies and cross tabulation.

Single selection tasks

The nature of the errors was categorized in four different categories: Zero error (task successful completed), 1-2 errors, 3+ errors and Fatal error (the subject gave up during the task).

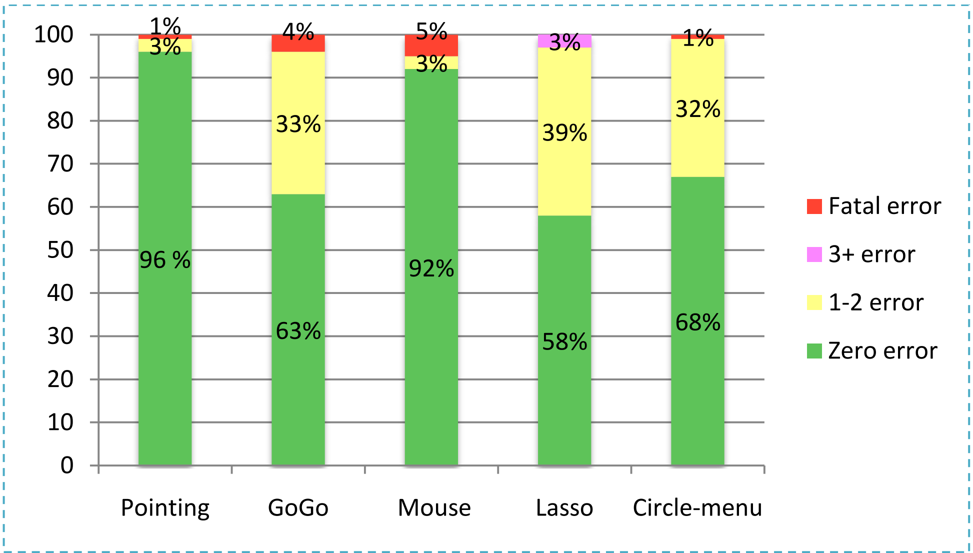


Figure 60 - Depicts the different categories of how the task error were rated in the single selection tasks; Zero error (task completion), 1-2 errors, 3+ errors and Fatal errors (subject gave up during the task).

Pointing had a success rate of 96%, 3% were 1-2 error tasks and 1% was fatal error tasks. GoGo had a success rate of 63%, 33% 1-2 error tasks and 4 % fatal error tasks. The Mouse technique had a success rate of 92%, 3% 1-2 error tasks and 5% fatal error tasks. Lasso had a success rate of 58%, 39% were 1-2 error tasks and 3% were 3+ error tasks. Circle menu had a success rate of 67% and 32% of 1-2 error tasks and a fatal task error rate of 1%.

Going deeper in further detail into the amount of errors that were conducted, the total amount and mean distribution can be calculated for each of the single selection tasks.

Amount of errors Single selection	Pointing	GoGo	Mouse	Lasso	Circle-menu
Task 1	2	17	0	26	12
Task 2	0	8	1	12	5
Task 3	1	7	0	13	14
Mean distribution of errors	0,03125	0,333333333	0,010416667	0,53125	0,322916667

Table 8 - Depicts the sum of errors to each of the selection techniques conducted in the single selection tasks. The mean distribution of each task is calculated in regards to the amount of subjects that were able to perform a task without fatal error.

The mean distributions of errors for single selection tasks are respectively: Pointing (0,03 errors per task), GoGo (0,33), Mouse (0,01), Lasso (0,53) and Circle-menu (0,32 error per task).

Multiple selections tasks

The division of different errors in multiple selection tasks was categorized in the same manner as in single selection task.

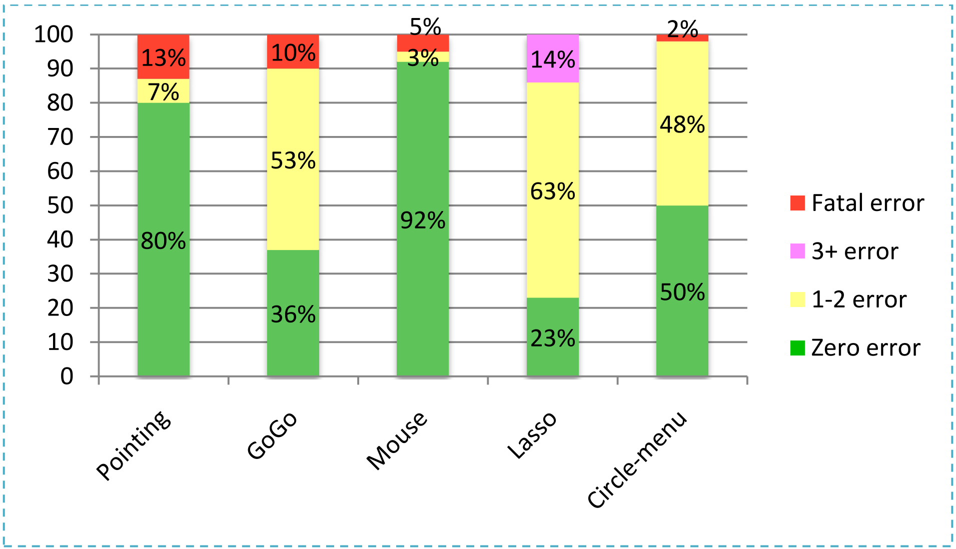


Figure 61 - Depicts the different categories of how the task error were rated in the single selection tasks; Zero error (task completion), 1-2 errors, 3+ errors and Fatal errors (subject gave up during the task).

The Pointing technique had in multiple selection tasks a success rate of 80%, 7% 1-2 error tasks and 13% fatal error tasks. GoGo success rate was 37%, having 1-2 error rate of 53% and a fatal error rate of 10%. The Mouse technique had a success rate of 92%, 3% 1-2 error tasks and 5% fatal error tasks. Lasso technique had 23% successful task completions, 63% 1-2 error tasks and 14% 3+ error tasks. The Circle-menu had a success rate of 50%, 48% 1-2 error tasks and 2% fatal error tasks.

Amount of errors Multiple selection	Pointing	GoGo	Mouse	Lasso	Circle-menu
Task 4	3	12	2	39	18
Task 5	0	17	1	40	28
Task 6	4	31	0	44	7
Mean distribution of errors	0,072916667	0,625	0,03125	1,28125	0,552083333

Table 9 - Depicts the sum of errors to each of the selection techniques conducted in the multiple selection tasks. The mean distribution of each task is calculated in regards to the amount of subjects that were able to perform a task without fatal error.

The mean distribution of task errors per techniques is Pointing (0,07), GoGo (0,63), Mouse (0,03), Lasso (1,28) and Circle-menu (0,55 errors per task).

27.4 Analysis of task results

The methodology of analysis will be the same as in the case of the questionnaire data, using ANOVA to acquire a general picture and t-test to get into details. There are two categories that were covered in the task results and these are Learnability and Efficiency.

Learnability

The Analysis of variance (ANOVA) was conducted to investigate if there was a difference among the subjects, and if there was a difference of tutorial time among the different selection techniques.

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Rows	96434,69176	4	24108,67294	25,98054867	1,19065E-15	2,444766161
Columns	72800,17674	31	2348,392798	2,530729648	0,000162793	1,543950052
Error	115065,9088	124	927,9508777			
Total	284300,7773	159				

Table 10 - Depicts the Analysis of variance among subjects (rows) and techniques (Columns) in regards to the tutorial time. The green highlighted P-value numbers indicate a significant difference among users and techniques.

The analysis rejects the null hypothesis of the subjects being identical (p<.05). Furthermore the techniques are also regarded as significantly different (p<.05).

Since it is clear that there is a difference between the techniques, a t-test analysis will depict the significant difference between the techniques.

P-G	1,15648E-08
P-M	0,205706481
P-L	0,00505294
P-C	0,002624957
G-M	1,84112E-08
G-L	2,20011E-05
G-C	4,72289E-06
M-L	0,107905939
M-C	0,162922139
L-C	0,659394772

Table 11 - Depicts if one of the techniques; Pointing (P), GoGO (G), Mouse (M), Lasso (L) and Circle-menu (C) is significant different (p<.05) from the other techniques in regards to the tutorial time. The boxes that are highlighted with green indicate that there is a significant difference between two techniques.

The t-test indicates that Pointing differs from GoGo, Lasso and Circle-menu, but there is no significant difference between Pointing and Mouse technique. The GoGo technique is significant different from all techniques. Mouse technique is significant different from GoGo, but not from the other techniques. The Lasso technique is significant different from Pointing and GoGo, but not from the other two techniques. Circle-menu differs from Pointing and GoGo, but not from Mouse and Lasso.

Efficiency

The efficiency was covering two types of tasks: single and multiple selections. These tasks will not be compared against each other, but there will be an individual investigation to find out if there was a significant difference between the techniques and the users in the respectively single and multiple selections tasks.

Single selection

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Rows	3192,172083	95	33,6018114	1,117601076	0,234369184	1,291052925
Columns	884,2751096	4	221,0687774	7,352779308	1,03363E-05	2,395429935
Error	11425,08593	380	30,0660156			
Total	15501,53312	479				

Table 12 - Depicts the Analysis of variance among subjects (rows) and techniques (Columns) in regards to single selection tasks efficiency. The green highlighted P-value number indicates a significant difference (p<.05) among the techniques.

The analysis rejects the null hypothesis concerning the selection techniques (p<.05).

Multiple selection

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Rows	8248,656573	95	86,82796392	1,435254313	0,009747658	1,291052925
Columns	5518,23559	4	1379,558897	22,80391901	6,67404E-17	2,395429935
Error	22988,69684	380	60,49657064			
Total	36755,58901	479				

Table 13 - Depicts the Analysis of variance among subjects (rows) and techniques (Columns) in regards to multiple selection tasks efficiency. The green highlighted P-value numbers indicate a significant difference (p<.05) among users and techniques.

The analysis of variance rejects the null hypothesis concerning the selection techniques (p<.05).

Since the analysis of variance has identified a variance in the data concerning the efficiency of selection among single and multiple selections tasks, a t-test will be utilized in order to get a full overview of these differences.

	Single (P)	Multi (P)
P-G	0,814041705	0,076336271
P-M	0,589824085	0,001879445
P-L	0,834747629	0,000100301
P-C	8,1907E-06	6,56654E-16
G-M	0,386442628	0,070233895
G-L	0,904945335	0,053805373
G-C	2,13461E-05	4,74806E-09
M-L	0,541433558	0,384959047
M-C	0,000103152	5,29839E-08
L-C	3,73453E-13	1,14909E-08

Table 14 - Depicts if one of the techniques; Pointing (P), GoGO (G), Mouse (M), Lasso (L) and Circle-menu (C) is significant different (p<.05) from the other techniques in regards to the efficiency in regards to single and multiple selection tasks. The boxes that are highlighted with green indicate that there is a significant difference between two techniques.

