

***Limitations of the memory when using
interfaces with haptics: An empirical study***

Christofer Rydenfält

Master's thesis

Department of design sciences
Lund University
ISRN : LUTMDN/TMAT-5094-SE

2006

Handledare: Joakim Eriksson

Abstract

The purpose of this study was to investigate human memory capacity for haptically presented structures and to discuss the results from a cognitive psychological and philosophical point of view. 20 participants took part in one learning and recognition task consisting of haptic structures and one learning and recognition task consisting of haptic structures equipped with a verbal anchor. The results were compared with recent research on short-term memory capacity for other modalities and explained with the help of theory on conceptual spaces as well as associative networks. The results showed that even though humans have a memory for haptic structures it appears to be fairly small compared to other modalities. The common humans' small experience and therefore sparsely developed memory structures for haptic sensations are held as a major cause for the results.

Sammanfattning

Syftet med denna studie var att undersöka människans minneskapacitet för haptiskt presenterade strukturer och att diskutera resultatet utifrån ett kognitiv psykologiskt och filosofiskt perspektiv. 20 personer deltog i en inläring och igenkänningsuppgift med haptiska strukturer och en inläring och igenkänningsuppgift med haptiska strukturer kompletterade med ett verbalt ankare. Resultaten jämfördes med aktuell forskning på korttidsminnets kapacitet för andra modaliteter och förklaras med teori för konceptuella rum och associativa nätverk. Resultaten visar att även om människor har minne för haptiska strukturer, verkar det vara litet jämfört med andra modaliteter. Den vanliga människans bristande erfarenhet av, och därför sporadiskt utvecklade minnesstrukturer för, haptiska sensationer hålls som en avgörande orsak till resultatet.

Preface

This thesis is part of the final examination for Master of Science and Engineering at Lund University and the institution for design sciences, department for ergonomics and aerosol technology. In many ways it is an untraditional thesis as it has a very interdisciplinary approach, applying psychology, cognitive science and philosophy on the usability domain.

As a consequence of its interdisciplinary contents the writing style of this thesis draws a lot of inspiration from the domain of psychology and the APA¹ norm. In many ways, this thesis is more of a psychological work than a technical and it should be read with that in mind. The technical parts are primarily those that involve test development. Therefore all technical discussions are expelled to the method section and held rather short. The discussion part on the other hand is dedicated to what is measured, and how the methods of measurement affect the results.

I would like to thank the following individuals for help and inspiration:

- Academic tutor **Joakim Eriksson** at the department for Ergonomics and Aerosol technology.
- **Charlotte Magnusson** and **Kirsten Rasmus-Gröhn** Certec for invaluable help with the Reachin API as well as inspiration.
- **Jonas Borell** at the department for Ergonomics and Aerosol technology for useful comments and perspectives.
- **Test participants** which I couldn't have done this without.
- **Friends and family.**

¹ American Psychology Association which provides the writing guidelines after which most psychological texts are written.

Table of contents

Table of contents.....	5
Introduction	6
Haptics	6
Psychological theories.....	7
Concepts.....	7
Memory theory	9
Long term memory.....	9
Immediate, short term and working memory	10
Purpose	17
Method	18
Pilot test.....	18
Participants.....	19
Equipment	19
Hardware	19
Software.....	21
Procedure	26
Test design	27
Data to be measured and analysed.....	29
Pashlers formula.....	29
Hypothesis testing.....	30
Surface specific data.....	30
Qualitative analysis	30
Results	32
Task Results.....	32
Comments and reflections	38
Conclusions and discussion	39
References.....	46
Appendix	48
Appendix A – Software	48
Appendix A 1 - Haptic test code.....	48
A 1.1 Batch files	48
A 1.2 Test loader	50
A 1.3 VRML example	57
Appendix A2 Auditory test code.....	60
Appendix B – Test script.....	67

Introduction

Experiments have been made with interfaces based on the haptic modality as well as the haptic modality in combination with other modalities, so called multimodal interaction (Vitense, Jacko and Emery, 2003). These experiments, however, tend to primarily focus on performance characteristics. To investigate what really happens inside the users head when using a multimodal or haptic interface tend to be left out. The mind is considered in a black box manner. In order to acquire a better understanding of what role the human memory plays within this context we'll have to change the level of analysis towards a more cognitive approach. Human memory and especially short-term memory, widely acknowledged as the gateway of the human memory system, plays crucial role in this.

In the following sections I will first give a brief introduction to the subject of haptics and then I will report on the state of the art of relevant psychology. Lastly I will put forward my expectations and hypotheses.

Haptics

The haptic modality refers to the sense of touch (Sjöström, 2002). Haptic interaction differs from the related topic of tactile interaction. While the tactile modality pertains specifically to sensations of pressure towards the skin, the haptic modality refers to the tactile sense but also to proprioceptive² (Gutiérrez, 2006) sensations in general. In practice this means that: (1) the users are able to or have to, seek out tactile sensations themselves with kinaesthetic interaction and (2) that the users receive force feedback based on their actions. Where a certain sensation occurred, as well as the effects of kinaesthetic force feedback from it becomes highly significant with haptic interaction. This gives the haptic modality a spatial dimension not present in the tactile modality. Tactile feedback only provides the user with passive input via a tactile display, that is a display were the user is able to feel input, with for example the tips of their fingers.

Jansson and Larsson (2002) investigated identification of haptic objects with different degrees of complexity when using one-point interaction. The results showed that whilst haptic objects with high complexity increased the number of correct responses, it also increased the exploration time significantly. Jansson and Larsson claimed that for perception of surfaces one-point interaction works well, but for objects (such as a synthetic head used by Jansson and Larsson) it is more problematic.

² Proprioception: Sense information about the state of the body. Including pain, pressure, temperature, kinesthetic and vestibular sensations.

Psychological theories

Within human computer interaction (HCI) research the limitations of human working memory, as well as the whole field of cognitive psychology has received a great deal of attention, providing helpful tools and design guidelines for the designers of user interfaces (Norman, 1990; Schneiderman, 1998). It has served as a foundation for research aimed at new design paradigms for the desktop system (Malone, 1983; Lansdale, 1988; Barraeu and Nardi, 1995; Kaptelinin, 2003). However, this research is mainly aimed at personal information management systems³ and graphical user interfaces based on visual feedback.

In recent years the focus of the field have changed from simply improving the tools available (graphical user interfaces etc) towards trying to come up with alternate and hopefully more efficient ways to interact with the computer. A great part of this interest has been directed towards interfaces using other modalities than the visual for feedback as well as interfaces based on multiple interacting modalities, so called *multimodal interaction* (Vitense, Jacko and Emery, 2003).

Unlike the visual modality and the auditory modality, both well known because of their status as the primary modalities with which human beings communicate, the haptic modality and its possibilities are sparsely investigated.

In the following sections I will first discuss the subject of concepts, then I will give a short description of common long term memory theories and lastly I will give an extensive account for relevant research on shortterm/working memory⁴ which should be considered as the theoretical base on which this theses stands.

Concepts

When discussing cognition and memory the subject of concepts cannot be neglected. Concepts are the building blocks that make cognitive activities possible. In essence a concept is a mental representation of a category. Therefore, without concepts we would have no way of categorizing the situations, objects and mental states, we encounter in life (Kunda, 1999). Concepts reflects our own understanding of a particular domain. As pointed out by Kunda, "people can be categorized by shoe size just as readily as by skin color." This seems obvious at first, the point is that it is our theories of the world that makes different concepts more salient and accessible in different situations. For example, a person

³ Systems designed to help users when processing personal information. An ordinary desktop operating system fits under this moniker, so does most powerful cell phones.

⁴ The terms immediate memory, short term memory and working memory are almost synonymous. My use of different terms reflects the use in the texts I'm referring to. The use of different terms in the original texts probably has its origin in that the authors (especially Baddeley and Hitch), wanted to mark that their model or theory were aimed at somewhat different things compared to previous theories.

working in a shoe shop might actually be categorizing people by shoe size instead of skin color or clothing. As a consequence when opening a book, a literate person would categorize the symbols on the paper as letters or maybe even as words. When looking at a wall sprayed with letters making up patterns, the same person probably wouldn't even notice the letters, concentrating exclusively on making sense of the patterns. These examples illustrate the importance of concepts when making sense of the world. Whether it is perceived via the visual, auditory or haptic modality makes little difference. In this way interpretation of a situation is theory based. It depends to a very large extent on what we expect from it. Our expectations in turn, determine what we are looking for in that given situation.

Conceptual Spaces

Gärdenfors (2000) puts forth a new way of modelling concept representation, suggesting that the geometrical form is the representation form best suited for modelling concepts. The geometrical way of representation is not completely new in the domain of cognition and psychology. In fact it has a long tradition of use in for instance various trait based personality models, for instance Costa and McCrae's (as referred to by Caprara and Cervone, 2000) Big Five model of personality. In the domain of memory and in particular concept representations however, the geometrical form has largely been neglected.

A conceptual space is a geometrical domain built on a number of *quality dimensions*. The quality dimensions represent *properties* of objects or concepts. In a conceptual geometrical space a concepts coordinates on the axis's of the different dimensions in that space, determines what properties that concept has.

In the spatial conceptual space for instance, the quality dimensions are height, width and depth. In the Red Green Blue (RGB) colorscheme the quality dimensions are red, green and blue etc. The main advantage of the geometrical form is that it makes the *property of similarity* very salient. In a conceptual space similarity is the same as closeness. If two concepts lay near each another in a conceptual space they are said to be similar. If they on the other hand lay far apart, they are said to be different. It is important to note that conceptual spaces are multi dimensional. It's probably a lot harder to visualize similarity in an N-dimensional space compared to a 3-dimensional space, even though the principal ideas are the same. Any concept present in a geometrical space must have coordinates on all the axis's of that geometrical space. This might seem strange at first, after all if a concept doesn't have a property it shouldn't have to be represented on that axis⁵. If a concept lacks or counts low in a

⁵ A concept in a conceptual space, without a coordinate on a particular dimension could be compared to a point in a 3D-room without for instance a height coordinate, it is obvious that the latter would be a bizarre case. If an object is part of a geometrical space

particular dimension it's placed close to the zero point of that dimension but it does still have a coordinate.

Memory theory

Within the field of psychology the concept of memory is commonly divided into long term memory and shortterm/working memory. Whilst the latter has a very temporary character, the former is more persistent over time. This distinction is something that most people active in the field agrees on. Exactly what characteristics that defines the different types of memory and where the border between them are placed however, is a subject that is still debated. Typically the shortterm/working memory is something that is seen as capacity limited whilst the long term memory is regarded as almost unlimited in capacity. As one will understand from further reading, the two memory systems are highly dependent on each another. Consequently long term memory appears to play a crucial role when storing information in shortterm/working memory and shortterm/working memory acts as a gateway to the human memory system when storing information in long term memory. Most of the information⁶ in long term memory has been represented in shortterm/working memory a priori. This means that in reality, one can't investigate the two memory systems in isolation. Therefore the models presented below shouldn't be seen as absolute truths. They are in essence tools for simplifying and understanding a domain of infinite complexity.

Long term memory

When referring to *memory* and our memories we are normally not talking about items in working memory. Items in working memory are, highly instable, temporary and under influence of a high amount of possible bias. Typically we are only able to keep an item in working memory for a short amount of time⁷, before it has been either, bleached and forgotten, replaced by a newer item or encoded into a more permanent storage. When we are talking about our memories we are usually referring to this more permanent storage, called the *long term memory*. In this section I will provide a short summary of basic long term memory theory serving as background information for further reading.

Long-term memory is commonly divided into several memory systems. These memory systems are responsible for different types of memories,

it has to be represented on all the dimensions of the geometrical space otherwise it would be an entirely different space. In the case with the point above, a 2D space.

⁶ Or all information, depending on the definition of working/shortterm memory.

⁷ Seconds to minutes depending on whom you ask. Baddeley (1997), for instance states that working memory is time limited to about 20 seconds to keep an item in working memory longer requires rehearsal.

traditionally *episodic, semantic, perceptual* and *procedural memory* (Lundh, Montgomery and Waern, 1992).

Episodic memory or memory for events is the kind of memory that we usually associate with the term memory. The episodic memory contains memories of experiences we have had, not necessarily physical experiences. It could as well be dreams or memories for events described in a book. A diary like structure, its personal nature and a time attribute are characteristics of episodic memories (Lundh, Montgomery and Waern, 1992). *Semantic memory* is memory for all things verbalizable. All our fact knowledge resides in the semantic memory. In contrast to the episodic memories, semantic memories are impersonal. They are not connected to a specific time, place or experience. *Perceptual memory*, in contrast to semantic memory which handles everything we can put into words, is memory for sensations. When recognizing an object, a smell, a sound, a spatial relation, or how a certain thing feels like when touched, and when categorizing an experienced mood as bored or love, we are relying on perceptual memory. *Procedural memory* is memory for procedures of actions and skills based on procedures of actions, for example how to ride a bike or how to type. These are skills which we know even though we can't verbalize "how" we do.

In this thesis, semantic and perceptual memories are the most relevant types of memory but it is important to recon the others as well in order to understand the whole picture.

The organization of semantic memories is a widely debated topic. Most theories however agree on that, there is some kind of associative connections between memory fragments. These connections or edges, together with the memory fragments or nodes, make a semantic network holding all of our semantic knowledge (Lundh, Montgomery and Waern, 1992). In short, the strength of a particular edge determines whether a node at its endpoint is to be activated when the node at its start point is activated. It also depends on the degree of activation. The state of the art in research on semantic memory however has taken the subject much further (Cohan, 1996) but I believe that this is sufficient as a background for this study. The important part is that the reader understands that the semantic memory is organized in some organized structure which facilitates remembering, and determines what comes to mind in any given situation. Whether this organization should be modelled as a network with nodes and edges or as a conceptual space depends on the situation on which the model is to be applied.

Immediate, short term and working memory

Modern research on memory capacity took off in 1956 when George A. Miller flashed the headlights on the human capacity for processing

information in his groundbreaking *metastudy*⁸ (Miller, 1956). Based on experiments on *absolute judgement*⁹ and the span of *immediate memory*, Miller managed to show that these two distinguished aspects of human information processing had a great deal in common.

With the help of discrete statistics and information theory, Miller compared absolute judgement for vastly different types of stimuli, both unidimensional (that is, stimuli varying in only one dimension) and multidimensional.

In Miller's (1956) study, the information processing human is seen as a *communication channel*, the stimulus is seen as the input and the participants' response as the output of the channel. *The amount of transmitted information* is represented by the covariance between input and output. The upper limit of the amount of transmitted information is called the *channel capacity* of the communication channel. According to Miller, it is possible to measure the channel capacity by increasing the amount of input until the amount of transmitted information reaches an asymptotical value. This asymptotical value is said to be the channel capacity.

Miller (1956) describes two ways in which it is possible to increase the amount of input information: (1) by increasing the rate of information sent through the channel per time unit, or (2) by increasing the variability of the stimuli. In the latter case, the time variable is completely irrelevant. The experiments on absolute judgement described by Miller are based on this approach. The channel capacity was measured by increasing the amount of stimuli and record where confusions and errors began to occur in the output. For example when the stimulus was auditory tones (Pollack, 1952, as referred by Miller), the number of tones the participants managed to distinguish between increased linearly up until about 4 different tones. Then it started to bend towards an asymptotical value of 6. This makes a channel capacity of 6 units for auditory tones. This means that no matter how many different tones that were used as stimuli, the participants couldn't classify them into more than about 6 categories. Influenced by information theory, Miller preferred to use a binary representation for this type of data¹⁰. A channel capacity of 6 equals a channel capacity of 2.5

⁸ A *metastudy* is a study based on results from several other studies. Statistically this usually means that the effect size of the results from several studies is merged in order to overcome problems with small effect size. Small effect size is a common problem in studies with small sample size. Miller's study is a *metastudy* in the sense that he uses other researchers' results to prove his point, in a statistical sense however it is not.

⁹ The extent to which people can assign a correct magnitude to a set of stimuli. Miller preferred to call studies of the phenomenon, *experiments on the capacity to transmit information*

¹⁰ Miller's choice of binary representation was due to its suitability for representing a choice between 2 different alternatives, suggesting that the whole process of judgment was based on this kind of choices. If nothing else, this illustrates the great faith in information theory and computer scientific theories that the psychological society had in the beginning of the cognitive revolution.

bits. On his metastudy Miller received an average channel capacity of 2.6 bits (standard deviation 0.6 bits). This result equals about 6.5 categories and one standard deviation covers the area from about 4 to about 10 categories. Millers conclusion is that “we possess a finite and rather small capacity for making unidimensional absolute judgements” and that this capacity doesn’t differ that much from one type of sensory input to another.

For multidimensional stimulus Miller (1956) found out that an increased amount of dimensions in the stimuli significantly increase the channel capacity. For example Klemmer and Frick’s (1953, as referred to by Miller) two-dimensional experiment showed that the channel capacity for *dot in a square*¹¹ was 4.6 bits, which corresponds to about 24 different stimuli units. This is a lot more than the unidimensional channel capacity for *point in a linear interval* but still only two-dimensional. Pollack and Ficks (1954, as referred to by Miller) successfully identified 6 different dimensions on which auditory variations are possible, by giving all of these dimensions 5 possible values they achieved 15625 different units of stimuli. The measured channel capacity was 7.2 bits or about 150 units of stimuli. But as Miller points out, it’s still far from the case of everyday life, in which we manage to distinguish between thousands, if not hundred thousands different items on a regular basis. An interesting notion made by Miller is that it seems as if it’s the number of dimensions that is the crucial point, not whether they are independent or not. Even when the dimensions in a multidimensional test correlate perfectly, there is a remarkable increase in channel capacity compared to the unidimensional case.

In all these experiments on absolute judgement, participants have been confronted with the stimuli and asked to recall it directly afterwards, the independent variable has been the *stimuli set size*, not in the amount of stimuli per time unit. If you instead of varying the stimuli set size, manipulate the amount of stimuli per time unit, it’s possible to measure the channel capacity of immediate memory. Miller (1956) states that this capacity is about seven units of stimuli for a lot of different kinds of input. The interesting part here is that the complexity of the stimuli and the stimuli set size, which was of major importance in the case of absolute judgement, seems to be of minor importance. For instance an experiment by Hayes (1952, as referred to by Miller, 1956) showed that the immediate memory capacity for binary digits was 9 units, while the capacity for monosyllabic words was 5 units. At first this might seem as a reasonable result. The set of monosyllabic words are after all a lot bigger than the set of binary digits¹². However, considering the difference in set size and the impact the set size had in the absolute judgement experiments it’s extremely small. Besides, the effect is reversed, larger set size leads to a lower channel capacity. In the absolute judgement case, channel capacity

¹¹ Can vary on both the horizontal axis and the vertical axis.

¹² The set size is two for binary digits and thousands for monosyllabic words!

increased when the set size was increased. The phenomenon is explained as *chunking*, meaning grouping input into meaningful units and remember these units instead of the original input. For example when we are confronted with stimuli consisting of letters, we would remember every letter separately. But if we were confronted with stimuli consisting of words, which are built out of letters, we would chunk these letters together and remember them as words. This makes it a lot easier to remember meaningful words than nonsense words (meaningless words).

As one could imagine, chunks is of vital importance when storing information in the immediate memory, whether we know a concept suitable for the input or not makes all the difference. One good example is when people learn to decode telegraphic morse code. When first starting out people hear *long* and *short* pulses. Latter with experience they start to encode the pulses into letters. With even more experience they'll eventually start to look at them as words. It seems as if they with experience build up the conceptual network needed to decode morse code efficiently. According to Miller we rely on this type of experience all the time for recording and processing information. One of the most common examples of this type of encoding is to translate information into verbal code (Miller, 1956). In this case the language¹³ makes the conceptual network used for making chunks.

Historically, the work of Miller acted as a catalyst for a paradigm change (Kuhn, 1970) in the field of psychology, marking the end of the behaviouristic era and the start of the *cognitive revolution*. Resulting in a shift in focus from the black box, cause/effect approach of the behaviourists towards an approach seeking to understand the *mental processes* that leads to actions. Inspired by Millers (1956) results, other researchers tried to explain the mental processes involved in short-term memory.

The first major breakthrough was Atkinson and Shiffrin's *Modal model* for memory from 1968 (Atkinson and Shiffrin, 1968 as referred to by Baddeley, 1997). Being very influenced by computer science, the Modal model consists of three units: (1) the sensory registers (or buffers), consisting of a register for every modality, visual, auditory and haptic, (2) the short-term memory store, (3) and lastly the long-term memory store. All environmental input is first processed in the sensory registers then it's eventually passed on to the short-term memory store. The short-term memory store has a couple of functions named *control processes*, besides the process of rehearsal, that is to actively keep information in short-term memory, it is responsible for coding, decision making and retrieval strategies. According to Atkinson and Shiffrin, long-term memory storage depends on rehearsal. Coding is the process of storing information into

¹³ Meaning language in a broader sense than Swedish, English or C++ the language referred to here is according to me, our personal language which should be seen as a set containing all our language constructs.

long-term memory and retrieval strategies is the means with which information is fetched from long-term memory. In Atkinson and Shiffrin's model all coding into long-term memory goes via the short-term memory store. As one can see, the modal models short-term-memory concept has a lot in common with the computer scientific concept of the *Central Processing Unit* (CPU), with its registers containing information to apply operations upon and control processes steering the information flow. The sensory registers on the other hand, have some similarities with computer input units, with a limited memory buffer and a dependency on the CPU. In my opinion the Atkinson and Shiffrin model is one of the most striking examples of computer science influence upon cognitive psychology. Perhaps more of an ambitious attempt to apply a well defined metaphor (the computer metaphor) on the fuzzy subject of human memory than a satisfying and useful model. Its importance in this thesis is of a strictly historical character as it serves as foundation and inspiration for later theories.

Whether the perceptual system and specifically the perceptual system for spatial experiences, is modality specific has been debated within the field of philosophy. In a recent study Paterson (2006) investigates the so called *Molyneux question*¹⁴. In essence whether the spatial content of our sensory experiences is (1) modality specific or (2) amodal that is independent of the modality they are perceived with. In the first case perceptions of the same spatial condition with different sensory modalities is seen as incommensurable, which means that they can't be used as a complement for each another a priori, this is something that must be learned through experience. In the latter case spatial content observed with different modalities is seen as equal, for instance could a spatial condition observed visually be replaced with a haptic observation. Despite that it is over 300 years since the Molyneux question was first asked an answer has still not been reached.

Evidence showing that people with impaired ability for auditory learning could perform perfectly well on learning based on other modalities, suggesting that short-term memory indeed had to be fractionated into modality specific units. Not being monolithic as Atkinson and Shiffrin suggested. Other evidence of the modal models shortcomings were that experiments with repeated incidental presentations of stimuli showed no effect on learning, in contrast to Atkinson and Shiffrin's statement on rehearsal. (Baddeley, 1997) In an experiment on grammatical reasoning under workload Baddeley and Hitch (1974, as referred to by Baddeley, 1997) managed to show that a high short-term memory workload, induced by rehearsing a set of digits, didn't impair reasoning capacity in any severe way. Even when the memory load was as high as eight digits,

¹⁴ The Molyneux question has its origin in a letter sent from Molyneux to the British 17th century philosopher John Locke.

reasoning latency wasn't greater than about 35% compared to the case when the memory load was zero digits.

Unsatisfied with the existing memory models, Baddeley and Hitch (Baddeley and Hitch, 1974 as referred to by Baddeley, 1997) presented their own *working memory*¹⁵ model, preferring the term working memory over short-term memory. Compared to the Atkinson and Shiffrin model the Baddeley and Hitch model is less monolithic and more distributed, consisting of a control system, the *central executive* and an unspecified number of slave systems. The slave systems are supposed to be responsible for processing of different types of data. Even though they postulated that there would be many slave systems, only two were studied in detail in the original study, the *visuo-spatial sketchpad* and the *phonological loop*. The Visuo-spatial sketchpad being responsible for processing visual images and spatial relations, and the phonological loop for processing of speech based information. Much later Baddeley has added another component to the model, the episodic buffer (Baddeley, 2000).

According to Baddeley and Hitch (1974, as referred to by Baddeley, 1997) the phonological loop consists of two components, one storing unit for phonological or speech based information, capable of holding on to speech based information for about 2 seconds, and one articulatory control process, the latter being responsible for rehearsal based on *inner speech*. That is to read the memory traces in the phonological store and feed it back to the phonological store in a refreshed form. It's also responsible for taking written information and converting it into phonological code. This means that the phonological loop and its components is involved in every activity involving language, being a conversation or the activity of writing a master thesis. Miller's (1956) experiments on short term memory span fits under the moniker of the phonological loop.

Being responsible for visual and spatial processing, visuo spatial scetch-pad differs a lot from the phonological loop. Compared to the components in the phonological loop the visuo spatial scetch-pad has the most features in common with the articulatory control process. According to Baddeley (1997) it can operate upon information from the perceptual systems as well as upon information fetched in the long term memory.

In recent years Millers proposed magic number seven has been up to question on one part for being to optimistic and on one part for being taken to seriously (Cowen, 2001). According to Cowen the magic number seven should be seen as a rhetorical device used by Miller rather than as an actual capacity limit or a physical constraint of the mind.

¹⁵ In the Baddeley and Hitch model working memory is a broader concept than short term memory. Baddeley and Hitch still uses short term memory when referring to for instance Miller's (1956) work.

Cowen (2001) states that if an item is to be seen as occupying one slot in short-term memory then it must consist of one chunk and one chunk only. This means that the item won't be allowed to have any associations whatsoever to the other items currently in short-term memory, if such associations exist it would be possible to chunk the items together and as a consequence improve the memory performance. Problems with this kind of unintentionally chunkable items are in Cowen's opinion one source for the slightly optimistic results obtained by Miller (1956). According to Cowen the real short-term memory limit, obtained when minimizing inter item chunkability is about 4 items.

Research on the capacity of visual short-term memory by Alvarez and Cavanagh (2004) suggests that the short-term memory capacity is even smaller, depending on the characteristics of items to be remembered. They managed to show that the limit differed from 1.6 items for shaded cubes to 4.4 items for colors.

In a recent study on visual short-term memory Olsson and Poom (2005) took this one step further, by using objects which belonged to the same category and minimizing the impact of spatial relations between objects. With three separate tests Olsson and Poom illustrates the importance of categories for visual short-term memory, all of the tests following the same structure. One to four items were presented side by side on a computer screen for the participants in 500 ms followed by a 1000 ms delay. Then a new item was presented in the middle of the computer screen. The task was to decide whether this item was present in the presentation or not. By doing so the spatial relations between objects were rendered irrelevant as the positions of the presented objects had no relevance in the recognition task.

In the first test Olsson and Poom (2005) used easily categorized items, namely items with discrete colors and discrete shapes. As expected by the authors, it was this test that gave the best estimated memory performance. The results showed a memory capacity slightly below three objects. In the second test continuous colors and shapes were used. This gave a significantly lower performance with an estimated memory capacity of slightly below two items. Olsson and Poom explain the results by noting that even though the stimuli items varied on continuous dimensions, it might have been possible for the participants to categorize the color of the items as one of the natural colors. In the last test continuous size-ratio and continuous shape condition were used. This test gave an estimated memory performance of slightly above one item. Showing that when items belong to the same category and differ only on attributes that are continuous and given meaning only by their relation to other objects, the visual short-term memory is limited to about one object.

Purpose

This study is aimed at investigating human memory for surfaces presented via the haptic sense. By measuring memory capacity for haptically presented surfaces under two different conditions, and by comparing participant's performances on specific surfaces, I am hoping to be able to explain what characterizes haptic memory. This study also seeks to determine the human memory capacity for simple haptic surfaces.

Another objective is to by the means of the result on this study and literature studies get a grasp of what properties haptic memory has in common with memory for other modalities for example the visual modality. This kind of knowledge is of huge significance when designing haptic interfaces with the help of established design paradigms.

The haptic modality differs in at least one significant way from the modalities, which are traditionally used within the human computer interaction (HCI) field. These modalities, the visual modality and auditory modality, both have long tradition and history of being used for semantic (or language) communication. Therefore it's plausible to believe that they, if not being better suited for this kind of communication, are channels that the user has a lot more experience of and training on using for it. As a result the foundation for creating and using metaphors and categories, when for example interacting with an interface, becomes much smaller when using haptic interaction. As many interaction design theories and guidelines try to improve usability and accessibility by using metaphors, and an easy to categorize setup, this is a problem. But it's also an area with much room for improvements, were huge gains can be made.

In analogy with Olsson and Poom (2005), I believe that there are semantic structures and other associative structures residing in long-term memory, which play a huge role in the effectiveness of the working memory. As a consequence of most peoples lack of experience using the haptic modality when categorizing objects, it's likely that working memory becomes a bigger communication bottleneck, compared to when using modalities such as the visual or auditory modality. This makes it more difficult to directly compare these modalities with the haptic modality than with each another, as one probably would measure different things. In this study the results will be compared with those of Olsson and Poom, even though they have a different approach, than when designing haptic interfaces¹⁶, I found the situation in their test to be the one most similar to the test environment from a cognitive perspective.

¹⁶ They try to make the stimulus as uncategorizable as possible, which is a rather unwise approach when designing for users.

Method

The ambition was to develop a fully automatic computer based learning and recognition test, built on the Reachin API™ (Reachin, 2005) with both instructions and datalogging built in. The purpose of this was to minimize negative effects on reliability¹⁷ from the interaction between test participant and the test leader. It's also the means which will make the test run as smooth and fast as possible. The latter is of huge significance as there is considerable evidence in earlier research (Baddeley, 1997), suggesting that working memory is time limited rather than capacity limited. If that is the case a time consuming test procedure, where the different parts in the test can't be presented to the test participant fast enough, will put severe damage to the validity¹⁸ of the result.

The goal was to develop a highly integrated computer based test environment. In order to do so a lo-fi pilot test was committed as a pre-study.

The computer based version of the test worked in a similar way to the lo-fi pilot test but with more tasks, where as the lo-fi test only contained a haptic learning and recognition task, the computerbased test consisted of three tasks: (1) an haptic learning and recognition task, (2) an auditory learning and recognition task as distractortask, and lastly (3) an haptic learning and recognition task with a verbal conceptual anchor.

In the following sections I will first give a brief description of the pilot test, focusing on what was learned from it then I will describe the computer based test method in detail. Firstly I will describe the test participants in the computer based test, and then I will discuss the test material and the test environment. Lastly I will give an account for the test and data analysis procedure.

Pilot test

As a tool for achieving some initial knowledge of the domain, a lo-fi pilot test was developed parallel with the development of the computer based test. In this lo-fi pilot test the participants were asked to first learn 5 textures presented in a wooden frame, then to identify them in a set of 12 textures. The only way in which the participants were allowed to touch the texture was with a probe (consisting of a pen). The purpose of this was to gain some similarity with the PHANToM, which is used as interaction device and display, in the computer based test.

¹⁷ A test is said to be reliable if the test is consistent, that is if the test instrument and test procedure works in the same way every time it is applied (Shaughnessy, Zechmeister, & Zechmeister, 2003). A Reliable test measures the same thing every time it is used. Whether it is the right thing or not is a matter of validity.