In single selection tasks the Circle-menu is significant different from all techniques (p<.05). The other techniques are not significant different from other techniques than the Circle-menu.

In the multiple selection tasks the Pointing technique is significant different from Mouse, Lasso and Circle-menu. The Circle-menu is significant different from all techniques. The Mouse and Lasso are significant different from the Pointing and Circle-menu. The GoGo is only significant different from the Circle-menu.

The presentation and analysis of the final evaluation results will form the basis for the following discussion chapter.

PART VI - Discussion & Conclusion

The following discussion will be divided into different parts. In the first part we will discuss the results implication on the usability; here the focus is on the findings from the evaluation in relation to the parts constituting to the usability. In the second part we will try to understand the results in relation to the target group. In the third part we will discuss the results implications for interaction techniques; here the focus is on the generalization of the results in relation to the theory and a discussion of which design elements constitute to the success of a technique

28 Discussion

28.1 Results

The goal of the evaluation was to measure the usability of the implemented ITs, in order to find out which of these techniques is best suited for single and multiple selection of screen content. Since the evaluation was focused on the different parts of the usability as defined by Jacob Nielsen (Nielsen, 1993), the discussion will be structured matching the different parts constituting to this definition, which are: learnability, memorability, efficiency, errors, satisfaction and comfort.

Learnability

For the learnability of the techniques the data from the questionnaires showed no clear winner without ambiguity. The analysis showed that the mouse and the pointing technique received highest ratings compared to the other techniques. For the mouse technique significance was found compared to all techniques except the pointing technique. For the pointing technique significance was found compared to the lasso and the GoGo -technique. Although these results do not show an unambiguous winner, it shows a tendency towards the mouse and the pointing technique being the easiest techniques to learn. For the mouse technique 29 out of the 32 participants agreed or strongly agreed to the statement if the technique was easy to learn. It is interesting to note that none disagreed with the statement for the mouse technique. Although no unambiguous answer for the winner can be found the questionnaire data clearly shows that the GoGo technique was rated lowest, with significance to all other techniques.

From the measured data of the time spend on the tutorial scenes for each technique similar results to the questionnaire data can be found. Here the GoGo technique shows significant differences in learning time spend when compared to the other techniques. The pointing technique shows significant difference to all other techniques except the mouse technique. The mouse technique however shows no significance to the lasso and the circle-menu technique. Although again a winner without ambiguity cannot be found the tendencies show that the pointing technique and the mouse technique were easiest to learn, whereas the GoGo technique clearly is harder to learn the others.

For the learnability it can be discussed that both questionnaire and measured data gave weak indications towards the mouse and the pointing techniques being the easiest techniques to learn. The clear looser is the GoGo technique, which can be concluded to be harder to learn than the other techniques. Although no clear winner can be found for the learnability the tendencies meet our expectations of that the mouse and the pointing techniques should be the easiest to learn. The intention with the design for these two techniques was also laid out to be easy to learn. Further discussion of which aspects of the design might constitute to the ease of learning will be presented in the third part of the discussion chapter.

Memorability

Results from the questionnaire data for the memorability aspects of the usability show that almost all participants (31 out of 32) were able to remember all the techniques. Only one participant did forget the actions used for the lasso-tool. These results are positive, though the long time effect was not measured. The memorability only was measured as an immediate effect right after the test was completed. For a correct measurement of the memorability another study would be needed, which measures on returning participants with a period of time in between the first study. However, we will use the clear data as an indication of the memorability, but not conclude on it. This means that if no participant could remember the techniques right after the test would indicate that the actions required were simply too complicated to remember. With the obtained results we can assume that it should not be too complicated to remember the techniques, whereas another study is needed to confirm this assumption.

Efficiency

For the efficiency both questionnaire results and quantitative task time and errors were analyzed.

The questionnaire results for the efficiency should give an indication of the users' subjective feel towards the efficiency of the techniques. The results show that the circle-menu received highest rated feedback, with significance to all other techniques except the mouse technique. The mouse technique showed significance to the lowest rated GoGo technique and the also low rated pointing technique. From this result a clear unambiguous answer to which technique was perceived the most effective cannot be concluded, however the tendencies point towards the circle-menu technique. The questionnaire data also shows that the GoGo technique received lowest ratings with significance to all other techniques. From this it can be concluded that the GoGo technique was seen as the least effective compared to the other techniques. Also the pointing technique received fairly low responses with significance to all other techniques, concluding that it was also considered as not being very effective. For the lasso technique the responses were very ambiguous with a high variance and no clear tendency.

The quantitative data for the efficiency were measured based on six different tasks. For all tasks the completion time was measured. The results from this quantitative data can show a more nuanced picture of the techniques efficiency, also since the tasks were divided into single and multiple selections. For the single selection tasks the clear and unambiguous winner is the circle-menu technique, outperformed all other techniques in task completion times. The results that were found for multiple selection tasks also showed that the circle-menu technique outperformed all other techniques significantly. The other techniques had all similar task completion times for single selection, so no tendency can be found for these. For multiple selection tasks the pointing technique showed to be significantly slower compared to all other techniques, except the GoGo technique. This gives an indication of the pointing techniques efficiency when many screen objects need to be selected, towards being less effective than the other techniques. A more nuanced picture of the techniques efficiency can be found when looking at the individual tasks. Since three different levels of placement for the boxes were used in the test, it can be seen that these placements also had an influence on the task completion time. In this way was it significantly more difficult to select the box in the first task (placed further away from the camera) when compared to the third task (placed closest to the camera). One can argue that this difficulty can be strongly influenced by the accuracy of the tracking, and thus also favored the circle-menu technique in the harder tasks. When solely looking at the data from the third task where the accuracy of the tracking should not have a big influence, it can be seen that the mouse technique performed equally well with the circle-menu technique. Although we have no quantifiable data on the tracking accuracy we assume that there is a correlation between accuracy and task completion time; a correlation that shows to be important to address in the design of the techniques, particularly through the use of constrains. We will discuss this finding more thoroughly in the third part of the discussion chapter.

Overall it can be concluded from both the measured data and the questionnaire data that the circle-menu technique showed to be the most effective. The ambiguity found in qualitative data and the single selection in the third task may though indicate that the mouse technique could perform equally well, though only in single selection tasks and also dependant on the layout of the objects and assumedly the accuracy of the tracking.

Errors

For the evaluation of errors both measured data and questionnaire data was analyzed. The errors in regard to the usability refers to that a system should have a low error rate, so that users make possibly no errors while using the system; if errors occur users should be able to easily recover from them. Also catastrophic or fatal errors must not occur. In regard to the evaluated selection techniques, the errors refer to errors made during the selection, meaning that wrong objects are selected. Users were also offered the possibility of a catastrophic error in terms of aborting the task if they did not feel able to solve it. However it was also chosen to deliberately not allow the users to recover from errors, since this would influence the efficiency measure and require actions for doing so.

From the data obtained from questionnaires no clear tendency of the technique perceived as being the least error prone can be found. A weak tendency might be seen for the mouse technique however without significance to the other techniques. The only clear result is found in the GoGo technique, which was rated the lowest, meaning that participants did not feel they selected the objects they intended to select with that technique. Here significance is found in relation to all other techniques. The very ambiguous results found in this qualitative data might be due to the feedback offered during the tasks, which did not show the participants if they selected the correct objects. More clear results might be obtained if feedback on the success of the tasks is given.

The quantitative data was measured during all tasks, where the number of wrong selected objects was counted together with the number of catastrophic errors in form of task abortion. For both single and multi selection the pointing and the mouse technique were the most error free techniques. Most errors were done with the lasso and the GoGo technique followed by the circle-menu. The data collected shows that the error rate for these techniques is very high. It has to be said that the reason for most errors might be found in the implementation, which still is a prototype and possible enhancements could and should reduce the errors made with all techniques. For the circle-menu it was noticed during the test that errors occurred due to a too low threshold for the confirmation gesture (push). We think that the error rate for this technique would be reduced with a better gesture detection. For the lasso technique same reasons can be found, however here we think the problem is also caused by the accuracy of the tracking. Since in the lasso technique participants needed to draw their selection, the accuracy influenced this drawing and made it harder to precisely select the intended objects; a correlation which only can be assumed without prove. For the GoGo technique the high number of errors might be ascribed to the general design of the technique, which also showed to be problematic to learn as explained earlier.

Altogether it can be concluded that participants made least errors with the pointing and the mouse technique. Also the data indicates that more work should be spent on reducing the occurrence of errors and avoiding fatal errors in the GoGo, lasso and circle-menu technique.

Satisfaction & Comfort

For the satisfaction and the comfort only questionnaire data was analyzed.

The satisfaction refers to the subjective experience with a technique and it gives an indication of which techniques were liked the most by the users. The results point to the fact that the users rated the circle-menu, the mouse and the lasso techniques equally high. Although no significant difference can be found between these techniques, tendencies suggest that the circle-menu received best satisfaction ratings, followed by the mouse technique. The GoGo technique showed to be significantly unsatisfactory compared to all other techniques. Also the pointing technique received low satisfaction ratings with significant differences compared to all other techniques.

Similar results can be found in the comfort ratings. The comfort ratings show how close to normal users perceived their level of comfort to be while using a technique; since all techniques use arms and hands for selecting the content, the comfort can be influenced. From the results the mouse, circle-menu and lasso technique received equal ratings, close to the normal condition. The GoGo technique was rated significantly lowest, with an average showing that most participants felt moderate discomfort while using this technique. The pointing technique also received significant ratings compared to the other techniques with an average close to moderate discomfort. We believe that the lower ratings in these two techniques might be caused by the way the user has to hold his arm while using both techniques. The GoGo and pointing technique required the user to stretch out his arm, while the other techniques could be used with an angled arm - a posture that puts less strain on the arm. The comfort ratings however only show an immediate effect influenced by the time users spend on the techniques, which was around 15 minutes on all tasks and tutorials. It has to be assumed that longer use of all techniques results in a higher level of discomfort.

28.2 Target group and usability

In order to get closer to an answer to the problem statement of this thesis a discussion of the results importance within the usability is needed. Having described the obtained results in relation to the different parameters constituting to the usability, the important question of how to weight these parameters remains. It is also important to notice that the measured usability results are relative to the user group partaking in the evaluation.

The intention with the study was to evaluate the usability on novice users, which we defined as users without special experience in the use of 3D interaction techniques, especially without having used similar systems as the one tested on. Although it can be argued that the demographical data obtained shows that the user group had a fairly high experience in computer technology and alternative interaction devices (like the Nintendo Wii / Playstation Move), we still assume that the users partaking had no prior experience in the devised ITs or similar methods of interacting with screen content. This assumption is supported by the demographics showing that most participants had no prior experience in 3D tracking devices such as the Kinect.