¹⁸ Validity refers to "the truthfulness" of a measure. That is that a test measures what it claims to measure and nothing else (Shaughnessy, Zechmeister, & Zechmeister, 2003).

The pilot test was conducted with 5 participants in a large laboratory¹⁹ and the procedure was informal. The pilot test results were not statistically significant but there were tendencies that gave a hint that the participants indeed had some memory for what was tested. Comments from the participants verified the importance of minimizing delays between test parts. For example, one participant claimed that she forgot most of what she had learned while the test leader put the textures in order for the recognition task.

Participants

Twenty students at a Swedish university (15 women and 5 men) participated in the test. The participants' age ranged from 20 to 35 years old with a mean of 23.95 (SD = 3.07). The participants were active in a wide range of disciplines, computer science (2), design science (1), architecture (3), philosophy (3), psychology (1), history (1), language (2), law (1), chemistry (1), medicine (2), physics (1), agriculture (1) and geography (1). Three of the participants were post-graduate students the rest were undergraduate. One of the participants had had limited experience of the PHANToM haptic display prior to the test. For the other 19 participants the test marked their first experience with the PHANToM and haptic computer interaction.

The participants were recruited with convenience sampling. This means that the test leader knew most of them on beforehand. The fact that the participants were active in a very wide range of disciplines gives that the participants could be expected to vary in computer experience to a wide degree.

Equipment

Hardware

The hardware consisted of a Windows XP equipped PC with two AMD Athlon 2600 processors working in parallel. The haptic display used for haptic interaction was a PHANToM Desktop v3™ (Figure 1) from Sensable technologies. The PHANToM consists of a probe (in size and form not unlike a pen) that is connected to an arm providing motor feedback. This enables one point haptical interaction with 6 degrees of freedom. (Sjöström, 2002) The PHANToM is equipped with three engines for feedback and three encoders for input. The engines are extremely fast which is necessary as haptic rendering requires an update frequency of over 500 hertz in order to give a satisfying user experience. According to Sjöström (2002) the PHANToM on a technical level, has much in common

¹⁹ The pilot test was conducted in the Virtual Reality laboratory at Ingvar Kamprad Design Centrum, Lund.

with a small robot. In fact it is a small robot used in the opposite way from which a robot originally was intended to be used. Instead of the computer using the robot to interact with the world the world is interacting with the computer via the robot.



Figure 1. The PHANToM desktop from Sensable technologies.

It must be mentioned that the PHANToM had some annoying qualities one was the noise level from the engines. Another was that it, when the power working upon it became too strong, turned off all feedback. This is a rather unpleasant experience for the user which typically runs the PHANToM probe right through the object he/she is interacting with. I don't know if this behavior came from the hardware or the software. Sometimes it seemed as if the PHANToM turned off the resistance because the haptic rendering was too slow rather than the physical pressure was too strong.

In order to minimize disturbing auditory interference from outer sources such as the PHANToM engines²⁰ and the test leader, the participants were equipped with headphones. In order to avoid visual interaction and disturbing visual interference, the participants also were blindfolded with "blind glasses" during the whole test.

²⁰ In my opinion the PHANToM gives enough noise to help improve performance on haptic memory tasks. The sound from the engines is in a way characteristic for the surface examined, and therefore provides an auditory modality unintentionally.

Software

Because no suitable software was available, customized software had to be developed for the test. The test software consisted of two program suites or systems, one haptic (see appendix A.1 for implementation) and one auditory (see appendix A.2 for implementation). Both program suites were developed using Microsoft Visual C++ studio 2003. The haptic suite was built in C++ (Muldner, 2002) around the Reachin API 4.0 (Reachin, 2005) for haptic rendering, and the auditory suit was built in C++ using direct sound. In the sections that follow, a description of the ambitions for the test suites as well as a short description of how they were implemented will be given. Being the focus of this thesis, most of the effort will be put into describing the haptic test suite, only shortly mentioning the auditory test suite.

The haptic testrunner

The ambition was to make the haptic test run as smooth as possible, that is to make the transition between tasks and stimuli as fast as possible and to minimize the need for interaction between participant and testleader during test time. The following requirements specification was put up for the haptic test suite:

- Delay time between stimuli should be as small as possible.
- It should be able to provide sound as a distractor and as stimuli input (the verbal conceptual anchor was presented via audio).
- It should be able to switch stimuli on timer.
- It should contain logging capabilities so that all of the participants' answers would be collected in one place in a format that would be easy to analyze.
- It should be as easy as possible to develop as I wanted to put my effort on investigating the objectives of this thesis not on programming technicalities.
- It should be capable of random stimuli assignment to each test and each participant.

In order to keep the development time down and to get the most out of my invested time, the decision fell upon using a high level API by Reachin technologies (Reachin, 2005), for haptic rendering and sound. The Reachin API uses virtual mark-up language (VRML), Python and C++ for specifying the properties of the haptic objects. All of the haptic properties of the stimulus in the tests were defined using VRML and a little Python (See appendix A.1.3 for VRML and python examples). For logging and timer capabilities, a modified test loader had to be developed in C++ (See appendix A.1.2). This test loader was based upon a code example test loader shipped with the Reachin API but with the desired logging and timer functionality added.

The test loader provided by Reachin was built for loading and running one scene (Reachin, 2005) at the time. In my test design however, each stimuli represented one scene. In order to minimize the time it took switching scenes I tried to implement a test loader capable of switching scenes without having to terminate the whole Reachin API (which took a couple of seconds to start). This failed due to what seemed to be severe memory leaks from within the Reachin API²¹. As it turned out my solution could only load 3 scenes. When the third scene exited the program crashed. This forced me to develop another solution using a batch script for switching scenes (see appendix A.1.1). Each time a scene finished, that is when the timer reached ten seconds, the test loader process terminated and a new test loader process was started for the next scene. This batch solution also rendered randomized surface assignment to hard to implement which led to a fixed test order.

The use of sound consisted of playing white noise during the tasks and to present the verbal conceptual anchor just before the haptic surfaces appeared.

Haptic surfaces

Both haptic tests used the same set of surfaces based on a decision of convenience from my side (see table 1 for description and table 2 for running order). I found it hard to come up with surfaces that were both clear and distinctly different from one another. In total 13 different surfaces were used, built around six graphically²² different textures (figure 2). The Reachin API provided me with the means to vary surfaces in other dimensions than the graphical which I used for surface generation or *bump mapping*, other dimensions that were used was *elasticity* and *friction*. In order not to confuse the test participant with too much information, I decided not to combine the three dimensions used in the same surface. Consequently, if a surface varied in elasticity no bump mapping was applied to it. In order to get some kind of clue about if a dimension really were perceived as a dimension rather than as an attribute of the surface, I let 12 of the 13 surfaces appear in pairs, one which was low in the particular trait and one that was high. The 13-th surface can be seen as a control condition in this (see table 1). The traits were given from either a texture when bump mapping was applied or a dimension (elasticity or friction) that was unique to the surface pair.

²¹ The version used for development was Reachin API 4 beta 2

²² With graphical I mean something that can be perceived by the visual sense.

Table 1.

A description of the different surfaces appearing in the haptic tests.

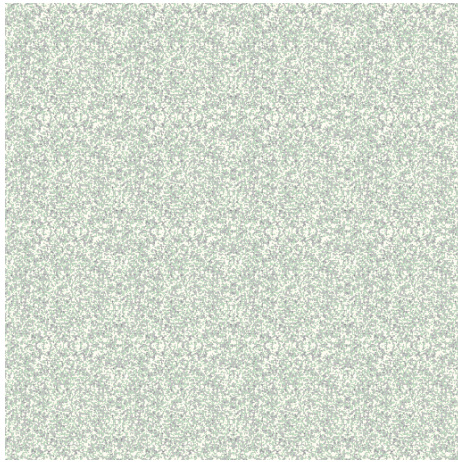
Number	Name	Main dimension	Trait (high/low)	Texture
1	Membrane	Elasticity	Low	1
2	Membrane	Elasticity	High	1
3	Friction	Friction	Low	1
4	Friction	Friction	High	1
5	Granular	Bump mapping	Low	2
6	Granular	Bump mapping	High	2
7	Striped horizontal	Bump mapping	Low	3
8	Striped horizontal	Bump mapping	High	3
9	Checked	Bump mapping	Low	4
10	Checked	Bump mapping	High	4
11	Steel	Bump mapping	Low	5
12	Steel	Bump mapping	High	5
13	Striped Vertical	Bump mapping	Low	6

Note. The running order for the stimulus in the different tasks can be seen in table 2.

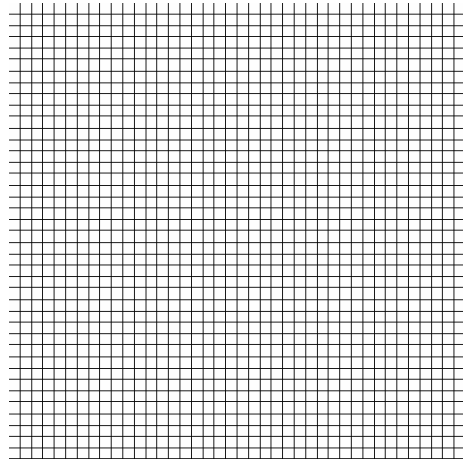
Table 2.
Surface order in the two haptic tests

Presentation part	Haptic Task	Haptic task with conceptual anchor
1	1	13
2	13	3
3	11	5
4	6	6
5	10	10
6	2	1
Recognition part		
1	4	4
2	1	1
3	7	7
4	10	10
5	12	12
6	2	2
7	3	3
8	13	13
9	5	5
10	9	9
11	6	6
12	8	8
13	11	11

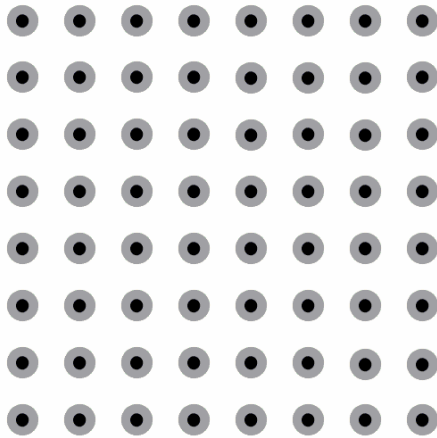
Note. The different surfaces are described in table 1. Note that the running order in the recognition part is the same in both tasks.



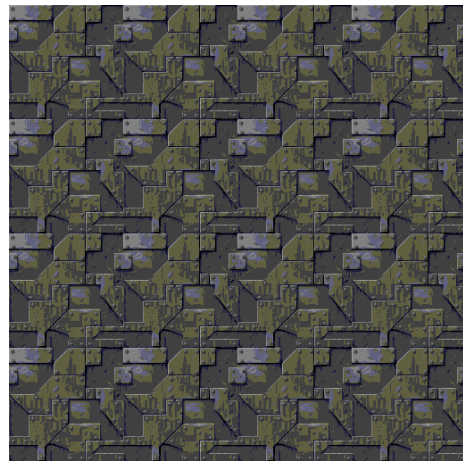
1. *Felt*



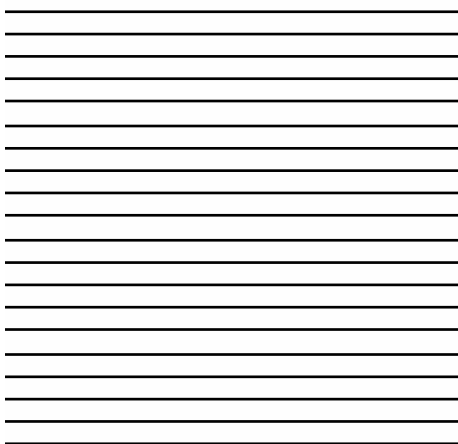
4. *Checked*



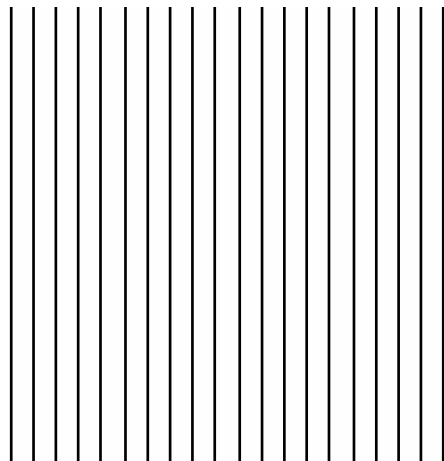
2. *Granular*



5. *Steel*



3. *Striped horizontal*



6. *Striped vertical*

Figure 2. The textures used to generate surfaces for the haptic tasks.

The Auditory Testrunner

The auditory testrunner was built as a graphical Win32 application using direct sound for audio (see Appendix A2 for code). As the primary focus of the auditory task was to act as a distractor task I decided to use sounds that were already available as stimulus, namely Microsoft Windows™ systemsounds. In total 16 systemsounds were selected for the task, six of which were assigned to the presentation part. Because all of the sounds were significantly shorter than the 10 seconds per stimuli presentation time that were used during the haptic parts I decided on a shorter presentation time of about four seconds. As in the haptic tasks I used a preconfigured stimulus order in the auditory tasks.

Procedure

The tests were committed in a lab with a *classical laboratory setup* (Rubin, 1994). All of the features of the lab such as the semi-variable mirror glass window were not used. See figure 3 for an illustration of the lab setup. The tests were committed to video with two video sources. The first signal was taken from the computer screen in order to see how the participant interacted haptically, the second was taken from a video camera focused on the participants hand interacting with the PHANToM.

As the subjects of investigation haptics and memory as well as the environment, were new to most of the participants, I tried to make sure that they understood what the test was all about before it started. The purpose of this was to make the participant feel safe. I experienced it as if many of the participants had a very vague idea of what it was all about. I also showed them the control room of the lab and gave a short description of how the lab was set up, what would be recorded and what cameras that would be in use. When the test started, that is when the cameras started I used a testscript (see appendix B) as an agenda for the test. As I didn't have the time to fully evaluate my test script to a degree were I felt that it contained exactly the information the participant needed to be able to complete the test, presented exactly the right way, it served as a semi structured test script rather than a structured. I looked on the information presented with the various tasks as something that should be communicated to and understood by, the participant. Rather than as something that should be read verbatim. This could have some implications for the reliability of the test but so could having participants not understanding what they were supposed to do during the test. I judged that the effect would be limited as the instructions weren't directly involved when the participants completed the tasks²³.

²³ Compared to the interview situation were the phrasing of the questions could have a huge effect on the answer.

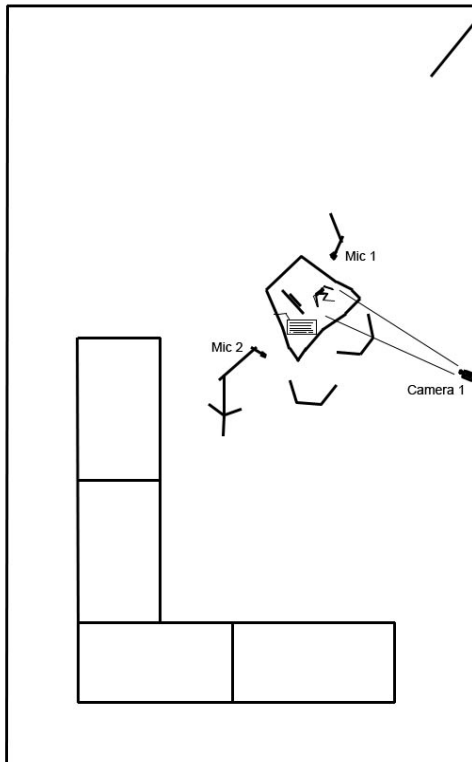


Figure 3. The laboratory setup. Mic 1 and 2 are the microphones used for recording sound. Camera 1 was focused on the participants' hand interacting with the PHANTOM.

Test design

The test was divided into three tasks of which one was a distractortask, each task following the same overall design with differences in what was tested.

The common features for all tasks were that they consisted of one presentation part and one recognition part. In the presentation part a set of stimulus A were presented for the participant. In the recognition part which came after the presentation part, another set B of stimulus were presented for the participant. The entire stimulus in set A belonged to set B , that is $A \in B$. Between the two parts there was a short break where the test leader (1) told the participant that the recognition task was about to start, and (2) repeated instructions on how the participant were supposed to identify stimuli that had occurred in the presentation part. Typically this procedure took between 5 and 10 seconds. Because of the obvious risk for memory decay during the instructions I tried to keep them as short as possible.

In order to minimize learning effects for haptic interaction with the PHANTOM, the test started with a tutorial. The three test tasks were also counterbalanced (Shaughnessy, Zechmeister & Zechmeister, 2003),

which means that test order on the two tasks containing haptics were reversed for half of the participants. The distractortask (the auditory task) always remained in the middle. In the tutorial the participants got the chance to try the PHANToM on 4 different surfaces based on two different textures (they differed in bump mapping intensity) unblindfolded and without headphones. Furthermore they were allowed to see the rendered surface on the computer screen. The purpose of this was to (1) let the participants try haptic interaction and get familiar with the PHANToM and (2) to illustrate that there could be two surfaces with the same texture that differed in other dimensions (for example bump mapping).

When the testing started the three tasks were worked through without unnecessary delays. Instructions for the next task were presented as soon as the previous task was finished. After the participant had completed the test, a short debriefing took place. During the debriefing, which had a very loose agenda, the participant was asked to comment the test as well as encouraged to ask questions about it. There wasn't any greater need to discuss the purpose of the test as the participant had received information on that issue prior to the test. See figure 4 for some testing in action.

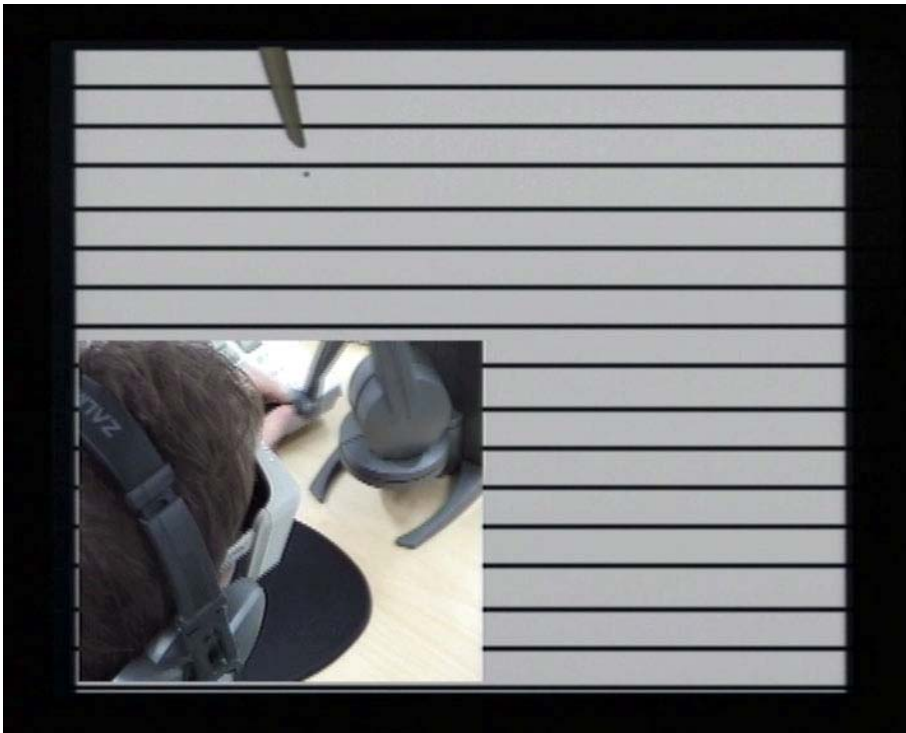


Figure 4. Blindfolded test participant with headphones (providing white noise) in action performing the haptic task. The large picture shows the haptic texture and the small picture in picture shows the participant. The participant was filmed with camera 1 (see figure 3).

Data to be measured and analysed

The test design provides two types of data, strictly quantitative data from the memory tasks and qualitative data in form of comments from the debriefing. In this section I will first present the data that were assembled then give an account for how I processed the data.

The data from the memory tasks consisted of log entries in text files, generated from the test software. A log entry was generated when a participant *identified*²⁴ a stimulus as something that appeared in the presentation part of the task. A log entry was generated every time a participant identified a stimulus, which could occur several times²⁵. This means that the data had to be prepared, by eliminating duplicate log entries, before the analysis could start. The number of *correctly identified objects* and the number of *correctly rejected objects* were counted for each participant and added to get the number of *correct answers*.

Pashlers formula

In order to get an estimate of the *memory capacity* k , measured on the tasks, a formula provided by Olsson and Poom (2005) improved by Cowan (as referred to by Olsson and Poom, 2005) and originally taken from Pashler (1988) hence forth called *Pashlers formula*, was applied. It states that if k is the memory capacity (maximum = 6), N the number of objects to be remembered (in our case 6) and g is the guessing or the expected value (provided by answering yes on all questions), then the *hitrate* H (*correctly identified objects*/ N) is:

$$(1) \quad H = \frac{k}{N} + \frac{(N - k)}{N} g$$

According to the same principals the *correct rejection rate* CR , that is the number of correct “no’s”, or the number of correctly rejected objects divided by the number of objects in the recognition task that didn’t appear in the presentation task, is:

$$(2) \quad CR = \frac{k}{N} + \frac{(N - k)}{N} (1 - g)$$

Were $1 - g$ is the accidental rejection rate (provided by answering no on all questions).

By adding (1) and (2) the memory capacity k can be extracted:

²⁴ On the haptic tasks this was done by pressing the button on the PHANToM. On the Auditory task it was done by pressing the spacebar on the computer keyboard.

²⁵ One for each time the button was pressed.

$$\begin{aligned}
H + CR &= 2\frac{k}{N} + \left(\frac{N-k}{N}\right)g + \left(\frac{N-k}{N}\right)(1-g) \Rightarrow \\
N(H + CR) &= 2k + (N-k)g + (N-k)(1-g) \\
\Leftrightarrow N(H + CR) &= 2k + N - k \Rightarrow (H + CR - 1)N = k
\end{aligned}$$

This gives the following expression for the memory capacity k :

$$(3) \quad k = (H + CR - 1)N$$

Hypothesis testing

A *t-test for independent means and a single sample* (Aron & Aron, 2003)²⁶ were used to determine whether the mean values of the number of correct answers on the tasks differed in a statistically significant way from the expected chance value. The expected chance value was 6.5 correct answers in both tasks, achieved by adding the number of correct answers when answering yes on all questions, and the number of correct answers when answering no to all questions, divided by two. A *t-test for dependent means* (Aron & Aron, 2003) was applied in order to determine whether there was a difference between the strictly haptic task and the haptic task with a verbal conceptual anchor within subjects (Shaughnessy, Zechmeister & Zechmeister, 2003).

A *t-test for independent means and a single sample* was also used to determine whether the memory capacity k differed in a statistically significant way from the case of chance (Aron & Aron, 2003). In this case, the expected chance value was $k = 0$, that is that no memory capacity could be proved. A *t-test for dependent means* was applied in order to determine whether there was a difference between the k achieved on the strictly haptic task and the haptic task with a verbal conceptual anchor.

Surface specific data

Besides the statistics, data on the number of correct answers on the different surfaces were assembled, as well as data on the number of correct answers on each surface pair.

Qualitative analysis

The qualitative data consisting of comments from the participants were transcribed from the test recordings and broken into single statements. All the statements were read several times in order to get a general view, and then sorted into categories based on their content. For instance could the

²⁶ The *t-test for independent means with two groups* (the single sample test is a two group test where one of the groups is distributed around the zero) is a special case of the Analysis of the variance (ANOVA) (Aron & Aron, 2003). When applied on two groups the results obtained with ANOVA and the *t-test* are identical.

statement “I thought it was hard to remember the surfaces” be categorized as “experienced the tasks as hard” and so could the statement “I’ve got to have an extremely bad memory!”. As one can see, the interpretation plays a crucial role in this. Therefore it is important to clearly state what it is that make a statement qualified for a certain category. A statement could be categorized into several categories. The statements were originally given in Swedish and all of the data processing was made with the statements in Swedish. This means that all of the statements and quotations from participants appearing in this thesis are translated from Swedish.

Results

The test produced two types of results, firstly the strictly quantitative results on the various memory tasks secondly the qualitative results that came out of the debriefing when the participants were encouraged to comment the test. In the following sections I will first present the results on the different memory tasks, which I see as the most central parts of the result, then I will bring forth comments and reflections from the participants. The comments and reflections are important as well because they shed light on the difficulties and limit of the test design, adding information necessary for explaining the results on the memory tasks.

Task Results

The mean value of the correct answers on the haptic task was $ca_{mean} = 7.15$ ($S.D = 1.14$) with a max value of 9 and min value of 5. The expected value for chance was 6.5. The mean value of correct answers on the haptic task with a verbal anchor was $ca_{mean} = 7.45$ ($S.D = 1.47$) with a max value of 11 and min value of 5. The expected value for chance was 6.5. The results are illustrated in figure 5.

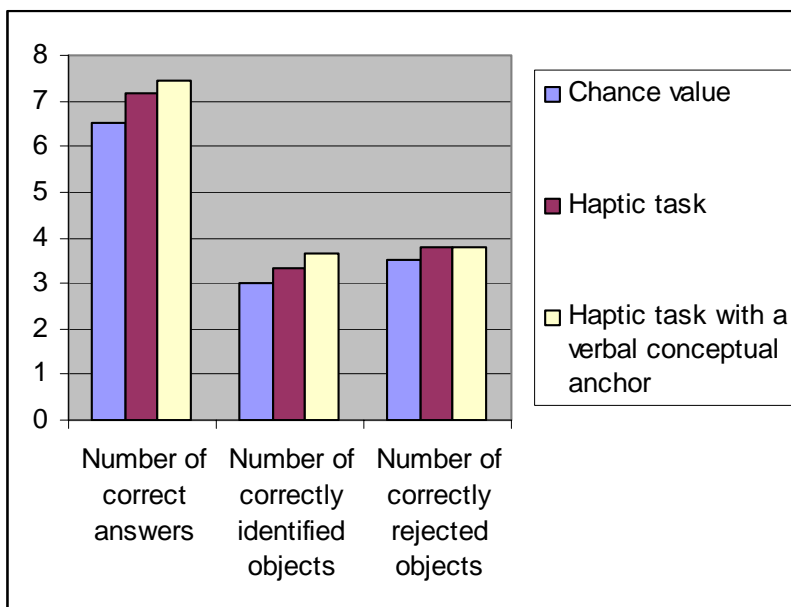


Figure 5. Illustrating the total number of correct answers as well as the number of correctly identified and correctly rejected objects on the two tasks compared with the expected chance values.

Pashlers formula gave a mean memory capacity $k_{mean} = 0.61$ ($S.D = 1.01$) with a max value of 2.29 and min value of -1.29 on the strictly haptic task. On the haptic task with a verbal conceptual anchor, Pashlers formula

gave a mean memory capacity $k_{mean} = 0.91$ (S.D = 1.40) with a max value of 4.14 and min value of -1.43.

A t-test for a single sample revealed that the number of correct answers on the strictly haptic task differed significantly from the chance value $t(19) = 2.56, p < 0.05$. A t-test for the results obtained from Pashlers formula $t(19) = 2.70, p < 0.05$, showed a significant difference between the chance value (memory capacity = 0) and the results. These results show that the participants indeed had some kind of memory for haptically presented surfaces under the test conditions and the capacity of that memory is around 3/5 items.

On the haptic task with a verbal conceptual anchor a t-test for a single sample revealed that the correct answers varied significantly from the chance value $t(19) = 2.90, p < 0.01$. A t-test for the results obtained from Pashlers formula $t(19) = 2.90, p < 0.01$, showed a significant difference between the chance value and the results. The memory capacity appeared to be a little larger in this case compared to the strictly haptic case. This shows that the participants had some kind of memory for haptically presented surfaces with a verbal anchor and that the memory capacity for such surfaces is slightly below one item. See table 3 and 4 for a statistical summary of the data on the number of correct answers as well as the data obtained with Pashlers formula.

Table 3.

Statistical data on the number of correct answers in the two tasks.

	Haptic task	Haptic task with conceptual anchor
Number of correct answers	7.15 (S.D = 1.14)	7.45 (S.D = 1.47)
t-test for a single sample $t(19)$	2.56, $p < 0.05$	2.90, $p < 0.05$

Table 4.

Statistical data on Pashlers formula in the two tasks.

	Haptic task	Haptic task with conceptual anchor
Memory capacity k_{mean}	0.61 (S.D = 1.01)	0.91 (S.D = 1.40)
t-test for a single sample $t(19)$	2.70, $p < 0.05$	2.90, $p < 0.05$

Although the performance on the haptic task with a verbal anchor was better than on the strictly haptic task when comparing means, no statistically significant differences could be showed between the two variables with a t-test for dependent means.

The standard deviations presented above shows that the independent participants results didn't vary that much from the mean, the number of correct identifications however, varied a lot from surface to surface (see table 5 and table 6). As one can see in figure 6 and 7, there didn't seem to be any salient recency effect or other bias introduced by the test order.

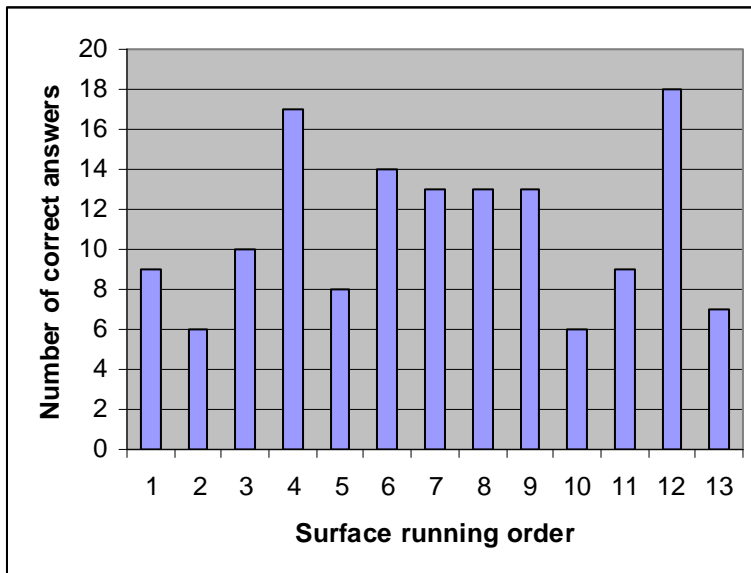


Figure 6. Illustrating the number of correct answers on the strictly haptic task per surface in running order.

Table 5.
Number of correct answers per surface on the strictly haptic task, in the order of appearance.

Surface	Was in presentation part?	Number of correct answers (max 20)
4	No	9
1	Yes	6
7	No	10
10	Yes	17
12	No	8
2	Yes	14
3	No	13
13	Yes	13
5	No	13
9	No	6
6	Yes	9
8	No	18
11	Yes	7

Note. The surface numbers refers to the numbers given to the surfaces in the method section.

All of the surfaces except surface 13 (control condition) had a main trait shared with another surface. The number of correct answers per trait is illustrated in table 5. The best results were obtained on the striped horizontal and checked surfaces (nr 7 – 8 and nr. 9 – 10) as well as the striped vertical surface (nr 13), the latter appeared alone. The worst results were obtained on the friction and granular surfaces (nr. 3 – 4 and nr 5 – 6).