Having evaluated a novice user group, we argue that the different usability aspects results cannot be given equal weight. The learnability is especially important for the novice user, since the system should allow the user to learn and handle the system easily. Also the Memorability should be weighted high for novice users, since it is important to remember how to control a system in order to be able to return back to using it. The efficiency, on the other hand, normally refers to the expert users steady state of performance, after the learning curve has flattened out. In our case the efficiency therefore only shows a momentary picture of the participants partaking in the evaluation. By this the efficiency must be given less weight when wanting to determine which of the techniques performed best in regards to novice users. The errors and satisfaction however involves and relates to all user groups and should be weighted equally. On the other hand for an expert user group the learnability becomes unimportant since such a group should have used the system so that the learning curve has flattened out, whereas the efficiency becomes especially important.

In regards to the obtained results the tendencies for learnability, memorability and errors would favor the mouse technique as being the one technique best suited for the novice user group. The mouse technique showed to be easy to learn, it had a low error rate and users were satisfied using it. However, the results were not unambiguous in all usability aspects, but overall the mouse technique performed well in all aspects and must therefore be concluded to be the most usable technique for novice users. The circle-menu technique performed overall equally well, except for the error aspects of usability. It can be speculated that a large part of these errors could be avoided by a better implementation and would in such case make the circle-menu technique the overall best suited technique, since it performed better in the less rated efficiency. However, this cannot be concluded based on the results and remains speculative.

At this point it is important to stress out the fact that the described aspects only were compared relative to each other, meaning that the techniques were compared to each other in order to find out which performed best in the different usability aspects. When evaluating the usability of a system normally a set of specific usability goals is used, with at least a minimum level of requirements specified to be reached in order to release a system or product to the public (Nielsen, 1993). Such requirements are normally based on measures of an existing product. In our case such specific goals were not stated as the product is new and no data for comparison existed. Seen in the light of the iterative process utilized in this project such specific goals could be stated and evaluated as the next step in the iteration process. Ultimately the techniques should be tested in a real application to prove their usefulness in context, a goal which we did not set out for from the beginning of this project nonetheless would be interesting to investigate.

28.3 Generalization and design guidelines

This last part of the discussion tries to explain which design factors are important for usable selection techniques using markerless motion tracking. In this part we try to gather all the knowledge learned throughout the project, including the theoretical research and the practical knowledge gained while working with the Kinect camera and the NITE tracking framework for creating the implementation. This knowledge forms the bases for a "best guess" heuristics concerning the parts that are especially important to include in the design of selection techniques using the described hardware; in the end of this section this will be finalized as a list of design guidelines.

Though the discussion presented here is without direct proof, we think that the results indicate - and the knowledge gained throughout the project justifies - the elements presented in this part.

Having concluded that the mouse technique showed to be the overall most usable technique for the described target group, the important question to ask at this point is which of the design elements utilized in this technique constitute to its success? The goal with this techniques design was to resemble the behavior of a regular mouse pointer, as found in most operating systems. This design choice builds on the users knowledge, it allow users to form a mental model of a well known form of interacting with computers. This choice constitutes to both the learnability and the memorability of the technique and it might even satisfy users by giving them what they already know.

Another factor influencing the learnability may be found in the actions required to use a technique. By this we understand the mouse technique uses two actions, moving the hand and pressing with the hand. The equally high rated pointing technique in learnability uses only one action, pointing with the arm / hand. On the other hand the GoGo technique requires four actions, pushing and pulling, navigating the arm and touching the objects; possibly explaining why users had a much more difficult time learning the technique. In general we think that the number of actions required when using a selection technique, should be kept low when learnability is important.

Our point of view is that many aspects concerning the memorability, can be explained by the general way the selection techniques are operated, by using body movements to control the techniques. In this way the techniques involve user's sensory memory, building on users motor skills and by this actively providing memorization through bodily action.

It is hard to determine which design aspects constitute the satisfactory aspects of a technique. This subjective aspect might be influenced by several reasons. The possibly most valuable aspect to consider for the satisfaction is to know the targeted audience well. We also think that a part can be found in the aesthetics of a technique, the visual appeal and the comfort. On the other hand we believe that utilizing principles, like the ones described in the analysis section of this thesis, can play an important role in the users satisfaction towards a technique. The magical principles could for example stimulate a user's sense of imagination, by that possibly contributing to the satisfaction. However, the techniques implemented in this project are not fully utilizing the possibilities of these magical principles. It would be interesting to further investigate these principles and to create techniques that encourage users' imagination.

The efficiency and errors are especially influenced by the input methodology. In order to design effective selection techniques for the Kinect and the used tracking framework, it showed to be important to address tracking inaccuracy through the design. One of the design elements important for allowing effective selection even with inaccurate tracking is the use of constrains and mapping. The circle-menu technique maps in this way the users hand position effectively by constraining the input axis. The outcome is that the technique only depends on the hands x and y position relative to a fixed point, which in turn eliminates a large part of the inaccuracy of the tracking. Similar are constrains utilized to restrict the input for the mouse technique. The other factor important is to consider the distance of motion needed to select objects, like described in Fitts' law. A general advice for overcoming tracking inaccuracy and also to lower the error rate is to design for coarse movement. Most results obtained showed a high error rate and lower efficiency in tasks where fine tuned movement was required. This can largely be avoided by designing both the content (the VE) and the selection techniques to circumvent such movement. Especially the unconstrained techniques like the pointing technique are more prone to such inaccuracy, which contrary to design suggestions in the literature (e.g. (Bowman et al., 2005)) makes them less attractive for selection techniques using markerless motion tracking.

The following list summarizes the usability aspects discussed in a list of design guidelines for selection techniques using markerless motion tracking, like the Kinect and the NITE framework:

Learnability

- Selection techniques that require least possible actions in order to perform a selection are easy to learn
- Utilizing the user knowledge (principles and metaphors) is important for the ease of learning
- Selection techniques that do not require confirmation actions are especially easy to learn

Memorability

- Using the body as control (controller free input) allows users to easier remember selection techniques compared to mediated techniques (controller based input)

Satisfaction

- Detailed knowledge of the target group helps improving the satisfaction with selection techniques

Efficiency and errors

- Constrain the input to overcome tracking inaccuracy
- Avoid fine movement and design for coarse movement to overcome tracking inaccuracy
- Avoid timed confirmation methods when efficiency is more important than learnability
- Utilize Fitts' law in the design of selection techniques
- Avoid selection errors by constraining confirmation actions
- Reliable gesture detection is of great importance to avoid errors

Comfort

- If possible allow users to rest their arms
- Avoid techniques that require outstretched arms

29 Conclusion

In the work of this thesis we investigated the possibility of a cable-free VE, by using the recently released Kinect as input together with the publicly available NITE tracking framework. The focus was set on interaction techniques allowing selection of screen content. The problem addressed was formulated as:

Which body controlled interaction techniques are best suited for single and multiple selection of screen content, using optical markerless motion capture (Kinect and NITE) and which design elements constitute the success of a technique?

Through the theoretical knowledge gained from the literature we developed a taxonomy describing the important aspects of selection techniques. Together with the principles important for VEs we used the theory to developed five interaction techniques representing the major aspects of our taxonomy: an object touch technique, two pointing techniques, a technique using HUD elements and a gesture based technique. These techniques were refined and implemented based on an iterative development process and finally evaluated in a user study addressing the important aspects of the usability, which are: learnability, memorability, efficiency, errors, satisfaction and comfort. Since these aspects are relative to the intended target group, we assessed and weighted the outcome of the study in relation to the evaluated user group, which we defined as novice users. For this target group the study showed that the implemented pointing technique, which we called mouse technique, was the best suited technique for both single and multiple selection of screen content. With this result being relative to our specific implementation and the target group, we tried to outline throughout the discussion which elements are important when designing for optical markerless tracking, as the Kinect and the NITE framework. The outcome was concluded by a list of design guidelines applying to such input.

30 Bibliography

Antle, A.N., Corness, G. & Droumeva, M., 2009. What the body knows: Exploring the benefits of embodied metaphors in hybrid physical digital environments. *Interacting with Computers*, 21, pp.66-75. Special issue: Enactive Interfaces.

Baddeley, A., 1997. *Human Memory, Theory and Practice*. Revised Edition ed. East Essex, United Kingdom: Psychology Press Ltd.

Bowman, D.A., 1998. Interaction Techniques for Immersive Virtual Environments: Design, Evaluation, and Application. *Journal of Visual Languages and Computing*, 10, pp.37-53.

Bowman, D.A., 1999. *Interaction Techniques for Common Tasks in Immersive Virtual Environments*. PhD Thesis. United States of America: Georgia Institute of Technology. A Thesis Presented to The Academic Faculty.

Bowman, D.A. et al., 2006. New Directions in 3D User Interfaces. *International Journal of Virtual Reality*, 05(02), pp.3-14.

Bowman, D.A. & Hodges, L.F., 1999. Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. *Journal of Visual Languages and Computing*, 10, pp.37-53.

Bowman, D.A., Koller, D. & Hodges, L.F., 1997. Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques. In *Virtual Reality Annual International Symposium, 1997., IEEE*. Albuquerque, NM, 1997. IEEE.

Bowman, D.A., Kruijff, E., LaViola, J.J. & Poupyrev, I., 2005. *3D User Interfaces: Theory and Practice*. New York: Addison-Wesley.

Bruder, G., Steinicke, F. & Hinrichs, K.H., 2009. Arch-Explore: A Natural User Interface for Immersive Architectural Walkthroughs. In *3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on*. Lafayette, LA, 2009. IEEE.

Bullinger, H.-J., Bauer, W., Wenzel, G. & Blach, R., 2010. Towards user centred design (UCD) in architecture based on immersive virtual environments. *Computers in Industry*, 61(4), pp.372-79. Human-Centered Computing Systems in Industry - A Special Issue in Honor of Professor G. Salvendy.

Burmeister, O.K., 2000. Usability Testing: Revisiting Informed Consent procedures for testing Internet sites. In *CRPIT '00 Selected papers from the second Australian Institute conference on Computer ethics*. Canberra, 2000. Australian Computer Society.

Buxton, W. et al., 2011. *Human Input to Computer Systems: Theories, Techniques and Technology*. [Online] Available at: http://www.billbuxton.com/inputManuscript.html [Accessed 01 March 2011]. Chapter 7.

Chen, J. & Bowman, D., 2009. Domain-specific design of 3d interaction techniques: An approach for designing useful virtual environment applications. *Presence: Teleoper. Virtual Environ.*, 18(5), pp.370-86.

Chen, J., Pyla, P.S. & Bowman, D.A., 2004. Testbed Evaluation of Navigation and Text Display Techniques in an Information-Rich Virtual Environment. In *Virtual Reality, 2004. Proceedings. IEEE.*, 2004. IEEE.