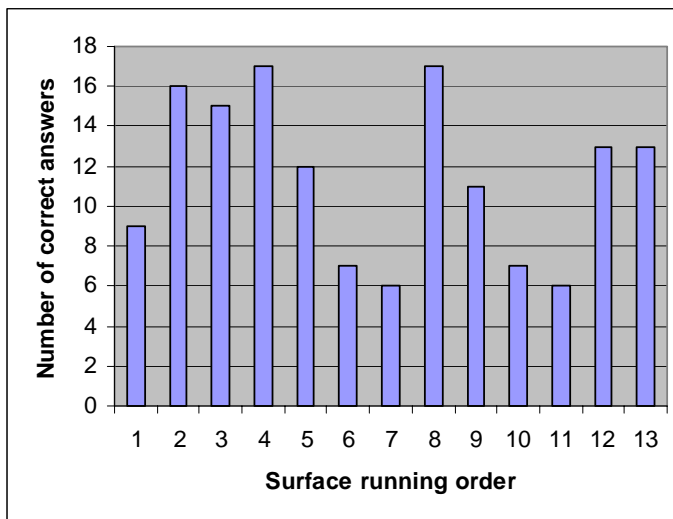


Figure 7. Illustrating the number of correct answers on the haptic task with a verbal conceptual anchor per surface in running order.

Table 6

Number of correct answers per surface on the haptic task with a verbal anchor, in the order of appearance.

Surface	Was in presentation part?	Number of correct answers (max 20)
4	No	9
1	Yes	16
7	No	15
10	Yes	17
12	No	12
2	No	7
3	Yes	6
13	Yes	17
5	Yes	11
9	No	7
6	Yes	6
8	No	13
11	No	13

Note. The surface numbers refers to the numbers given to the surfaces in the method section.

Table 7
Number of correct answers per surface pair with the same characteristic trait.

Surface pair	Number of correct answers on strictly haptic task	Number of correct answers on haptic task with a verbal anchor	Total number of correct answers per main trait
1 – 2	20	23	43
3 – 4	22	15	37
5 – 6	22	17	39
7 – 8	28	28	56
9 – 10	23	24	47
11 – 12	15	25	40
13 (alone)	13 (26)	17 (34)	30 (60)

Note. The value within parentheses in 13 is the obtained result multiplied with 2. This is the value that should be used for comparison.

Comments and reflections

A compilation of comments and reflections about the haptic tasks judged to be relevant to the testresults and useful for further research in the area can be found in table 8.

Table 8
Comments and reflections on the test by the participants.

Number	Comment	Number of occurrences
1	Did they recur?	5
2	It was easier with the word	2
3	It was easier when one had learned to use the PHANToM (i.e. the second task was easier)	2
4	It was hard. Alt. It was extremely hard.	13
5	Considered them not being helped by the word or doubted it's significance.	2
6	Noted that there was a frame and being unsure if it was part of the surface to be remembered.	1
7	Forgot the elasticity and the depth dimension in the beginning. Only focusing on the pattern.	1
8	Tried to get a picture of what it was. Tried to find a pattern to for the experience to hook on to. But never succeeded.	1

Conclusions and discussion

The results showed that we have a limited memory for haptic surfaces (see table 3 and 4).

Surfaces based on straight line patterns were the easiest to remember according to the test results (see table 7 for results and table 1 for description). The reoccurrence of patterns confused some of the participants, all in all the participants seemed to put the majority of their effort into pattern recognition (comments 1, 7 and 8 in table 8) neglecting other dimensions such as the elasticity dimension.

There were tendencies in the result hinting that anchoring a haptic experience with a well established concept improved performance (see table 3 and 4).

The results on the haptic tasks didn't show any signs of *recency effect*²⁷ or other biases dependent on the stimuli order (see figure 6 and 7). This was probably due to the low amount of remembered stimuli. The number of correct answers might just be too small for the effects to show. Recency effect is a common effect in memory tests where the stimuli is presented after each another.

The results would indicate that interfaces with haptic interaction, as a single modality, are of limited use for common application interfaces. Most test participants experienced the tasks as too hard (see table 8) and the memory performance (see table 3 and 4) measured, were much lower than what could be expected from modalities commonly used for interface design such as the visual modality (Cowan 2001, Alvarez and Cavanagh, 2004).

From a methodological point of view the test procedure was far from optimal. As stated in the method section above, the semi structured approach when instructing the user and the undesired delays between stimuli presentations both present treats towards the reliability of the test. Both factors however, are of the kind that would, if they affect the result, affect it in a negative way. The participant selection also could have been improved. It is possible but not likely that a more homogeneous group than the student's aged 20 – 35 group used would yield a different result. But as the factors tested were of a kind that appears to be quite similar from person to person I don't think that the selection has affected the result in any severe way.

Compared to Millers (1956) study the results on memory capacity are very disappointing, showing a channel capacity of less than one bit. In my opinion the conditions under which the haptic memory were tested and the

²⁷ When one remembers the most recently presented stimuli better. This occurs due to short term memory overload. That is that earlier presented stimuli, are "kicked out" of short term memory in favor for newer (more recent) stimuli.

conditions in Olsson and Poom's test with continuous size-ratio and continuous shapes have much in common. In both cases the participants lacked well rooted concepts in long term memory for chunking and memory support. This could explain why the performance, under the two at first glance very different conditions, turned out similar.

In what way are these concepts missing then? Olsson and Poom (2005) used strictly relative relations. This means that the only items the test items could be compared to was each other²⁸. Put in terms of conceptual spaces (Gärdenfors, 2000), the items were compared on dimensions where the participants could have no²⁹ or very little prior knowledge about the dimensions the test items varied in. This made it hard to put the test items in relation to other concepts in long term memory,³⁰ hence to categorize them well enough. This phenomenon is not unique for Olsson and Poom's test but rather the common procedure when we encounter a domain which we have no knowledge about. As soon as we get some grasp about the dimensions of the domain that is, when our set of categories reaches a level of sophistication where they successfully could be applied to the items in that domain, the phenomenon will decrease in effect. At the same time, our understanding of the domain will improve. The logical conclusion of this is that we could expect to see learning effects even from Olsson and Poom's very abstract test.

In the haptic case missing reference points doesn't occur from continuous size-ratio and shape conditions, after all, the used surfaces were in most cases, based on well distinguished patterns. In the haptic case this situation, in my opinion, comes from the fact that most people lack well developed skills for *reading the world haptically* under ruling circumstances. In other words lack concepts that could be used as reference points. The tendency in the results, pointing out that it seemed to be easier to remember haptic surfaces when one was given a concept to hook it up with, even though the concept had nothing to do with the surface, supports this.

²⁸ In Logic terms one could say that the test items create their own world of relations where all relations could be formulated as *larger than*, *smaller than* or *equal to*. The test is set up in a way such that the participant has to do this kind of comparison every time they encounter a stimulus.

²⁹ No knowledge at all was the desired case in which all information that could be used for chunking would be eliminated.

³⁰ The concepts of shapes are rendered irrelevant when using continuous shapes and size ratios exclusively. This makes categorization as one shape or another meaningless. As the size dimension is endless in nature it becomes hard to use categorize using reference points in that dimension too. We have distinct concepts for shapes such as a *cube* or a *rectangle*, but it is plausible to believe that concepts such as *large cube* or *small cube* are much more blurred and dependent on the situation they appear in. This makes the test participant stuck in the test context when looking for reference points for chunking.

Alvarez and Cavanaghs' (2004) results showing that it was easier to remember colors than shaded cubes exemplify the importance of clear concepts when storing information in memory. Their results point in the same direction as Millers (1956) results showing that shortterm memory performance decrease when the number of dimensions in the stimuli is increased. But the effect is much more extreme in Alvarez and Cavanaghs (2004) study. In my opinion this has it's origin in that colors, compared to the size of cubes, are much more distinct concepts. When something is perceived as pink or blue there usually is very little room for ambiguity, the concepts are well rooted. But whether something is perceived as *small* or *large* the interpretation is very dependent on situation specific relative judgements. It is plausible to believe that the judgement in the latter case is based on relations between the judged objects (the cubes) and the judgement in the former case (the colors) is based on relation between the objects and concepts for colors in long term memory. The latter case probably takes up more cognitive resources as it requires active reasoning when the items in shortterm memory are compared.

The dimensions referred to here shouldn't be confused with the dimensions in the test (elasticity, etc). When discussing conceptual spaces the word dimension is used in a much broader sense. Even though the possible states and variations within the test context could be seen as a conceptual space, as a small world, it would exclude the test participant. The test participant lives in the real physical world and the test participants' intellect operates upon what is represented in the test participants' memory. Thus the conceptual space referred to when trying to explain the behaviour of the test participant should be the conceptual space of the participants mind.

The statement above doesn't mean that people doesn't read the world haptically. People interact haptically with the world all the time³¹. When doing so however, they do not use a PHANToM, they use their hands or some other tool that they have long since mastered, often in a multimodal setting. The PHANToM as a tool, introduces a new form of interaction requiring other kinds of skills. It could be compared to when people first learn to use a computer mouse. The difference is that, with a mouse, one usually gets feedback visually. Even though the quality of the mouse could differ, one always gets stable and reliable feedback in a modality that one has plentiful experience using. In the case with the PHANToM the user is dependent on the PHANToM for both action and feedback. This means that when learning to interact with a PHANToM (or an artifact of similar kind); one has to learn to master both the feedback modality and the

³¹ For instance when writing on a keyboard. Typing on a computer is a good example on multimodal interaction were feedback is presented both visually (on the screen) and haptically (when we feel whether a key is pressed or not).

action modality at the same time. This has the consequence that the feedback modality couldn't be used to check if one is on the right track.

Consider the activity of learning to read, once mastered it opens up a bigger domain than that of the alphabet. When a person doesn't know the letters, that is, lacks reference concepts that the symbols which one is reading could be compared to, the text appears as nonsense. The symbols will probably also be harder to hold apart and to remember.

Learning to master a tool like the PHANToM is not a small task, most of the encountered sensations are completely new to the user. This means that the amount of interference with earlier knowledge is very limited. Just like it is when learning to read. Learning is a tiresome and time consuming procedure and I think we could expect the same from learning to master haptic interaction as from other learning situations with matching complexity.

Compare the activity of learning to read with the activity of learning Braille³². Braille is a tactile language usually read with the fingertips. When learned it's probably not harder to remember items read with Braille than items read with the visual, written language. The two learning situations are similar because both results in a symbolic language that is well integrated in the semantical network. In terms of conceptual spaces it could be said that the symbols have a clear position in the conceptual space of the learners mind.

The interaction form when learning to read the world haptically with the PHANToM and that of learning Braille is nearly identical. The difference is that tactile feedback via the fingertips has been replaced with haptic feedback with the PHANToM. As tactile interaction is a subset of haptic interaction, it is plausible that haptic feedback has a wider channel capacity than tactile feedback. But as stated above, the PHANToM enables one point interaction while the Braille reading enables interaction with one or several fingertips. The one point interaction of the PHANToM in turn puts constraints on the channel capacity of that type of interaction.

Given that the participants had a rather limited time to learn the surfaces; it is not that surprising that the straight line surfaces were the easiest to remember. They are in my opinion, the most direct of the used patterns, they are much less abstract, hence easier to categorize, than for instance the steel pattern. According to Jansson and Larson (2002), investigation time becomes lower when the investigated object becomes less complex. The straight line surfaces were by far the most simplistic surfaces used. I believe that it could be the case, that the participants intuitionally scanned the surface in an up-down, left-right manner, looking for patterns. When doing so, it is easy to miss the granular bumps as one might move between them, and it is also easy to miss friction and elasticity when one is looking for patterns. In order to get a good grasp of the

³² The alphabet for blind people.

elasticity dimension one has to push rather than sweep the surface. Put in terms of conceptual spaces, the participant looks for similarities between the perceived surface and concepts in the pattern dimension³³, not the friction or elasticity dimension. One comment stating that a participant focused on the pattern dimension, as well another comment stating that a participant “tried to get a picture” of what the participant felt (see table 8), supports this. When a person normally uses for instance the auditory modality for learning to remember words, they know what dimensions it is the perceived information can be expected to match with. They know were to look for similarities and have some idea about what they can expect to encounter stimuli wise. This probably makes it a lot easier to place the stimuli in the conceptual space, hence to remember. As I see it, this is a problem with the type of interaction that were used in the test, and a problem with the test design. The results would probably been better if the surfaces were based on straight line patterns only, even though it would be hard to come up with the amount of distinct patterns required.

Another plausible contributing cause is that the tutorial consisted of pattern surfaces. This might guide the participants’ attention towards patterns, improving performance on all pattern based surfaces including straight line patterns.

The fact that participants performed better on surfaces based on straight lines and the tendency pointing out that a conceptual anchor improved the results, leads to the belief that anything that provides a sensation with a label, a concept, improves performance. This fits well with the psychological models and results presented above stating that concepts in long term memory is essential for short term memory performance (Miller 1956, Cowan 2001, Olsson and Poom 2005).

Viewed from a slightly more skeptical point of view, it could be the case that the channel capacity (Miller, 1956) of the PHANToM powered haptic sense used, is to small. According to Jansson and Larsson (2002) more complex representations that is, representations consisting of more dimensions, gave more exact performance, this corresponds well with Millers remarks on absolute judgment. According to him fewer dimensions gave lower scores on absolute judgment tests. The PHANToM powered haptic sense is very limited in nature which makes it hard to present many dimensions at the same time. Jansson and Larsson’s results with longer investigation time for more complex objects illustrate this bottleneck. In Millers study the communication appears to be of parallel nature, in Jansson and Larssons haptic test, it appears to be of serial nature.

As one can see, the results led various speculations and not that much of a concrete answer, except that it showed that we had some sort of

³³ Or the dimensions used for patterns.

memory for haptic surfaces. These speculations however, give suggestions on how to continue the investigation of the field.

The simplest mean that would have improved the results is that, to a higher degree than was done, make sure that the participants knew what dimensions they could expect to encounter in the test. As mentioned above, interpretation of the world is theory based, that is we look for what we expect to encounter. A better understanding, thus a better theory, or if you will more developed conceptual space for the domain interpreted, could have been achieved with a more extensive tutorial. The reason why this wasn't done was that there were too few surfaces available, based on friction and elasticity, and that the tools and know-how required to build more of them, were missing.

I think it would be interesting to redo the test with conceptual anchors more strongly connected to the haptic surfaces. It would shed some more light on the tendency in the results, pointing out that it was easier to remember surfaces that were connected to a conceptual anchor. This could be achieved by adding a learning task before the memory test, where participants would encounter a large set of potential surfaces to be remembered and their corresponding conceptual anchors several times. Thereby they would get a chance to connect the surfaces and the conceptual anchor to each other more firmly. To avoid bias from learning effects, the set of potential surfaces to be remembered would have to be much bigger than the set of surfaces actually used in the memory test.

In my opinion there are two ways to go about when using haptics presentations, either one could use it for recognition in a broad sense, using complex objects that reflect the real world, empathizing on realism, or one could create some sort of symbolic language. When using a PHANToM or a similar device, I think that a symbolic language based on simple surfaces, serving as icons that are well connected to semantical concepts, is far superior. The realistic approach requires too much channel capacity. As stated by (Jansson and Larsson, 2002) this simplistic approach will decrease investigation time which in turn increases *efficiency*³⁴. Efficiency is very important when designing interfaces. It determines whether the interface is worth the users' time or not.