Choumane, A., Casiez, G. & Grisoni, L., 2010. Buttonless Clicking: Intuitive Select and Pick-release Through Gesture Analysis. *Virtual Reality Conference (VR), 2010 IEEE* , pp.67-70.

Cohen, B.H. & Lea, R.B., 2004. *Essentials of statesitcs for the social and behavioral sciences*. Hoboken, New Jersey, United States of America: John Wiley & Sons.

Denzin, N.K., 2009. *The research act, a theoretical introduction to sociological methods*. First paperback printing ed. United States of America. Was originally printed in 1970.

Dominjon, L. et al., 2005. The "Bubble" Technique: Interacting with Large Virtual Environments Using Haptic Devices with Limited Workspace. In *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint.*, 2005. IEEE.

Drettakis, G., Roussou, M., Reche, A. & Tsingos, N., 2007. Design and Evaluation of a Real-World Virtual Environment for Architecture and Urban Planning. *Presence: Teleoper. Virtual Environ.*, 16(3), pp.318-32.

Encyclopædia Britannica, 2011. *biology, philosophy of.* [Online] Available at: http://www.britannica.com.zorac.aub.aau.dk/EBchecked/topic/681551/biology-philosophy-of/283549/Taxonomy [Accessed 21 April 2011].

Ericsson, K.A. & Kintsch, W., 1995. *Long-Term Working Memory*. technical report. University of Colorado, Boulder.

Eysenck, M.W. & Keane, M.T., 2005. *Cognitive Psychology, a student's handbook*. First Edition ed. East Sussex, United Kingdom: Psychology Press Ltd.

Forsberg, A.S. et al., 2000. Immersive Virtual Reality for Visualizing Flow Through an Artery. In *Visualization 2000. Proceedings*. Salt Lake City, UT, USA , 2000. Brown University, Providence.

Foxlin, E., 2002. Motion Tracking Requirements and Technologies. In K. Stanney, ed. *Handbook of Virtual Environment Technology*. Lawrence Erlbaum Associates. pp.163-210.

Froehlich, B. & Bowman, D., 2009. 3D User Interfaces. *IEEE Computer Graphics and Applications*, Nov./Dec. pp.20-21.

Frøkjær, E., Hertzum, M. & Hornbæk, K., 2000. Measuring Usability: Are Effectiveness, Efficiency, and satisfaction really correlated? In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '00)*. The Hague, Netherlands, 2000. ACM Press.

Gerber, D. & Bechmann, D., 2005. The Spin Menu: a Menu System for Virtual Environments. In *Virtual Reality, 2005. Proceedings. VR 2005. IEEE.*, 2005. IEEE.

Gibson, J.J., 1979. *The ecological approach to visual perception*. Boston: Houghton Mifflin.

Gutiérrez, M., Thalmann, D. & Vexo, F., 2005. Semantic Virtual Environments with Adaptive Multimodal Interfaces. In *Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International.*, 2005. IEEE.

Hirose, N., 2002. An ecological approach to embodiment and cognition. *Cognitive Systems Research*, 3, pp.289-99.

Hoang, T.N., Porter, S.R. & Thomas, B.H., 2009. Augmenting Image Plane AR 3D Interactions for Wearable Computers. In *AUIC '09: Proceedings of the Tenth Australasian Conference on User Interfaces.*, 2009. Australian Computer Society, Inc..

Igarashi, T., Matsuoka, S. & Tanaka, H., 2007. Teddy: A Sketching Interface for 3D Freeform Design. In *SIGGRAPH '07 ACM SIGGRAPH 2007 courses.*, 2007. ACM.

Jacob, R.J. et al., 2008. Reality-based interaction: a framework for post-WIMP interfaces. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI '08).* Florence, Italy, 2008. ACM.

Kinect Games, X.3., 2010. *Xbox 360 Kinect Games.* [Online] Available at: http://www.xbox.com/en-US/Kinect/Games [Accessed May 2011].

Kinect, M., 2010. *Kinect.* [Online] Available at: http://www.xbox.com/en-US/kinect [Accessed May 2011].

Klatzky, R.L. & Lederman, S.j., 2003. Touch. *Expermental Psychology*, 4, pp.147-76.

Kulik, A., 2009. Building on realism and magic for designing 3D interaction techniques. *IEEE Comput. Graph. Appl.*, 29(6), pp.22-33.

LaViola, J. et al., 2009. CHI 2009 Course Notes: User Studies and 3D UIs. In *3D User Interfaces: Design, Implementation, Usability - CHI'09.*, 2009. ACM.

Lee, M. & Billinghurst, M., 2008. A Wizard of Oz Study for an AR Multimodal Interface. In *ICMI '08 Proceedings of the 10th international conference on Multimodal interfaces.*, 2008. ACM.

Liang, J. & Green, M., 1994. JDCAD: A Highly Interactive 3D Modeling System. *Computers & Graphics*, pp.499-506.

Lindeman, R.W., 1999. *Bimanual Interaction, Passive-Haptic Feedback, 3D Widget Representation, and Simulated Surface Constraints for Interaction in Immersive Virtual Environments*. The School of Engineering and Applied Science of The George Washington University. ph.d. thesis.

Lindeman, R.W., Sibert, J.L. & Hahn, J.K., 1999. Towards usable VR: an empirical study of user interfaces for immersive virtual environments. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit.*, 1999. ACM.

Liu, L., Martens, J.-B. & Liere, R.v., 2010. Revisiting Path Steering for 3D Manipulation Tasks. *International Journal of Human-Computer Studies*, 69(03), pp.170-81.

MacKenzie, I.S., 1992. Fitts' Law as a Research and Design Tool in Human-Computer Interaction. *Human-Computer Interaction*, 7(1), pp.91-139.

Maguire, M., 2001. Methods to support human-centred design. *International Journal of Human-Computer Studies*, 55(4), pp.587-634.

Maier, P. et al., 2010. What Do You Do When Two Hands Are Not Enough? Interactive Selection of Bonds Between Pairs of Tangible Molecules. *3D User Interfaces (3DUI), 2010 IEEE Symposium on* , pp.83-90.

Martens, J.-B. et al., 2004. Experiencing 3D Interactions in Virtual Reality and Augmented Reality. *EUSAI '04: Proceedings of the 2nd European Union symposium on Ambient intelligence* .

McGurk, H. & MacDonald, J., 1976. hearing lips and seing voices. *Nature*, 264, pp.746-48.

McTear, M.F., 2001. Spoken Dialogue Technology: Enabling the Conversational User Interface. *ACM Computing Surveys (CSUR)*, 34(1), pp.90-169.

Miller, G.A., 1955. The Magical Number Seven, Plus or Minus Two Some Limits on Our Capacity for Processing Information. *Psychological Review*, 101(02), pp.343-52.

Mine, M.R., 1995. *Virtual Environment Interaction Techniques*. Report TR95-018. University of North Carolina. Computer Science Technical Report.

Nielsen, J., 1993. *Usability Engineering*. San Francisco: Morgen Kaufmann.

Nielsen, J., 2000. *Why You Only Need to Test With Five Users*. [Online] Available at: http://www.useit.com/alertbox/20000319.html [Accessed 21 April 2011].

Nielsen, J., 2005. *Ten Usability Heuristics*. [Online] Available at: http://www.useit.com/papers/heuristic/heuristic_list.html [Accessed 15 March 2011].

Norman, D.A., 1990. *The Design of Everyday Things*. New York: Basic Books.

Nvidia, 2011. *Nvidia 3D Vision*. [Online] Available at: http://www.nvidia.com/object/product_GeForce_3D_VisionKit_us.html [Accessed May 2011].

OpenNI, 2010. *Documentation*. [Online] Available at: http://www.openni.org/documentation [Accessed 12 May 2011].

OpenNI, 2011. *OpenNI*. [Online] Available at: http://www.openni.org [Accessed May 2011].

Pai, D.K., 2005. Multisensory Interaction: Real and Virtual. *Springer Tracts in Advanced Robotics*, 15, pp.489-98.

Peck, T.C., Fuchs, H. & Whitton, M.C., 2010. Improved Redirection with Distractors: A Large-Scale-Real-Walking Locomotion Interface and its Effect on Navigation in Virtual Environments. In *Virtual Reality Conference (VR), 2010 IEEE*. Waltham, MA, 2010. IEEE.

Phillips, L., Ries, B., Kaeding, M. & Interrante, V., 2010. Avatar self-embodiment enhances distance perception accuracy in non-photorealistic immersive virtual environments. In *Virtual Reality Conference (VR)*. Waltham, MA , 2010. IEEE.

Pierce, J.S. et al., 1997. Image Plane Interaction Techniques In 3D Immersive Environments. *I3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics* , April. pp.39-43.

Pierce, J.S., Matthew, C., Dantzich, M.v. & Robertson, G., 1999. Toolspaces And Glances: Storing, Accessing, And Retrieving Objects In 3D Desktop Applications. In *I3D '99 Proceedings of the 1999 symposium on Interactive 3D graphics.*, 1999. ACM.

Pierce, J.S., Steams, B.C. & Pausch, R., 1999. Voodoo Dolls: Seamless Interaction at Multiple Scales in Virtual Environments. In *I3D '99 Proceedings of the 1999 symposium on Interactive 3D graphics.*, 1999. ACM.

Poupyrev, I., Billinghurst, M., Weghorst, S. & Ichikawa, T., 1996. The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp.78-80.

PrimeSense Ltd., 2010. *Reference Design*. [Online] Available at: http://www.primesense.com/?p=514 [Accessed 25 February 2011].

PrimeSense, 2011. *NITE Algorithms 1.3 (User Manual)*. [Online] Available at: http://www.primesense.com [Accessed May 2011].

PrimeSense, 2011. *PrimeSense*. [Online] Available at: http://www.primesense.com/?p=515 [Accessed May 2011].

PrimeSensor, 2011. *PrimeSense - PrimeSensor*. [Online] Available at: http://www.primesense.com/?p=514 [Accessed May 2011].

Rademacher, P.M., 2002. *Measuring the Perceived Visual Realism of Images*. Ph.d. Chapel Hill: University of North Carolina.

Ray, A.A., 2008. *The Interaction Framework For Innovation: A Method to Create Reusable Three-Dimensional Interaction Techniques*. Blacksburg, VA: Virginia Polytechnic Institute and State University. ph.d. thesis.

Sarter, N.B., 2006. Multimodal information presentation: Design guidance and research challenges. *International Journal of Industrial Ergonomics* , 36, pp.439-45.

Sato, S. & Salvador, T., 1999. Playacting and Focus Troupes: Theater techniques for creating quick, Intense, immersive, and engaging focus Group sessions. *Interactions - New York*, 6(5), pp.35-41.

Schuchardt, P. & Bowman, D., 2007. The benefits of immersion for spatial understanding of complex underground cave systems. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology (VRST '07)*. Newport Beach, California, 2007. ACM.

Shneiderman, B., 1983. Direct manipulation: a step beyond programming languages. *IEEE Computer*, 16(8), pp.57-69.

Shneiderman, B., 2003. Why Not Make Interfaces Better than 3D Reality? *IEEE Comput. Graph. Appl.*, 23(6), pp.12-15.

Shneiderman, B., 2005. Leonardo's laptop: human needs and the new computing technologies. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. Bremen, Germany, 2005. ACM.

Shotton, J. et al., 2011. *Microsoft Research*. [Online] Available at: http://research.microsoft.com/apps/pubs/default.aspx?id=145347 [Accessed 12 May 2011].

Slater, M. & Steed, A., 2000. A Virtual Presence Counter. *Presence: Teleoper. Virtual Environ.*, 9(5), pp.413-34.

Society for Technical Communication, 2011. *Usability Techniques Methods for Successful "Thinking Out Loud" Procedure*. [Online] Available at: http://www.stcsig.org/usability/topics/articles/tt-think_outloud_proc.html [Accessed 14 April 2011].

Stoakley, R., Conway, M.J. & Pausch, R., 1995. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *CHI '95 Proceedings of the SIGCHI conference on Human factors in computing systems.*, 1995. ACM.

Suma, E.A. et al., 2010. Effects of Travel Technique and Gender on a Divided Attention Task in a Virtual Environment. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on.*, 2010. IEEE.

Terziman, L. et al., 2010. Shake-Your-Head: Revisiting Walking-In-Place for Desktop Virtual Reality. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*. Hong Kong, 2010. ACM.

Tullis, T. & Albert, B., 2008. *Measuring the user experience*. Burlington, United states of America: Morgan Kaufmann.

Unity3D, 2011. *http://unity3d.com*. [Online] [Accessed May 2011].

Valkov, D., Steinicke, F., Bruder, G. & Hinrichs, K., 2010. A multi-touch enabled human-transporter metaphor for virtual 3D traveling. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*. Waltham, MA , 2010. IEEE.

Watson, B., Spaulding, V., Walker, N. & Ribarsky, W., 1997. Evaluation of the effects of frame time variation on VR task performance. In *Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97)*. Washinton DC, 1997. IEEE Computer Society.

Welch, G. & Foxlin, E., 2002. Motion Tracking: No Silver Bullet, but a Respectable Arsenal. *Computer Graphics and Applications, IEEE* , 22(6), pp.24 - 38.

Wendt, J.D., Whitton, M.C. & Brooks, F.P., 2010. GUD WIP: Gait-Understanding-Driven Walking-In-Place. In *Virtual Reality Conference (VR), 2010 IEEE*. Waltham, Massachusetts, USA, 2010. IEEE.

White, S., Feng, D. & Feiner, S., 2009. Interaction and Presentation Techniques for Shake Menus in Tangible Augmented Reality. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on.*, 2009. IEEE.

Wingrave, C.A., 2008. *Concept-Oriented Design in Chasm: Conversational Domain Language Inspired 3D User Interface Design and Development*. Blacksburg, VA: Virginia Polytechnic Institute and State University. ph.d. thesis.

Wingrave, C.A., Bowman, D.A. & Ramakrishnan, N., 2002. Towards Preferences in Virtual Environment Interfaces. In *EGVE '02 Proceedings of the workshop on Virtual environments.*, 2002. Eurographics Association Aire-la-Ville.

Wixon, D., 2008. *UX Week - The Challenge of Emotional Innovation*. [Online video: http://vimeo.com/2893051] Available at: http://vimeo.com/2893051 [Accessed May 2011]. Citation can be found at 7.31 in the video.

Wyss, H.P., Blach, R. & Bues, M., 2006. iSith – Intersection-based Spatial Interaction for Two Hands. *3D User Interfaces, 2006. 3DUI 2006. IEEE Symposium on* , pp.59-61.

APPENDIX

I. Heuristic evaluation

The authors were conducting the evaluation, and the purpose of the heuristic evaluation, using guidelines from Nielsen (2005), was to construct usability friendly selection techniques from the proposed selection model. Nielsen's design guidelines constituted a list of ten subjects, but due to the prospect of constructing prototypes in order to conduct the test to prove the stated problem, three subjects were not taken in to consideration. Hence the total list consisted of seven subjects:

- **Visibility of system status**
- **Match between system and the real world**
- **User control and freedom**
- **Consistency and standards**
- **Error prevention**
- **Recognition rather than recall**
- **Aesthetic and minimalist design** (Nielsen, 2005)

The techniques that were chosen to undergo the heuristic evaluation were respective represent all parts of the selection model, and the results of this evaluation was the source of later design considerations forming the final selection of techniques to confirm of deny the stated problem.

a) Design considerations

The design will be presented in a structure that constitutes of a combination of Nielsen's design guidelines in regards to the chosen techniques that represent the following aspects of the selection model:

- o A technique using direct selection – by object touch
- o A technique using indirect selection – by a pointing method
- o A technique using indirect selection – by gestures
- o A technique using indirect selection – by a HUD element

General notes

Take an overall consideration towards the selection techniques in the VE.

- *Visibility of system status*
  - o The user should be able to see where his selection tool is in the scene, hence a mapping of the real user and the representation should be logical for the user.
  - o Receiving feedback when indicating and confirming the object selection with the given technique should give a fast and logical feedback.

- *Match between system and the real world*
    - The communication that the system is having with the user should be on the users knowledge level. Meaning if the user is making a mistake, he should be informed about his mistake through visual or textural information that is in his vocabulary.
    - The logical order that is situated in each of the selection techniques is as following:
        - The user makes and indication of his selection.
            - The system gives a feedback informing him if it is the right or wrong.
        - The user the makes a confirmation of his indication of selection.
            - The system gives a feedback to the user if the confirmation is registered or not.
                - Finally the object is underlined in a manner that the user is sure of his selection.
- *User control and freedom*
    - When the user is making a selection mistake in the system, he should be to unselect the object in order to select the right object. Here the undo and redo action should be possible.
- *Consistency and standards*
    - The selection tool should be consistent in its representation to the user. Meaning if choosing a "reality based interaction" or "imagination based interaction" as a mean of utilizing direct touch in the VE scene, the user is confined to the boundaries that are in that domain.
- *Error prevention*
    - In order to prevent errors in the selection process of the user, the limitations of the chosen platform and VE setup need to be known.
- *Recognition rather than recall*
    - In order to have an "intuitive" design, where the user does not have to think about his actions, one method can be to use conventions that are related to the platform, or to the design principles the selection techniques has its origin.
- *Aesthetic and minimalist design*
    - Since the feedback only will be visual, there can be limitation amount of information the user is presented; to avoid confusion the visual feedback should be as direct and simple as possible.

GoGo technique (direct touch)

- *Visibility of system status*
  - The feedback that the user receives from the system should be corresponding with the notion of user being able to touch the objects with a representation of him in the virtual environment.
- *Match between system and the real world*
  - The logical order is as following:
    - The user makes and indication of his selection by pseudo physically touching the object.
      - The system gives a feedback informing him if it is the right or wrong.
    - The user the makes a confirmation of his indication of selection.
      - The system gives a feedback to the user if the confirmation is registered or not.
        - Finally the object is underlined in a manner that the user is sure of his selection.
- *User control and freedom*
  - When the user is making a selection mistake in the system, he should be to unselect the object in order to select the right object. Here the undo and redo action should be corresponding with the notion of being physically in contact with the object.
- *Consistency and standards*
  - The selection tool should be consistent in its representation to the user. Meaning if choosing a "reality based interaction" as a mean of utilizing direct touch in the VE scene, the user is confined to the boundaries that are in that domain.
- *Error prevention*
  - Platform limitations for direct selection are that the user is not physically touching the objects, rather pseudo physically touching. This can result in a lower degree of disbelief, the user feel not embodied in the VE.
  - The setup limitations are related to the fact that the user will not be able to navigate in the scene, due to the focus on selection techniques, and not navigation.
- *Recognition rather than recall*
  - The actions the user is performing in order to conduct a selection will build on the notion of physically interacting with the objects.
- *Aesthetic and minimalist design*
  - The design should support feedback when the subject is directly touching the object, and a direct touch related feedback when the user has confirmed his selection.
  - The user should be able to know where he is in the VE scene; it can be through a representation of the user (to support the notion of direct touch) or a symbolic representation of the selection tool.

Pointing technique (pointing) and Lasso technique (gesture)

- *Visibility of system status*
  - The feedback that the user receives from the system should be corresponding with the notion of making the gesture of pointing.
  - The feedback will be directed by the design principles of "imaginary based interaction".
- *Match between system and the real world*
  - The logical order is as following:
    - The user makes and indication of his selection by pointing at the object.
      - The system gives a feedback informing him if it is the right or wrong.
    - The user the makes a confirmation of his indication of selection.
      - The system gives a feedback to the user if the confirmation is registered or not.
        - Finally the object is underlined in a manner that the user is sure of his selection.
- *User control and freedom*
  - Here the undo and redo action should be corresponding with the notion of not being physically in contact with the object.
- *Consistency and standards*
  - The selection tool should be consistent in its representation to the user. Meaning if choosing a "imaginary based interaction" as a mean of utilizing direct touch in the VE scene, the user is confined to the boundaries that are in that domain.
- *Error prevention*
  - Platform limitations for pointing are not the same as for direct touch, since pointing is an indirect touch action. The user can be confined to one location, without navigating in the VE scene.
  - The setup limitations can be the placement of objects (the risk of objects occluding each other) can be a limitation if the user in confined to not being able to do any form of navigation.
- *Recognition rather than recall*
  - The actions the user is performing in order to conduct a selection will build on the notion of non-physical interacting with the objects.
- *Aesthetic and minimalist design*
  - The design should support feedback when the subject is not directly touching the object, and a non-direct touch related feedback when the user has confirmed his selection.
  - The user should be able to know where he is in the VE scene; it can be through a symbolic representation of the selection tool.