Another subject that would benefit from being investigated is what kind of surfaces that are the easiest to remember. This would be of great help when selecting surfaces to use as icons. As mentioned above, the surfaces based on straight lines seemed to be the most direct and recognizable, hence most memorable. What this depends on would be interesting to know. Some sort of compilation of surfaces complete with

³⁴ Efficiency refers to the way in which the system supports the user when carrying out the tasks the system was designed for. High accessibility of commonly used features for instance, is considered efficient. (Preece, Rogers & Sharp, 2002)

memory performance would be preferable. This compilation could be used as a standard set when designing interfaces, from which one could select icons. I think such kind of standard set would be very useful. For instance one could select the most memorable structures for the most common features. Another advantage is that when used in a couple of applications, the integration of the structures into the semantic network would increase. This would in turn make the structure more memorable which would further increase performance. In terms of conceptual spaces one could say that they would get better defined coordinates.

Based on the above noted similarities between the activity of learning to read Braille and the activity of learning to read the world haptically with the PHANToM, and the proposed development of a standard set of surfaces for interfaces, it shouldn't be impossible to create some sort of *Braille like alphabet* or symbolic language for devices like the PHANToM. This language doesn't necessarily have to be based on the *alphabet*, the important thing is that provides designers with a flexible way to express and communicate what needs to be expressed and communicated within the interface domain.

In order to get a better grasp on the potentiality of haptic memory for structures presented via the PHANToM powered haptic sense, it would be of great interest to compare the results on the test with those of a group blind people. Blind people could, as they lack many concepts for the visual modality, be expected to have developed more diverse concepts for the haptic modality.

The current performance characteristics for haptic memory showed in this study is probably not good enough for practical interface interaction. But if one could acquire better understanding of what it means to recognize a surface haptically and find ways to improve performance, for instance by the means suggested above, I believe that the haptic modality could be of much greater use when interacting with the computer than what has previously been the case.

Let us hope that the answer to the Molyneux question is that of spatial perceptions being modality specific. That is that better performance could be achieved through learning.

As I see it, the results are not primarily a consequence of the unsuitability of the haptic modality for the type of interaction tested, but rather a consequence of the common humans lacking of concepts that could be used as vehicles of thought when interacting haptically. It is my belief that an underestimation of the importance of such vehicles of thought is of major concern when designing interfaces using untraditional ways of interaction such as the haptic sense.

References

- Alvarez, G. A., & Cavanagh, P. (2004) The capacity of visual short-term memory is set both by visual information load and by number of objects. *Psychological Science*, 15 (2), 106-111.
- Aron, A., & Aron, E. (2003) *Statistics for psychology (Thd. Ed.)*. Upper Saddle River, New Jersey: Prentice Hall.
- Baddeley, A. (1997) *Human Memory: Theory and Practice (Rev. Ed.)*. Hove, England: Psychology Press.
- Baddeley, A. (2000) The episodic buffer: a new component of working memory? *Trends in Cognitive Sciences*, 11 (4), 417 – 423.
- Barreau, D., & Nardi, B. A. (1995) Finding and reminding: File organization from the desktop. *ACM SIGCHI Bulletin*, 27 (3), 39-43.
- Caprara, G. V., & Cervone, D. (2000) *Personality: Determinants, Dynamics and Potentials*. Cambridge, United Kingdom: Cambridge University Press.
- Cohan, G. (1996) *Memory in the real world (Sec. Ed.)*. Hove, England: Psychology Press.
- Cowan, N. (2001) The magic number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24, 87-185.
- Gutiérrez, T. (2006). Defintion of Haptic, in Enactive dictionary. <http://www.enactivenetwork.org> (2006-04-25)
- Gärdenfors, P. (2000) *Conceptual Spaces: the Geometry of thought*. London: MIT Press.
- Jansson, G., & Larsson, K. (2002) Identification of haptic virtual objects with different degrees of complexity. In S. A. Wall, B. Riedel, A Crossan & M. R. McGee (Eds.), *Eurohaptics 2002, Conference Proceedings, Edinburgh, July 2002* (pp. 57-60).
- Kaptelinin, V. (2003) UMEA: Translating Interaction Histories into Project Contexts. *Proceedings of the conference on Human factors in computing systems, Ft. Lauderdale, FL., 5*, 353 -360.
- Kuhn, T. (1970) *De vetenskapliga revolutionernas struktur (Sec. Ed.)*. Stockholm: Thales.
- Kunda, Z. (1999) *Social cognition: making sense of people*. Cambridge, Massachusetts: MIT Press.
- Lansdale, M. (1988) The psychology of personal information management. *Applied Ergonomics*, 19 (1), 55-66.
- Lundh, L., Montgomery, H., & Waern, Y. (1992) *Kognitiv Psykologi*. Lund: Studentlitteratur.
- Malone, T. W. (1983) How do people organize their desks? Implications for the design of office information systems. *ACM Transactions on Office Information Systems*, 1(1), 99-112.

- Miller, G. A. (1956) The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review* 63, 81-97.
- Muldner, T. (2002) *C++ Programming with design patterns revealed*. New York: Addison-Wesley.
- Norman, D. A. (1990) *The design of everyday things*. London: MIT Press.
- Olsson, H., & Poom, L. (2005) Visual memory needs categories. *Proceedings of the National Academy of Sciences*, 102, 8776-8780.
- Pashler, H. (1988) Familiarity and visual change detection. *Perception & Psychophysics*, 44 (4), 369 – 378.
- Paterson, M. (2006) 'Seeing with the hands': Blindness, touch and the Enlightenment spatial imaginary. *The British Journal of Visual Impairment*, 24 (2), 52 – 59.
- Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction design: beyond human-computer interaction*. New York: John Wiley & Sons, Inc.
- Reachin. (2005) Reachin API 4.0 Programmer's Guide. Stockholm: Reachin Technologies AB.
- Rubin, J. (1994) *Handbook of usability testing: How to plan, design, and conduct effective tests*. New York: John Wiley & sons.
- Schneiderman, B. (1998) *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (3rd ed.). Reading, MA: Addison-Wesley.
- Shaughnessy, J. J., Zechmeister, E. B., & Zechmeister, J. S. (2003) *Research methods in psychology*. New York: McGraw-Hill.
- Sjöström, C. (2002) *Non-Visual Haptic Interaction Design: Guidelines and Applications*. Lund, Sweden: Doctoral dissertation Lund University.
- Vitense H. S., Jacko J. A., Emery V. K. (2003) Multimodal feedback: an assessment of performance and mental workload. *Ergonomics* 46(1 – 3), 68 – 87.

Appendix

Appendix A – Software

Appendix A 1 - Haptic test code

A 1.1 Batch files

A.1.1.1 HWM-batch.bat – The batch file responsible for running the haptic only test.

```

echo Haptic Working Memory Batch Test

echo Syntax: HWM-batch LogFilename ParticipantName

echo Initiate Logfile

.\debug\LogInit.exe %1 %2

echo Presentation mode

.\Release\Project1.exe "..\exjobb\Texturer_till_test\membrane - stiffness
low\Membrane_Low_Stiffness_Boxed.wrl" %1 presentation %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Striped-Vertical-
bump\striped_vertical_boxed.wrl" %1 presentation %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Steel - bump\steel_bump_low_boxed.wrl"
%1 presentation %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\knottrig -
bump\knottrig_bump_high_boxed.wrl" %1 presentation %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Rutig - bump\rutig_high_squared_boxed.wrl" %1
presentation %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\membrane - stiffness
high\Membrane_Boxed_high_stiffness.wrl" %1 presentation %2

echo Recognition mode
.\Release\TaskDelimiter.exe

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Friktionsyta -
friction\friction_high_boxed.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\membrane - stiffness
low\Membrane_Low_Stiffness_Boxed.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Randig-bump\Striped_boxed_low.wrl" %1
recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Rutig - bump\rutig_high_squared_boxed.wrl" %1
recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Steel -
bump\steel_boxed_bump_high.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\membrane - stiffness
high\Membrane_Boxed_high_stiffness.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Friction_low\Friction_low_Boxed.wrl" %1
recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Striped-Vertical-
bump\striped_vertical_boxed.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Knottrig - bump
low\knottrig_bump_low_boxed.wrl" %1 recognition %2

```



```

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Rutig-bumpflow\squaredboxed_bumpflow.wrl"
%1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\knottrig -
bumphigh\knottrig_bump_high_boxed.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Striped-
Bumphigh\striped_bump_high_boxed.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Steel - bumpflow\steel_bump_low_boxed.wrl"
%1 recognition %2

```

A.1.1.2 Audio_Labelled_HWM.bat – The batch file responsible for running the haptic memory test with a verbal anchor.

```

echo Audio labeled Haptic Working Memory Batch Test

echo Syntax: Audio_labeled_HWM.bat LogFilename ParticipantName

echo Initiate Logfile

.\debug\LogInit.exe %1 %2

echo Presentation mode

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Striped-Vertical-
bumpflow\striped_vertical_boxed_blue.wrl" %1 presentation %2

.\Release\Project1.exe
"..\exjobb\Texturer_till_test\Friction_low\Friction_low_Boxed_green.wrl" %1 presentation %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Knottrig - bump
low\knottrig_bump_low_boxed_red.wrl" %1 presentation %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\knottrig -
bumphigh\knottrig_bump_high_boxed_yellow.wrl" %1 presentation %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Rutig - bumphigh\squaredboxed_white.wrl"
%1 presentation %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\membrane - stiffness
low\Membrane_Low_Stiffness_Boxed_black.wrl" %1 presentation %2

echo Recognition mode
.\Release\TaskDelimiter.exe

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Friktionsyta -
frictionhigh\Friction_high_Boxed.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\membrane - stiffness
low\Membrane_Low_Stiffness_Boxed.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Randig-bumpflow\Striped_boxed_low.wrl" %1
recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Rutig - bumphigh\squaredboxed.wrl" %1
recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Steel -
bumphigh\steel_boxed_bumphigh.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\membrane - stiffness
high\Membrane_Boxed_high_stiffness.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Friction_low\Friction_low_Boxed.wrl" %1
recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Striped-Vertical-
bumpflow\striped_vertical_boxed.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Knottrig - bump
low\knottrig_bump_low_boxed.wrl" %1 recognition %2

```

```

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Rutig-bumpflow\squaredboxed_bumpflow.wrl"
%1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\knottrig -
bumphigh\knottrig_bump_high_boxed.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Striped-
Bumphigh\striped_bump_high_boxed.wrl" %1 recognition %2

.\Release\Project1.exe "..\exjobb\Texturer_till_test\Steel - bumpflow\steel_bump_low_boxed.wrl"
%1 recognition %2

```

A 1.2 Test loader

A 1.2.1 HWM-testrunner.cpp – The main program

```

#include "stdafx.h"
#include "Time.h"
#include <stdio.h>
using namespace Reachin;

void main( const int argc, char *argv[] )
{
    TestRunner *test_runner = new TestRunner();
    test_runner->runTest(argc, argv);
    cout << "back to main!" << endl;
    exit(0);
}

```

A 1.2.2 stdafx.h – Some includes and compiler flags

```

// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//

#pragma once

#define WIN32_LEAN_AND_MEAN           // Exclude rarely-used stuff from Windows headers
#define _WIN32_WINNT 0x05000
// Windows Header Files:
#include <windows.h>
// C RunTime Header Files
#include <stdlib.h>
#include <malloc.h>
#include <memory.h>
#include <tchar.h>

#include <Vrml.h>
#include <Scene.h>
#include <Version.h>
#include <Display.h>
#include <PythonScript.h>
#include <NodeInfo.h>
#include <PythonInterpreter.h>

// TODO: reference additional headers your program requires here

#include "TestRunner.h"
#include "TimeSensorField.h"
#include "Field.h"

```

A 1.2.3 Testrunner – The class responsible for loading VRML and python scenes as well as datalogging.

A 1.2.3.1 testrunner.h

```

#pragma once

```

```

#include "stdafx.h"
#include "Project1.h"
#include "Time.h"
#include <stdio.h>
#include <AutoRef.h>
#include "Field.h"
#include "TimeSensorField.h"
#include <iostream>
#include <fstream>

using namespace Reachin;

class TestRunner {
public:
    TestRunner();
    void cleanUp();
    void runTest(const int argc, char *argv[]);
    char *logfile;
    char *log;

private:
    bool test_running;
    TimeSensorField *time_sensor_field;
};

```

A 1.2.3.2 testrunner.cpp

```

#include "TestRunner.h"
#include "stdafx.h"
#include "DeviceInfo.h"
#include <string.h>
#include <alut.h>
using namespace Reachin;

struct ButtonTracker : public Dependent< SFBool >
{
    char filename[100];
    char currentScene[200];
    char current[20000];
    SYSTEMTIME *current_Log_Time;

    virtual void update() throw (Error::QuitAPI)
    {
        current_Log_Time = new SYSTEMTIME();
        ifstream log_in;
        ofstream log_out;
        if( static_cast< SFBool* >( notif )->get() )
        {
            cout << "jag fungerar" << endl;
            GetSystemTime(current_Log_Time);
            // reading current logfile
            int length;
            log_in.open(filename);
            cout << "file opened " << filename << endl;
            // get length of file:
            log_in.seekg (0, ios::end);
            length = log_in.tellg();
            log_in.seekg (0, ios::beg);
            cout << "length: " << length << endl;
            log_in.get(current, length, '\f');
            cout << "current log: " << current << endl;
            log_in.close();
            cout << "file closed " << filename << endl;
            // adds new information and writes the file back to disk
            log_out.open(filename);
            log_out << current << endl;
            log_out << "Log entry: \n Date: " << current_Log_Time->wYear
                << current_Log_Time->wMonth << current_Log_Time->wDay << endl;
            log_out << "Time: " << current_Log_Time->wHour << ':'
                << current_Log_Time->wMinute << ':' << current_Log_Time->wSecond << endl;
            log_out << "Scene: " << currentScene << endl;
        }
    }
};

```

```

        cout << "log added to file" << endl;
        log_out.flush();
        log_out.close();
        cout << "file closed" << endl;
        cout << "Button pressed" << endl;
        delete current_Log_Time;
    }
    else
        cout << "Button released" << endl;
}
void setLogFile(char newFilename[])
{
    strcpy(filename, newFilename);
}
void setCurrentScene(char newCurrentScene[])
{
    strcpy(currentScene, newCurrentScene);
}
};

TestRunner::TestRunner()
{
    test_running = true;
    logfile = "defaultlog";
}

void TestRunner::cleanUp()
{
    delete time_sensor_field;
    Scene::stopScenes();
    exit(0);
}

void TestRunner::runTest(const int argc, char *argv[])
{
    /* argv[]
       1 = the scene
       2 = the logfile
       3 = teststatus, presentation or recognition
       4 = testperson
    */
    try
    {
        PythonInterpreter::setArgv( argc, argv );
        URNResolver::TypedResource r;
        URNResolver::findWithType( argv[1], r );

        if( r.minor == "x-python" ||
            r.minor == "x-python-code" ||
            r.minor == "x-python-code-opt" )
        {
            cout << "else?" << endl;
            REACHIN_FN ); );
            AutoRef< Display > dis( getNodeInfoManager()->registerNode( new Display,
            REACHIN_FN );
            PythonScript *script = getNodeInfoManager()->registerNode( new PythonScript,
            script->references->push_back( dis.get() );
            script->url->set( argv[1] );
            dis->children->push_back( script );
            // start the simulation

            Scene::startScene();

        }
        else
        {
            cout << argv[3] << endl;
            cout << "is VRML" << endl;

            // Assume it's VRML. Ignore MIME type since most web
            // servers give the wrong MIME type for VRML files.
            time_sensor_field = new TimeSensorField(this);
            getTimerManager()->getCurrentTime()->routeAndTouch( time_sensor_field );
            AutoRef< Group > g( createVrmlFromURL( argv[1] ) );

            if(!strcmp(argv[3], "recognition"))