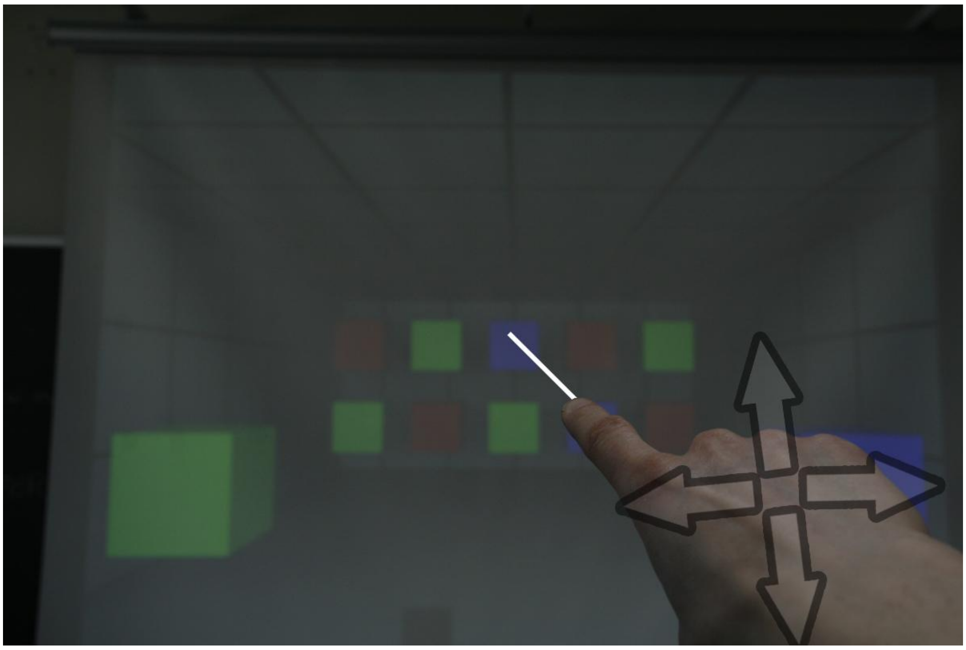
Circle-menu (HUD)

The Circle-menu differs from the other techniques by the fact that in order to make a selection a mediated representation of the virtual environment is presented to the user, this can be through a list, menu etc.

- *Visibility of system status*
    - The feedback that the user receives from the system can be based on the WIMP principles.
    - The feedback will be directed by the design principles of "imaginary based interaction".
- *Match between system and the real world*
    - The logical order is as following:
        - The user makes his object indication of his selection by indicating a representation of the object on a list.
            - The system gives a feedback informing him if it is the right or wrong.
        - The user the makes a confirmation of his indication of selection.
            - The system gives a feedback to the user if the confirmation is registered or not.
                - Finally the object is underlined in a manner that the user is sure of his selection.
- *User control and freedom*
    - Here the undo and redo action should be corresponding with the notion of not being physically in contact with the object.
- *Consistency and standards*
    - The selection tool should be consistent in its representation to the user. Meaning if choosing a WIMP design principles as a mean of utilizing non-direct touch in the VE scene, the user is confined to the boundaries that are in that domain.
- *Error prevention*
    - Platform limitations for pointing are not the same as for direct touch, since HUD is an indirect touch action. The user can be confined to one location, without navigating in the VE scene.
- *Recognition rather than recall*
    - The actions the user is performing in order to conduct a selection will build on the notion of WIMP interaction in a VE.
- *Aesthetic and minimalist design*
    - The design should support feedback when the subject is not directly touching the object, and a non-direct touch related feedback when the user has confirmed his selection.
    - The user should be able to know where the objects are in the VE scene; furthermore the WIMP menu representations of the objects should be clearly understood by the users.

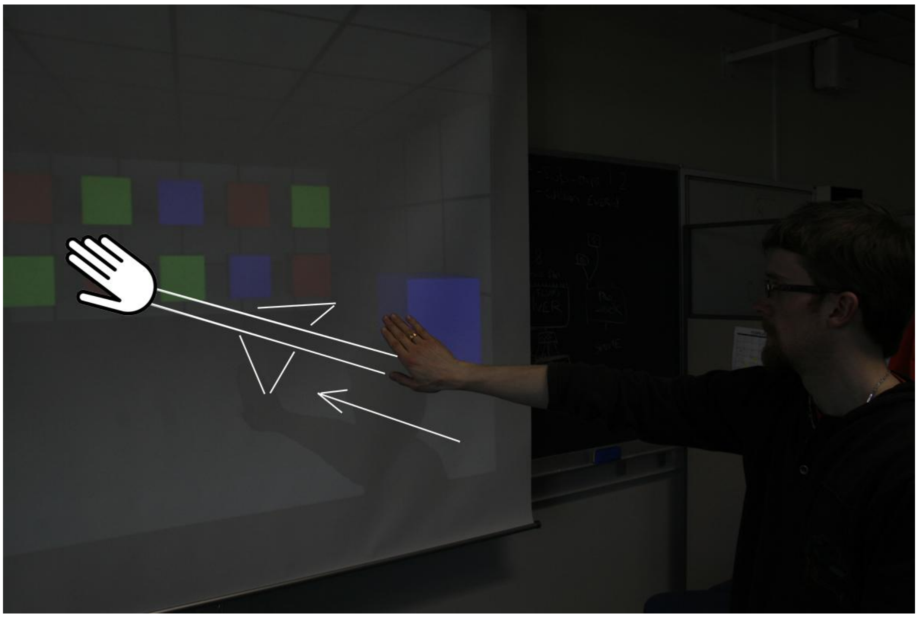
b) Results

Pointing technique



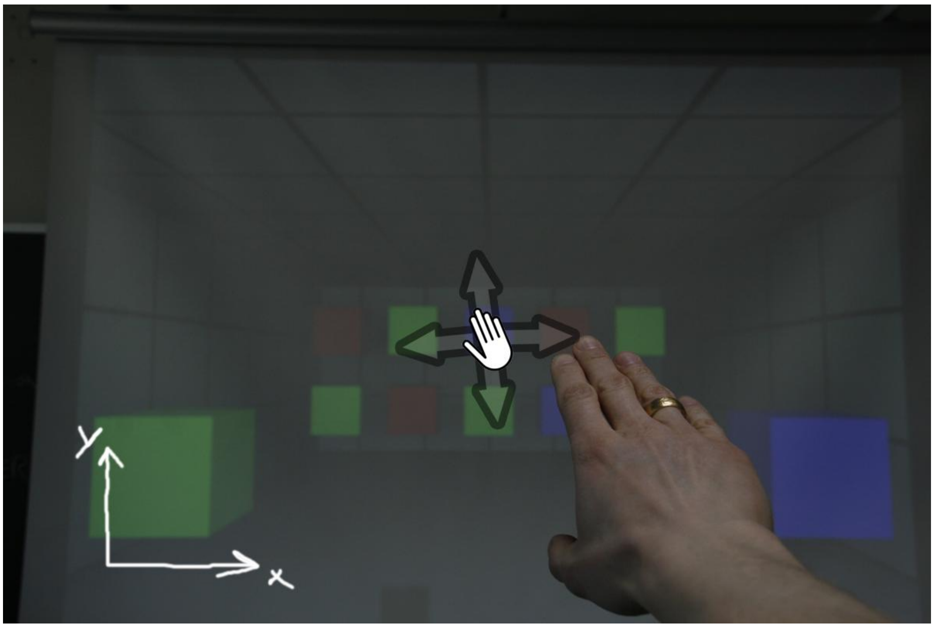
Initial concept of the Pointing technique: The subject points towards the box and there will cast a ray from his hand to indicate where he pointing towards. The user navigates in a two-dimensional space; up-down (y-axis) and right-left (x-axis).

GoGo technique



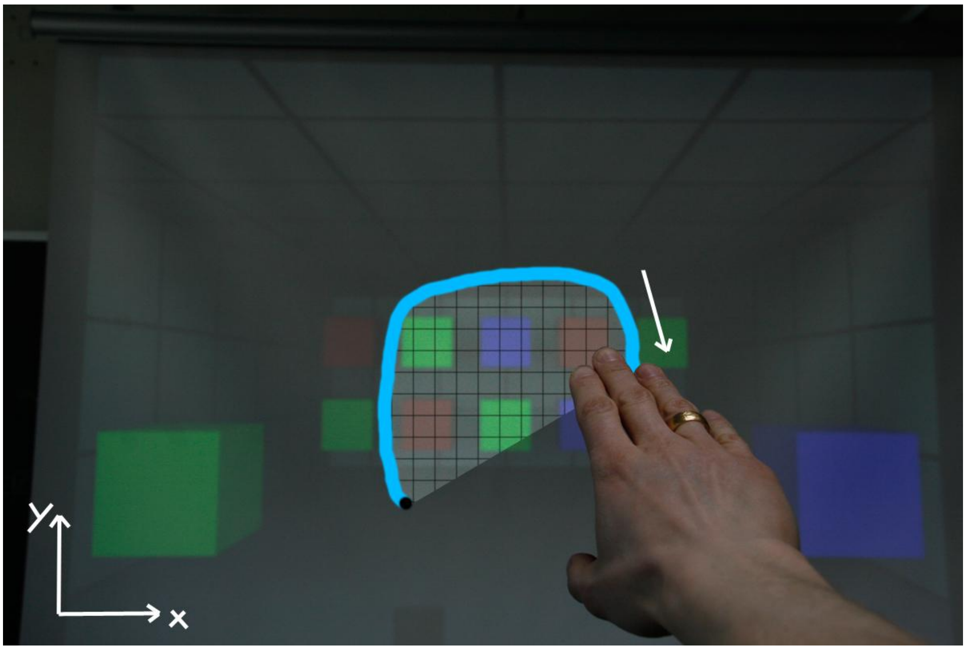
Initial concept for the GoGo technique: here the subject is controlling a hand that is directly touching the object to make a selection. The user controls the hand in a three-dimensional space, right-left (x-axis), up-down (y-axis) and in-out (z-axis).

Mouse Technique



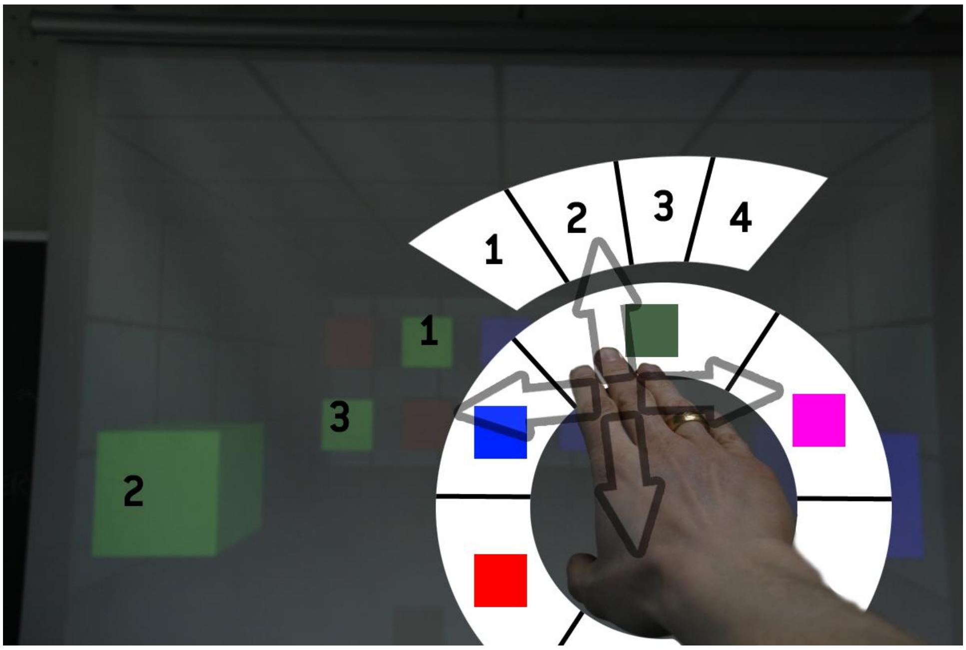
Initial concept of the Mouse technique. The user navigates a hand over a box to make a selection. The navigation is conducted in a two-dimensional space; up-down (y-axis) and right-left (x-axis).