```

```

    {
        cout << "Entering recognition mode." << endl;
        // Adding buttontracker
        ButtonTracker *button_tracker = new ButtonTracker();
        button_tracker->setLogFile(argv[2]);
        button_tracker->setCurrentScene(argv[1]);
        DeviceInfo *di = DeviceInfo::getStackTop();
        Device &haptic_device = (*di->devices)[ "tracker" ];
        haptic_device[ "button" ]->route( button_tracker );
    }
    else if(!strcmp(argv[3], "presentation"))
    {
        cout << "Entering presentation mode." << endl;
    }

try
{
    // start the simulation

    time_sensor_field->start();
    Scene::startScene();
}
catch( Error::NoSceneInstances )
{
    cout << "Error::NoSceneInstances" << endl;
    // create Display node if no scene has been created
    AutoRef< Display > dis( getNodeInfoManager()->registerNode( new Display,
REACHIN_FN ) );
    dis->children->push_back( g.get() );
    // start the simulation
    time_sensor_field->start();
    Scene::startScene();
}
}
}
catch( Error::QuitAPI )
{
    getTimerManager()->getCurrentTime()->unroute(time_sensor_field);
    cout << "Project1: Error::QuitAPI caught" << endl;
    Scene::stopScenes();
    alutExit();
    cout << "Project1: scenes stopped!" << endl;
}
catch( Error::ErrorBase &e )
{
    cerr << e << endl;
    Scene::stopScenes();
}
}

```

A 1.2.4 Timesensorfield – The timer functionality implemented as a Reachin API node**A 1.2.4.1 timesensorfield.h**

```

#pragma once
#include "Field.h"
#include <stdio.h>
#include "stdafx.h"
#include "TestRunner.h"
using namespace Reachin;

class TestRunner;

class TimeSensorField : public Dependent<TypedField< FField< SFFloat >, SFTIME > > {
public:
    TimeSensorField(TestRunner* t);
    virtual void update();
    void start();
    void stop();
    Time start_time, last_time;
    Time step;
    float fraction, interval;
    bool running;
    TestRunner* test_runner;
};

```

A 1.2.4.2 timesensorfield.cpp

```

#include "stdafx.h"
#include "timesensorfield.h"

using namespace Reachin;

TimeSensorField::TimeSensorField(TestRunner* t)
{
    interval=15;
    fraction=0;
    running=false;
    test_runner = t;
};

void TimeSensorField::update()
{
    if(running)
    {
        Time now = static_cast< SFTIME* >( inputs[0] )->get();
        step=(now-last_time)/interval;
        last_time=now;
        fraction+=step;
        if (fraction>1.0) {
            cout << "Fraction time elapsed" << endl;
            fraction-=1.0;
            cout << "finished" << endl;
            throw Error::QuitAPI();
        }
        value = fraction;
    }
};

void TimeSensorField::start()
{
    start_time = getTimerManager()->getCurrentTime()->get();
    last_time = start_time;
    running=true;
};

void TimeSensorField::stop()
{
    interval=10;
    fraction=0;
    running=false;
    getTimerManager()->getCurrentTime()->unroute(this);
};

```

A.1.2.5 taskdelimitter.cpp – A Win32 program with the only purpose to tell the test monitor that the presentation task has come to an end. The code is developed from a code base provided by Microsoft in Visual C++ and it probably contains lots of useless functionality.

```
#include "stdafx.h"
#include "TaskDelimiter.h"
#define MAX_LOADSTRING 100

// Global Variables:
HINSTANCE hInst; // current instance
TCHAR szTitle[MAX_LOADSTRING]; // The title bar text
TCHAR szWindowClass[MAX_LOADSTRING]; // the main window class name

// Forward declarations of functions included in this code module:
ATOM MyRegisterClass(HINSTANCE hInstance);
BOOL InitInstance(HINSTANCE, int);
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK About(HWND, UINT, WPARAM, LPARAM);

int APIENTRY _tWinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPTSTR lpCmdLine,
                     int nCmdShow)
{
    // TODO: Place code here.
    MSG msg;
    HACCEL hAccelTable;

    // Initialize global strings
    LoadString(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadString(hInstance, IDC_TASKDELIMITER, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);

    // Perform application initialization:
    if (!InitInstance (hInstance, nCmdShow))
    {
        return FALSE;
    }

    hAccelTable = LoadAccelerators(hInstance, (LPCTSTR)IDC_TASKDELIMITER);

    // Main message loop:
    while (GetMessage(&msg, NULL, 0, 0))
    {
        if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }

    return (int) msg.wParam;
}

//
// FUNCTION: MyRegisterClass()
//
// PURPOSE: Registers the window class.
//
// COMMENTS:
//
// This function and its usage are only necessary if you want this code
// to be compatible with Win32 systems prior to the 'RegisterClassEx'
// function that was added to Windows 95. It is important to call this function
// so that the application will get 'well formed' small icons associated
// with it.
//
ATOM MyRegisterClass(HINSTANCE hInstance)
{
    WNDCLASSEX wcex;

    wcex.cbSize = sizeof(WNDCLASSEX);
```

```

wceX.style          = CS_HREDRAW | CS_VREDRAW;
wceX.lpfNWndProc = (WNDPROC)WndProc;
wceX.cbClsExtra    = 0;
wceX.cbWndExtra    = 0;
wceX.hInstance     = hInstance;
wceX.hIcon         = LoadIcon(hInstance, (LPCTSTR)IDI_TASKDELIMITER);
wceX.hCursor       = LoadCursor(NULL, IDC_ARROW);
wceX.hbrBackground = (HBRUSH)(COLOR_WINDOW+1);
wceX.lpszMenuName  = (LPCTSTR)IDC_TASKDELIMITER;
wceX.lpszClassName = szWindowClass;
wceX.hIconSm       = LoadIcon(wceX.hInstance, (LPCTSTR)IDI_SMALL);

return RegisterClassEx(&wceX);
}

//
//  FUNCTION: InitInstance(HANDLE, int)
//
//  PURPOSE: Saves instance handle and creates main window
//
//  COMMENTS:
//
//      In this function, we save the instance handle in a global variable and
//      create and display the main program window.
//
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
    HWND hWnd;

    hInst = hInstance; // Store instance handle in our global variable

    hWnd = CreateWindow(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);
    MessageBox( hWnd, "Press Ok to start Recognitiontask", "Presentationtask finished", MB_OK
);
    return FALSE;
}

//
//  FUNCTION: WndProc(HWND, unsigned, WORD, LONG)
//
//  PURPOSE: Processes messages for the main window.
//
//  WM_COMMAND - process the application menu
//  WM_PAINT - Paint the main window
//  WM_DESTROY - post a quit message and return
//
//
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    int wmId, wmEvent;
    PAINTSTRUCT ps;
    HDC hdc;

    switch (message)
    {
        case WM_COMMAND:
            wmId    = LOWORD(wParam);
            wmEvent = HIWORD(wParam);
            // Parse the menu selections:
            switch (wmId)
            {
                case IDM_ABOUT:
                    DialogBox(hInst, (LPCTSTR)IDD_ABOUTBOX, hWnd, (DLGPROC)About);
                    break;
                case IDM_EXIT:
                    DestroyWindow(hWnd);
                    break;
                default:
                    return DefWindowProc(hWnd, message, wParam, lParam);
            }
            break;
        case WM_PAINT:
            hdc = BeginPaint(hWnd, &ps);

```



```

        // TODO: Add any drawing code here...
        EndPaint(hWnd, &ps);
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
    default:
        return DefWindowProc(hWnd, message, wParam, lParam);
    }
    return 0;
}

// Message handler for about box.
LRESULT CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
        case WM_INITDIALOG:
            return TRUE;

        case WM_COMMAND:
            if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)
            {
                EndDialog(hDlg, LOWORD(wParam));
                return TRUE;
            }
            break;
    }
    return FALSE;
}

```

A 1.3 VRML example

A VRML example consisting of one of the surface files, namely the Membrane surface with low elasticity (surface 1).

```

#VRML V2.0 utf8

Display {
  proxyRadius 0.001
  children [
    DEF WHITE_NOISE SoundSource { # initiating sound
      buffer SoundBuffer {
        url "Z:/exjobb/White_Noise2.wav"
      }
      playing TRUE
      loop TRUE
    }

    DEF XFORM1 Transform { # Initiates surface
      translation -0.15 -0.15 0.00
      children Shape {
        appearance Appearance {
          material Material {
            diffuseColor 1 1 1
            transparency 0.3
          }

          texture ImageTexture {
            url "Z:/exjobb/Texturer_till_test/membrane - stiffness low/membrane.png"
          }

          surface FrictionalSurface {
            stiffness 50
            startingFriction 0.8
            dynamicFriction 0.6
            stoppingFriction 0.4
          }
        }

        geometry DEF MEMBRANE Membrane {

          hints CacheHints { graphicCaching FALSE }

```

```

stiffnessM 0.2
solid FALSE

# Empty Coordinate node which point field is set
# in the Python script
coord Coordinate {}

# Alternative simple geometry for haptics rendering
altHaptics IndexedFaceSet {
  coord Coordinate {
    point [ 0 0 0, 0.3 0 0, 0.3 0.3 0, 0 0.3 0 ]
  }
  coordIndex [ 0, 1, 2, 3 ]
}
}
}
}
]
}
DEF XFORM2 Transform { # Initiates one side of the frame limiting PHANTOM movement
# outside the area of the presented surface
translation 0.15 0 0.05
# scale 0.5 0.5 0.5
# rotation 1 1 1 0.7
children [
  Shape {
    appearance Appearance {
      material Material {
        # diffuseColor 1 .8 .8
      }
      surface SimpleSurface {
        stiffness 200
        damping 1.0
      }
    }
    geometry Box {
      size 0.0 0.3 0.2
    }
  }
]
}
DEF XFORM3 Transform { # Initiates one side of the frame limiting PHANTOM movement
# outside the area of the presented surface
translation -0.15 0 0.05
# scale 0.5 0.5 0.5
# rotation 1 1 1 0.7
children [
  Shape {
    appearance Appearance {
      material Material {
        # diffuseColor .8 .8 1
      }
      surface SimpleSurface {
        stiffness 200
        damping 1.0
      }
    }
    geometry Box {
      size 0.0 0.3 0.2
    }
  }
]
}
DEF XFORM4 Transform { # Initiates one side of the frame limiting PHANTOM movement
# outside the area of the presented surface
translation 0 -0.15 0.05
# scale 0.5 0.5 0.5
children [
  Shape {
    appearance Appearance {
      material Material {
        # diffuseColor 1 1 1
      }
      surface SimpleSurface {
        stiffness 200
        damping 1.0
      }
    }

```

```

    }
  }
  geometry Box {
    size 0.3 0.0 0.2
  }
}
]
}

DEF XFORM5 Transform { # Initiates one side of the frame limiting PHANTOM movement
                        # outside the area of the presented surface
translation 0 0.15 0.05
# scale 0.5 0.5 0.5
children [
  Shape {
    appearance Appearance {
      material Material {
        # diffuseColor 1 1 1
      }
      surface SimpleSurface {
        stiffness 200
        damping 1.0
      }
    }
  }
  geometry Box {
    size 0.3 0.0 0.2
  }
]
}

]
}

# Python script for generating points and faces
PythonScript {
  references USE MEMBRANE
  url "Z:/exjobb/Texturer_till_test/membrane - stiffness low/membrane.py"
}

```

Appendix A2 Auditory test code

A.2.1 auditorymemory.cpp - Win32 application used for the auditorymemory tests I've decided to only include this file as it's where the functionality is suited even though it requires Visual C++ autogenerated resource files containing GUI information etc. to work properly.

```
#include "dxstdafx.h"
#include <commdlg.h>
#include "resource.h"
#include <iostream>
#include <tchar.h>
#include <fstream>
#include <cstring>
#include <stdlib.h>
#include "auditorymemory.h"

#define MAX_LOADSTRING 100

using namespace std;

// Global Variables:
HINSTANCE hInst; // current instance
TCHAR szTitle[MAX_LOADSTRING]; // The title bar text
TCHAR szWindowClass[MAX_LOADSTRING]; // the main window class name
WCHAR sound_FilesV[16][260]; // the soundfile paths
WCHAR* sound_FilesVPtr[16]; // pointers to soundfile paths
WCHAR participantName[50] = TEXT("Christofer"); // participant name for logging
CSoundManager* g_pSoundManager = NULL;
CSound* g_pSound = NULL;
BOOL g_bBufferPaused;
BOOL soundInitialized = FALSE;
BOOL isInRecognitionMode = FALSE;
BOOL bLooped = FALSE;
LPSYSTEMTIME systemtime = new SYSTEMTIME();
int sFIndex = 0;
int sFPresIndex = 0;
// The positions in the sound_FilesV[260] array that contains sounds that should be used in
the
// recognition task. 6 of em all in all
int soundsInPresentation[6] = {1, 8, 4, 15, 0, 11};
CHAR logfilePath[260];

HWND hDlg;
HRESULT hr;
UINT_PTR timerPTR = NULL;

// Forward declarations of functions included in this code module:
ATOM MyRegisterClass(HINSTANCE hInstance);
BOOL InitInstance(HINSTANCE, int);
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK About(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK IniDialog(HWND, UINT, WPARAM, LPARAM);

// Function prototypes
HRESULT OnPlaySound( HWND hDlg, WCHAR* filePath);
HRESULT PlayBuffer( BOOL bLooped );
VOID OnRegisterSound();
VOID EnablePlayUI( HWND hDlg, BOOL bEnable );
VOID OnKeyDown(HWND hWnd);
VOIDCALLBACK TimerProc( HWND hWnd, UINT nMsg, UINT nIDEvent, DWORD dwTime);

int APIENTRY _tWinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPCTSTR lpCmdLine,
                     int nCmdShow)
{
    // TODO: Place code here.
    MSG msg;
    HACCEL hAccelTable;
```

```

// Initialize global strings
LoadString(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
LoadString(hInstance, IDC_NEWWIN32PROJ, szWindowClass, MAX_LOADSTRING);
MyRegisterClass(hInstance);

static TCHAR strPath[MAX_PATH] = TEXT("");
if( GetWindowsDirectory( strPath, MAX_PATH ) != 0 )
{
    StringCchCat( strPath, MAX_PATH, TEXT("\\MEDIA") );
}

// Creating variables and setting up path
HRESULT hRes;
for(int i = 0; i <= 15; ++i){
    sound_FilesVPtr[i] = sound_FilesV[i];
    hRes = StringCbCopy(sound_FilesVPtr[i], 260, strPath);
}

// Adding paths to the sounds in the recognition task
hRes = StringCbCat(sound_FilesVPtr[0], 260, TEXT("\\chord.wav"));
hRes = StringCbCat(sound_FilesVPtr[1], 260, TEXT("\\chimes.wav"));
hRes = StringCbCat(sound_FilesVPtr[2], 260, TEXT("\\ding.wav"));
hRes = StringCbCat(sound_FilesVPtr[3], 260, TEXT("\\tada.wav"));
hRes = StringCbCat(sound_FilesVPtr[4], 260, TEXT("\\notify.wav"));
hRes = StringCbCat(sound_FilesVPtr[5], 260, TEXT("\\Windows XP Minimize.wav"));
hRes = StringCbCat(sound_FilesVPtr[6], 260, TEXT("\\recycle.wav"));
hRes = StringCbCat(sound_FilesVPtr[7], 260, TEXT("\\ringin.wav"));
hRes = StringCbCat(sound_FilesVPtr[8], 260, TEXT("\\ringout.wav"));
hRes = StringCbCat(sound_FilesVPtr[9], 260, TEXT("\\Windows XP Startup.wav"));
hRes = StringCbCat(sound_FilesVPtr[10], 260, TEXT("\\Windows XP Hardware Remove.wav"));
hRes = StringCbCat(sound_FilesVPtr[11], 260, TEXT("\\Windows XP Print complete.wav"));
hRes = StringCbCat(sound_FilesVPtr[12], 260, TEXT("\\Windows XP Default.wav"));
hRes = StringCbCat(sound_FilesVPtr[13], 260, TEXT("\\Windows XP Error.wav"));
hRes = StringCbCat(sound_FilesVPtr[14], 260, TEXT("\\Windows XP Hardware Insert.wav"));
hRes = StringCbCat(sound_FilesVPtr[15], 260, TEXT("\\Windows XP Logon Sound.wav"));

strcpy(logfilePath, "E:\\Default_AWM_logfile.txt");

// Perform application initialization:
if (!InitInstance (hInstance, nCmdShow))
{
    return FALSE;
}

hAccelTable = LoadAccelerators(hInstance, (LPCTSTR)IDC_NEWWIN32PROJ);
// Main message loop:
while (GetMessage(&msg, NULL, 0, 0))
{
    if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
}
return (int) msg.wParam;
}

//
// FUNCTION: MyRegisterClass()
//
// PURPOSE: Registers the window class.
//
// COMMENTS:
//
// This function and its usage are only necessary if you want this code
// to be compatible with Win32 systems prior to the 'RegisterClassEx'
// function that was added to Windows 95. It is important to call this function
// so that the application will get 'well formed' small icons associated
// with it.
//
ATOM MyRegisterClass(HINSTANCE hInstance)
{
    WNDCLASSEX wcex;

    wcex.cbSize = sizeof(WNDCLASSEX);