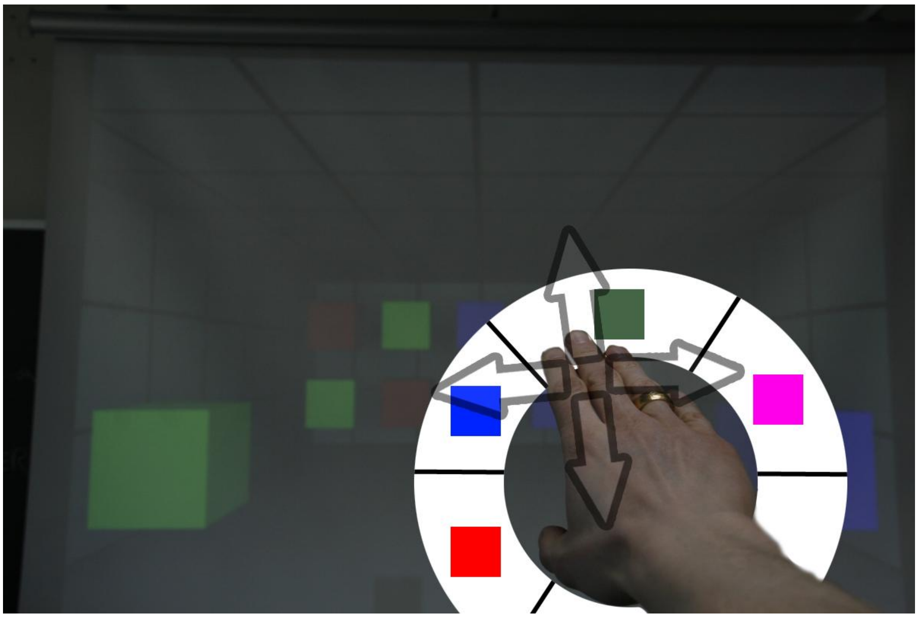
Lasso technique



Initial concept for the Lasso technique: the user selects boxes by drawing a path round the intended box. The navigation of path drawing is conducted in a two-dimensional space, up-down (y-axis) and right-left (x-axis).

Circle-menu





Initial concepts for the Circle-menu in respectively single and multiple selection mode, the user navigates on a menu to select the boxes, in single selection mode the user moves his hand over the number on the menu that corresponds to the box he wants to select. In multiple selection mode the user navigates over the color that corresponds to the boxes he wants to select. The navigation is conducted on a two-dimensional space. Right-left (x-axis) and up-down (y-axis).

II. Formative evaluation

Part of the development procedure of the selection techniques was to conduct formative usability testing in order to detect potential weaknesses in the design and implementation of the technique. The methodology of the test was a so-called discount usability test, introduced first by Nielsen, claiming that the most cost effective number of test subjects in a usability test is five. According to Nielsen 75% of the flaws in the system will be detected by this small number of test subjects (Nielsen, 2000). The test subjects were 4th and 10th Medialogy students at Aalborg University Copenhagen.

To support the use of few test subjects, a qualitative test method was utilized, the method of "thinking aloud" according to Nielsen, it strengths is the wealth of qualitative data that can be obtained from a small amount of users, on the other hand the disadvantage is that is does not lend itself well among most types of performance measurements (Nielsen, 1993).

The test procedure was constructed in such manner to ensure that the test conductor did not bias the results. In order to ensure this STC has developed a usability toolkit, whereas guidelines for thinking aloud (Society for Technical Communication, 2011) testing are part of it, these guidelines were utilized and prompting for answers was a part of our procedure.

Test procedure:

- Introduce the test subject to the purpose of the test, and the method of thinking aloud.
- The subject is asked to try out five interaction technique in conjunction with a series of tasks.
- While trying out the interaction technique, the subject is asked to think out lout her thoughts about she were progressing in the task, prompted for answers concerning the reason for her actions.
- Each of the techniques was introduced in random order, to ensure minimum of amount of biasing of the other techniques.
- There were a total of two test conductors, one moderator and one documenting the subject's thoughts about each of the techniques.
- The subject's feedback was documented by one of the test conductors writing notes.
- The whole test procedure was lasting approximately 15-20 minutes, depending on the subject.
- There were a total of six subjects participating in the test.

Test setup:

Due to the nature of how the Virtual Environment was presented to the test subject. Using a back-projection to visually introduce the environment and having the subject wearing NVIDIA® 3D Vision™ glasses. The setup constituted of having the test subject in front of the screen, trying out the techniques, the moderator sitting to the subjects left, and the documenter to sit behind the screen, controlling the environment and documenting the event.

a) Results

General notes:

- Mapping is not corresponding to the user arm position, meaning when the user is pointing with a pointer at virtual object the finger is not pointing at the corresponding area.
- The calibration figure symbol was confusing for the subjects, due to the wrong position of the figures arms.
- Subjects were suggesting having objects that invited to make the different confirmation action that each of the techniques had, such as a push button, boxes to be grabbed etc.
- Subjects suggested having relative small actions to conduct the confirmation, small arm movements, hand gestures and movements etc.

Pointing technique

- The accuracy of the pointing is better closer to the center of the screen the user is pointing.
- Pointing at the box in the back is straining on the arm, due to the concentration that is needed to make the selection
  - One alternative was to have the arm down at the side while pointing was more confortable, but was not considered as "natural" as normal pointing gesture.
- Users were making different hand gestures to make the confirmation; hence they did not initially understand that there was a time action to confirm the selection.
  - Subjects preferred to make small hand gesture such as mouse click actions

- The light yellow representation of the ray was disappearing against the white background.
- There was noticed a reaction delay, the pointer on the screen was not following the actions made with the hand. But this was not considered as a significant disturbance.

GoGo technique

- Using the dominant hand for pointing and the other for confirmation has the advantage that the selection becomes more accurate (no drifting issues).
  - The confirmation action is not affecting the accuracy of the selection.
- The disadvantage is that the action did not feel “natural”, one needed to think about the action, not considered intuitive.
- There was noticed a reaction delay, the pointer on the screen was not following the actions made with the hand. But this was not considered as a significant disturbance.

- It was easier to make selection of objects that were in the foreground, than in the background, due to the accuracy of the tool.

Mouse technique

- Accuracy of the selection tool is not good further from the center of the screen the subject is selecting objects.
    - Having the arm stretched out was more accurate.
- Having the arm stretched out in order to conduct a selection, was causing discomfort for the users.
- Current push confirmation has following disadvantages
    - When having the arm stretched out, it is hard to make the confirmation.
    - Having the same arm for selection and confirmation, causes the selection of the object to be difficult, due to drifting of the selection, when making the push gesture.
- Suggested solutions for this push confirmation problem were:
    - Move the body forward in order to make the confirmation.
    - Have a gesture that requires minimum amount of movement.
        - Open-close hand.
        - Mouse click action.

- Having the arm stretched out in order to conduct a selection, was causing discomfort for the users.
- Due to the symbolic hand representation on the screen the user felt like making a push action to perform the confirmation.
- There was discovered a occlusion problem; when the user wanted to make a top screen selection of a box, his arm was occluding his view.

Circle-menu

- The feedback when selecting the object confuses the subject.
    - The choice of color is not good.
- The menu is difficult to use
    - Having the dominant hand to select and confirm the selection.
    - One option suggested was using both hands, one for selection, the other for confirmation.
- The activation action in order to start the menu, was considered as a good alternative.
    - But it was suggested a better feedback was needed in order to remember and understand the action, such as an animation of the menu coming in to the scene from the swiping direction.
    - One suggestion was to instead of using swipe right and left to activate and deactivate the menu, was to use a circular action, due to the circular menu.

  - Furthermore it was suggested to have an swipe action that is not accidently activated while interacting with the menu.
- Due to missing feedback of the selection, subjects were struggling to make a connection between the menu and the corresponding boxes.
  - The focus therefore more directed to the menu, than on the boxes that needed to be selected.
  - One suggestion was to have the menu as a line instead of the circle that was occluding the scene.
- Subjects were questioning the use of menu in a relative simple Virtual Environment.

Lasso technique

- The tool activation and confirmation action was confusing to some of the subjects.
  - The user suggested using a throw action to confirm the action, an alternative to the current pull action needed to confirm.
  - Subjects were more open for the idea of having the dominating hand for making the activation, selection and confirmation, to the current two-hand action needed to make the actions.
  - The push activation and pull confirmation action were conflicting actions, hence the subjects were making errors.
  - Small gestures such as mouse click.

- The tool was quickly understood due to the name reference to the lasso tool in Adobe Photoshop.
- The feedback trail when making the action was dissolving to quickly; the user did not know where he had made the selection.
- The lack of feedback to indicate the technique was activated confused the subjects.
  - The feedback was occluded by the cursor on the screen.
- The tool was considered as a multiple selection tool, single selection with this tool was questioned due to the action needed to make the selection.

a) Design guidelines

From the results the following design guidelines were constructed:

General observations / Design guidelines

- When using a pointing technique, people prefer that it is mapped 1to1 so that the thing on the screen they are pointing at with their real finger matches the internal representation
    - People can however use systems also without this 1to1 mapping. The requirements for this are that good feedback is provided, so that user's mental model matches the conceptual. For hand and body interaction the arm/body needs to be represented when not mapped 1to1.
- Pointing techniques afford users to stretch their arms (when pointing) which can result in discomfort.
- Pointing at small targets can be difficult, due to tracking inaccuracy, and the effort required to hold the arm steady results quickly in discomfort.
    - When small objects need to be indicated, either use indirect selection or consider stabilizing the indication by computer aid (programmed behavior which estimates the indicated target and stabilizes the dynamic feedback / and / filters to reduce noise from the skeleton tracker).
- Confirmation gestures for pointing techniques need to consider that the arm might already be stretched, so that a "push" gesture would require the user to move the arm back first, which can result in selection errors and makes the gesture ineffective.
- Push or pull confirmation gestures can result in losing the indicated target while performing the confirmation.
- The most secure confirmation is to use timed confirmation (holding hand over object for X seconds). Advanced user might not prefer the wait time when confirming.
- Using one hand for indicating a selection and the second hand for confirmation eases the usage of "push" or "pull" gestures, but is not regarded as an intuitive action.
- When using indirect selection through a menu there must be a connection between items in the menu and their corresponding objects in the 3d world. E.g. having feedback for the selection on the menu and in the 3d world.
- When using confirmation gestures like left or right- "swipes" (among others) users prefer to have feedback matching the gesture (e.g. left-swipe causes an object to fly on or off the screen, matching to the hand motion)
- As there might not be any intuitive gestures for confirming selection, the most obvious solution is to use gestures that match the used techniques visual feedback. (e.g. a radial menu is opened with a radial motion, pointing at an object is confirmed by pushing on it – and can further be supported by an object affording to push (button))
- When using a drawing technique (like lasso selection) the feedback from drawing should be visible and not fade out or disappear before finishing the selection.

- Techniques using the same hand for indication and confirmation are preferred over the techniques using one hand for indication and another for confirmation.
- If possible users prefer to use finger gestures for confirming a selection (e.g. a mouse-click like motion or closing the hand in a grab motion etc.)
- The feedback used for representing the input can support techniques affordances. (e.g. a flat hand affords pushing etc)
- When using HUD / menu selection, be careful when placing the HUD elements. Make use of transparency or place them on the sides so the user still can see the 3d world. Indicating selected items parallel in the 3d world and on the HUD element might ease the understanding for the user.
- Consider the mapping of input to output, there are situations where users might block their view on the screen with their own hands (occlusion). This can happen when using screen or projections and is avoided when using HMDs.
- When a technique requires a confirmation (gesture) to start the technique, feedback needs to be displayed for novice users.
- Magical elements (like our lasso technique) encourage the user's imagination and might provide affordances where none exists.
- Confirmations using small gestures like "push" or "pull" or swipes, should require minimum possible effort to perform the gesture (little easy movements)
- Indirect selection via HUD elements should preferably be used when having large amount of data to select among, in order to ease the selection. In scenarios with small amount of data it is often easier to use one of the other techniques.
- Selecting single objects should require minimum effort, preferable a simple technique. Multiple selection techniques can be more advanced but should also require minimum effort. Using a complex technique designed for multiple selections in a single selection scenario might be overkill (too much effort required).

III. Summative evaluation tools

a) Questionnaire

The participants filled in an online questionnaire, located:
https://spreadsheets.google.com/viewform?hl=en&formkey=dGRmYjJ4bVBHbHc5LWloU01RbmZaQnc6MA#gid=0

Questions concerning demographic information

I. Participant Information

Gender:

☐ Female
☐ Male

Age: ___________

Dominant hand:
Which hand is your dominant hand?

☐ Right Handed
☐ Left-handed
☐ Mixed-handed

II. Computer Experience

How often do you use a computer? (Choose the best matching statement)

Daily	A few times a week	A few times a month	Rarely or never
☐	☐	☐	☐

How much experience do you have with one of the following types of computer interfaces?

	Never used	1 - 6 months	6 - 12 months	1 - 3 years	3+ years
Windows/Mac/Linux	☐	☐	☐	☐	☐

Which of the following input devices are you familiar with? (Check all that apply)

☐ Keyboard / mouse
☐ Joystick
☐ Touch Screen (iPhone / iPad / android / tabletops / etc…)
☐ Pen / Stylus / Drawing tablet
☐ 3D tracking devices (marker tracking, Kinect, other 3d trackers)
☐ Wii / Playstation Move
☐ Other:_________

If familiar with any of the above mentioned devices, please rate your experience with these devices

	Daily	A few times a week	A few times a month	Rarely or never
Keyboard / mouse	☐	☐	☐	☐
Joystick	☐	☐	☐	☐
Touch Screen (iPhone / iPad / android / tabletops / etc…)	☐	☐	☐	☐
Pen / Stylus / Drawing tablet	☐	☐	☐	☐
3D tracking devices (marker tracking, Kinect, other 3d trackers)	☐	☐	☐	☐
Wii / Playstation Move	☐	☐	☐	☐

Have you ever experienced a stereoscopic display (e.g. 3-D movie, Head-mounted display, etc.)? *

☐ Yes
☐ No

Are you familiar with 3D interaction techniques (as f.ex. used in Virtual Reality or Virtual Environments)?
If yes, please state in which in the answer "Other"

☐ Yes
☐ No
☐ Other:_________

Questions concerning each individual selection technique

III. To each individual technique

1.1 Do you remember this technique?

- ☐ Yes
- ☐ No

If yes, please describe the technique

1.2 Rate how comfortable it was to use the technique

Normal	Moderate Discomfort	Extreme Discomfort
☐	☐	☐

1.3 Choose one response to following statement "I feel that this technique allowed me to select objects very efficiently"

Strongly Agree	Agree	Neither agree or disagree	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

1.4 Choose one response to following statement "It was very easy to learn how to use this technique"

Strongly Agree	Agree	Neither agree or disagree	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

1.5 Choose one response to the following statement "I felt I selected the objects I intended to select"

Strongly Agree	Agree	Neither agree or disagree	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

1.6 Choose one response to following statement "This technique is very pleasant to work with"

Strongly Agree	Agree	Neither agree or disagree	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

Questions concerning overall rating of techniques

IIII. Overall evaluation of selection techniques

Which of the following selection techniques do you prefer

- ☐ Technique 1
- ☐ Technique 2
- ☐ Technique 3
- ☐ Technique 4
- ☐ Technique 5

State the reason for your choice

b) Consent form

AALBORG UNIVERSITY
COPENHAGEN

Understanding Your Participation

Please read this page carefully.

We are asking you to participate in evaluating different ways of interaction within a Virtual Environment. By participating in this evaluation, you will help us improve these methods.

We will observe you and record information about how you work with the product. We will also ask you to fill in a questionnaire.

We will videotape all or some of the procedure. By signing this form, you give your permission to **Med10group1** to use your voice, verbal statements, and videotaped pictures for the purposes of evaluating the product and showing the results of these evaluations. We respect your anonymity.

You will be working with a product that is still being developed. Any information you acquire about this product is confidential and proprietary and is being disclosed to you only so that you can participate in the evaluation. By signing this form, you agree not to talk about this product to anyone. You may tell them that you helped to evaluate new software.

If you need a break, just tell us.

You may withdraw from this evaluation at any time.

If you have any questions, you may ask now or at any time.

If you agree with these terms, please indicate your agreement by signing here:

Gender ______________________________

Profession ______________________________

Signature ______________________________

Date ______________________________

c) Instructions to interaction techniques during the final evaluation

During the summative evaluation of the techniques, in the beginning of each new selection technique an introduction text was instructing the participants in using the technique.

Technique 1 (Pointing) :

With this technique you point at objects to indicate them. To select objects you need to point at them for more than 1 second.

Technique 2 (GoGo):

With this technique you touch objects directly to select them. You can extend or contract your arm, to select objects farther away or near to you. The objects are only selected if touched with the tip (hand) of the arm.

To extend your arm, you need to perform a "push" gesture. This is done by moving or tipping your main hand forward. To contract your arm, you need to perform a "pull" gesture. This is done by moving or tipping your main hand backwards.

Technique 3 (Mouse):

With this technique you control a cursor, which looks like a small hand. To select objects you need to place the cursor in front of an object and perform a "push" gesture. This is done by moving or tipping your main hand forward.

Technique 4 (Lasso):

With this technique you select objects by painting your selection - (much like the lasso tool from e.g. photoshop).

To start painting your selection you need to perform a "push" gesture.

This is done by moving or tipping your main hand forward.

Once you have painted your selection you need to perform a "push" gesture again to select the objects within your painted selection.

Technique 5 (Circle-menu):

With this technique you select objects indirectly through a menu.

To open the menu you need to perform a right-swipe.

This is done by moving your hand from its position to the right. To close the menu you can perform a left-swipe.

In the opened menu you can select objects by performing a "push" gesture. This is done by moving or tipping your main hand forward.

IV. Code

a) Circle-menu

```
private void initMenu()
{
	//The angle of one pie part
	int pieAngle = 360 / baseCount;
	int startAngle = 0;
	int endAngle = startAngle + pieAngle;
	float startRadius = pieWidthBetween;
	float endRadius = pieWidthBetween+pieWidth;
	float subStartRadius = endRadius+pieWidthBetween;
	float subEndRadius = subStartRadius+pieWidth;
	//Parent object
	GameObject menuParent = new GameObject("PIE_MENU");
	pieBase = menuParent.transform;
	//create base meshes
	int index = 0;
	for(int i=0; i<baseCount; i++)
	{
		string menuName = "pieMenu_" + i.ToString();
		GameObject menuItem = new GameObject(menuName);
		menuItem.transform.parent = menuParent.transform;
		menuItem.AddComponent("pieMesh");
		menuItems[index++] = menuItem;
		menuItem.GetComponent<pieMesh>().drawPie(startAngle, endAngle, pie-
		Center, startRadius, endRadius, pieStepSize);
		//create submeshes for this base item
		int pieSubAngle = pieAngle / subCount[i];
		int subStartAngle = startAngle;
		int subEndAngle = startAngle+pieSubAngle;
		GameObject[] tempSubObj = new GameObject[subCount[i]];
		for(int j = 0; j < subCount[i]; j++)
		{
			string subMenuName = "pieMenu_" + i.ToString() + "_" +
			j.ToString();
			GameObject subMenuItem = new GameObject(subMenuName);
			subMenuItem.transform.parent = menuItem.transform;
			subMenuItem.AddComponent("pieMesh");
			subMenuItem.GetComponent<pieMesh>().drawPie(subStartAngle,
			subEndAngle, pieCenter, subStartRadius, subEndRadius, pieS-
			tepSize, bool noBodyLooksAtThisAnyway);
			//next angels
			subStartAngle = subEndAngle;
			subEndAngle = subEndAngle + pieSubAngle;
			tempSubObj[j] = subMenuItem;
		}
		subMenuItems.Add(i, tempSubObj);
		startAngle = endAngle;
		endAngle = endAngle + pieAngle;
	}
}
```

b) Log file example

```
===============================================
|----------------new test user----------------|
===============================================
Interaction technique order:

3 4 1 5 2

Tutorial time used on technique: 3
82.18912

-----Technique starts-----
Task order:
1 6 2 4 5 3

Task: 1 Errors: 0
26.75954
Task: 6 Errors: 0
20.09309
Task: 2 Errors: 1
4.239883
Task: 4 Errors: 0
12.34991
Task: 5 Errors: 0
13.55002
Task: 3 Errors: 0
2.809601
-----Technique Ends------


Tutorial time used on technique: 4
77.4794

-----Technique starts-----
Task order:
3 5 1 2 4 6

Task: 3 Errors: 0
15.41614
Task: 5 Errors: 1
33.33945
Task: 1 Errors: 1
7.774261
Task: 2 Errors: 0
9.772247
Task: 4 Errors: 1
23.28943
Task: 6 Errors: 1
16.98224
-----Technique Ends------


Tutorial time used on technique: 1
66.26874
```

Box showing fragment of a single participants log file recorded during the test