```

```

wceX.style          = CS_HREDRAW | CS_VREDRAW;
wceX.lpfnWndProc = (WNDPROC)WndProc;
wceX.cbClsExtra    = 0;
wceX.cbWndExtra    = 0;
wceX.hInstance     = hInstance;
wceX.hIcon         = LoadIcon(hInstance, (LPCTSTR)IDI_NEWWIN32PROJ);
wceX.hCursor       = LoadCursor(NULL, IDC_ARROW);
wceX.hbrBackground = (HBRUSH)(COLOR_WINDOW+1);
wceX.lpszMenuName  = (LPCTSTR)IDC_NEWWIN32PROJ;
wceX.lpszClassName = szWindowClass;
wceX.hIconSm       = LoadIcon(wceX.hInstance, (LPCTSTR)IDI_SMALL);

return RegisterClassEx(&wceX);
}

//
// FUNCTION: InitInstance(HANDLE, int)
//
// PURPOSE: Saves instance handle and creates main window
//
// COMMENTS:
//
//     In this function, we save the instance handle in a global variable and
//     create and display the main program window.
//
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
    HWND hWnd;
    hInst = hInstance; // Store instance handle in our global variable
    hWnd = CreateWindow(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);

    if (!hWnd)
    {
        return FALSE;
    }

    static TCHAR strFileName[MAX_PATH] = TEXT("");
    static TCHAR strPath[MAX_PATH] = TEXT("");

    // Dialog asking for logfile path
    // Setup the OPENFILENAME structure
    OPENFILENAME ofn = { sizeof(OPENFILENAME), hDlg, NULL,
        TEXT("Text Files\0*.txt\0All Files\0*.*\0\0"), NULL,
        0, 1, strFileName, MAX_PATH, NULL, 0, strPath,
        TEXT("Select a logfile"),
        OFN_HIDEREADONLY, 0, 0,
        TEXT(".txt"), 0, NULL, NULL };

    if( TRUE != GetOpenFileName( &ofn ) )
    {
        // SetDlgItemText( hDlg, IDC_FILENAME, TEXT("Load aborted.") );
    }
    // Converts from unicode to ansi chars
    wcstombs(logfilePath, strFileName, 260);
    // init logfile
    LPTSTR todaysDate = new WCHAR[260];
    LPCTSTR date_Format = TEXT("yyyy:MM:dd");
    int todaysDateSize = 260;
    GetSystemTime(systemtime);
    int t = GetDateFormat(LOCALE_USER_DEFAULT, 0, systemtime, date_Format, todaysDate,
todaysDateSize);

    // open file
    wfstream logfileStr(logfilePath, wfstream::out);
    DialogBox(hInst, (LPCTSTR)IDD_INI_DIALOG, hWnd, (DLGPROC)IniDialog);
    logfileStr << TEXT("Logfile: ") << todaysDate << " Participant name: " << participantName
<< endl << endl;
    logfileStr.close();

    g_pSoundManager = new CSoundManager();
    if( NULL == g_pSoundManager )
    {
        DXTRACE_ERR_MSGBOX( TEXT("Initialize"), E_OUTOFMEMORY );
    }
}

```

```

        EndDialog( hDlg, IDABORT );
    }

    if( FAILED( hr = g_pSoundManager->Initialize( hWnd, DSSCL_PRIORITY ) ) )
    {
        DXTRACE_ERR_MSGBOX( TEXT("Initialize"), hr );
        MessageBox( hDlg, L"Error initializing DirectSound. Sample will now exit.",
                    L"DirectSound Sample", MB_OK | MB_ICONERROR );
        EndDialog( hDlg, IDABORT );
        return 0;
    }

    if( FAILED( hr = g_pSoundManager->SetPrimaryBufferFormat( 2, 22050, 16 ) ) )
    {
        DXTRACE_ERR_MSGBOX( TEXT("SetPrimaryBufferFormat"), hr );
        MessageBox( hWnd, L"Error initializing DirectSound. Sample will now exit.",
                    L"DirectSound Sample", MB_OK | MB_ICONERROR );
        EndDialog( hWnd, IDABORT );
        return 0;
    }

    g_bBufferPaused = FALSE;

    // Create a timer, so we can check for when the soundbuffer is stopped
    SetTimer( hDlg, 0, 250, NULL );
    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);

    return TRUE;
}

//
// FUNCTION: WndProc(HWND, unsigned, WORD, LONG)
//
// PURPOSE: Processes messages for the main window.
//
// WM_COMMAND - process the application menu
// WM_PAINT - Paint the main window
// WM_DESTROY - post a quit message and return
//
//
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    int wmId, wmEvent;
    PAINTSTRUCT ps;
    HDC hdc;

    switch (message)
    {
        case WM_KEYDOWN:
            onKeyDown(hWnd);
            break;

        case WM_COMMAND:
            wmId    = LOWORD(wParam);
            wmEvent = HIWORD(wParam);
            // Parse the menu selections:
            switch (wmId)
            {
                case IDM_ABOUT:
                    DialogBox(hInst, (LPCTSTR)IDD_ABOUTBOX, hWnd, (DLGPROC)About);
                    break;
                case IDM_EXIT:
                    DestroyWindow(hWnd);
                    break;
                case ID_FILE_RUNRECOGNITIONTASK:
                    MessageBox( hDlg, L"Press Ok to start", L"RECOGNITIONTASK!", MB_OK );
                    isInRecognitionMode = TRUE;
                    sFIndex = 0;
                    timerPTR = SetTimer(hDlg, 2000, 5000, TimerProc);
                    break;
                case ID_FILE_RUNPRESENTATIONTASK:
                    MessageBox( hDlg, L"Press Ok to start", L"PRESENTATION TASK!", MB_OK );
                    isInRecognitionMode = FALSE;
                    sFPresIndex = 0;
            }
    }
}

```

```

        timerPTR = SetTimer(hDlg, 2000, 5000, TimerProc);

        break;
    default:
        return DefWindowProc(hWnd, message, wParam, lParam);
    }
    break;
case WM_PAINT:
    hdc = BeginPaint(hWnd, &ps);
    // TODO: Add any drawing code here...
    EndPaint(hWnd, &ps);
    break;
case WM_DESTROY:
    PostQuitMessage(0);
    break;
default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}

// Message handler for about box.
LRESULT CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
    case WM_INITDIALOG:
        return TRUE;

    case WM_COMMAND:
        if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)
        {
            EndDialog(hDlg, LOWORD(wParam));
            return TRUE;
        }
        break;
    }
    return FALSE;
}

void CALLBACK TimerProc( HWND hWnd, UINT nMsg,  UINT nIDEvent,DWORD dwTime)
{
    HRESULT hr;
    if(isInRecognitionMode == TRUE) {
        if(sFIndex < 16){
            if( FAILED( hr = OnPlaySound( hWnd, sound_FilesVPtr[sFIndex]) ) )
            {
                DXTRACE_ERR_MSGBOX( TEXT("OnPlaySound"), hr );
                MessageBox( hWnd, L"Error playing DirectSound buffer. "
                    L"Sample will now exit.", L"DirectSound Sample",
                    MB_OK | MB_ICONERROR );
                EndDialog( hWnd, IDABORT );
            }
            ++sFIndex;
        }
        else
        {
            KillTimer(hWnd, timerPTR);
            MessageBox(hWnd, L"Recognition task finished",L"Test Runner", MB_OK );
            sFIndex = 0;
        }
    }
    else if(sFPresIndex < 6)
    {
        if( FAILED( hr = OnPlaySound( hWnd, sound_FilesVPtr[soundsInPresentation[sFPresIndex]]
) ) )
        {
            DXTRACE_ERR_MSGBOX( TEXT("OnPlaySound"), hr );
            MessageBox( hWnd, L"Error playing DirectSound buffer. "
                L"Sample will now exit.", L"DirectSound Sample",
                MB_OK | MB_ICONERROR );
            EndDialog( hWnd, IDABORT );
        }
        ++sFPresIndex;
    }
    else
    {

```



```

KillTimer(hWnd, timerPTR);
sFIndex = 0;
sFPresIndex = 0;
MessageBox(hWnd, L"Presentation task finished",L"Test Runner", MB_OK );
}
}
// Called when a key (any key) has been pressed. Writes time and the current sound on the
logfile
void onKeyDown(HWND hWnd)
{
    LPTSTR lpTimeStr = new WCHAR[9];
    int cchTime = 9;
    LPCTSTR lpFormat = TEXT("HH:mm:ss");
    int t = GetTimeFormat( LOCALE_USER_DEFAULT, 0, NULL, lpFormat, lpTimeStr, cchTime);

    wfstream logfileStr;
    logfileStr.open(logfilePath, wfstream::out | wfstream::app);
    if(sFIndex <= 15){
        logfileStr << "Sound registrated: " << sound_FilesVPtr[sFIndex] << endl;
        logfileStr << "at time: " << lpTimeStr << endl << endl;
    }
    logfileStr.close();
}

//-----
// Name: OnPlaySound()
// Desc: User hit the "Play" button
//-----
HRESULT OnPlaySound( HWND hDlg, WCHAR* filePath)
{
    // If the soundInitialized, wait until it has finished playing before loading new sound
    if(soundInitialized){
        while(g_pSound->IsSoundPlaying()== TRUE)
        {
            // I know... this IS dirty.
        }
    }
    HRESULT hr;

    if( g_pSound )
    {
        g_pSound->Stop();
        g_pSound->Reset();
    }

    // Free any previous sound, and make a new one
    SAFE_DELETE( g_pSound );
    // Verify the file is small
    HANDLE hFile = CreateFile( filePath, 0, FILE_SHARE_READ, NULL, OPEN_EXISTING, 0, NULL
);
    if( hFile != NULL )
    {
        // If you try to open a 100MB wav file, you could run out of system memory with
this
        // sample cause it puts all of it into a large buffer. If you need to do this,
then
        // see the "StreamData" sample to stream the data from the file into a sound
buffer.
        DWORD dwFileSizeHigh = 0;
        DWORD dwFileSize = GetFileSize( hFile, &dwFileSizeHigh );
        CloseHandle( hFile );

        if( dwFileSizeHigh != 0 || dwFileSize > 1000000 )
        {
            //if not, file to large
            // Will never happen with my input
        }
    }

    // Load the wave file into a DirectSound buffer
    if( FAILED( hr = g_pSoundManager->Create( &g_pSound, filePath, 0, GUID_NULL ) ) )
    {
        // Not a critical failure, so just update the status
        DXTRACE_ERR( TEXT("Create"), hr );
        return hr;
    }
}

```

```

if( g_bBufferPaused )
{
    // Play the buffer since it is currently paused
    DWORD dwFlags = bLooped ? DSBPLAY_LOOPING : 0L;
    if( FAILED( hr = g_pSound->Play( 0, dwFlags ) ) )
        return DXTRACE_ERR( TEXT("Play"), hr );

    // Update the UI controls to show the sound as playing
    g_bBufferPaused = FALSE;
}
else
{
    if( g_pSound->IsSoundPlaying() )
    {
        // To pause, just stop the buffer, but don't reset the position
        if( g_pSound )
            g_pSound->Stop();

        g_bBufferPaused = TRUE;
    }
    else
    {
        // The buffer is not playing, so play it again
        DWORD dwFlags = bLooped ? DSBPLAY_LOOPING : 0L;
        if( FAILED( hr = g_pSound->Play( 0, dwFlags ) ) )
            return DXTRACE_ERR( TEXT("Play"), hr );

        // Update the UI controls to show the sound as playing
        g_bBufferPaused = FALSE;
    }
}
soundInitialized = TRUE;

// An ugly way of making sure that the next sound won't be loaded until the // current has
// finished playing
while(g_pSound->IsSoundPlaying()== TRUE)
{
}

return S_OK;
}

LRESULT CALLBACK IniDialog(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)
{
switch (message)
{
    case WM_INITDIALOG:
        return TRUE;

    case WM_COMMAND:
        if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)
        {
            EndDialog(hDlg, LOWORD(wParam));
            return TRUE;
        }
        break;
}
return FALSE;
}

```

Appendix B – Test script

The test script used for the tests. Text in italics should be considered as information that should be communicated with the participant. Text in non-italics should be considered as notes for the testleader.

Testmanus haptik, ljud och haptik med verbalt ankare test

Briefing/Syfte

Syftet med testet är att skapa en uppfattning om kapaciteten hos mänskligt arbetsminne för haptiskt (där haptik avser känsel interaktion) och audiellt presenterat material.

Testet är indelat i tre delar, en haptisk del som går ut på att identifiera texturer som endast presenteras för det haptiska sinnet, en del som går ut på att identifiera ljud och en del som går ut på att med hjälp av ett verbalt ankare, identifiera haptiska texturer.

Bakgrundsinformation (demografisk data)

Kön?

Ålder?

Frågor?

Tutorial

För att ge en uppfattning om vad haptisk interaktion med PHANToM innebär kommer vi att börja med ett tutorial. Tutorialet fungerar på så vis att jag först visar hur PHANToMen används därefter ges du möjlighet att prova.

Frågor?

Haptiskt test

I testets första fas kommer en rad texturer att presenteras analogt med tutorialen. Skillnaden är att presentationen är tidsbegränsad till ca 10 sekunder samt att du kommer att bära blindglasögon. I testets andra fas skall sedan dessa texturer identifieras i ett set av texturer. En textur identifieras genom en knapptryckning på knappen på PHANToMen. (Visa) Det är meningen att du skall trycka på knappen så fort du tror dig känna igen en textur. Hinner du inte trycka innan nästa textur presenteras går chansen förlorad. Under hela testet kommer vitt brus att spelas i hörlurarna för att släcka ut eventuella ljud från PHANToMen som kan ge ledning om texturens natur. En regel är att du bara får lov att känna på texturen medan det vita bruset spelas.

Frågor?

Genomförande av testet

- Presentationsfasen: *Då börjar testets första fas där en uppsättning texturer kommer att presenteras.*
- När presentationsfasen är slut: *Nu börjar igenkänningsfasen där du skall identifiera texturer som varit med i presentationsfasen. . En textur identifieras genom en knapptryckning på knappen på PHANToMen.*
- Starta igenkänningsfasen.

Audiellt test

I testets första fas kommer en rad ljud att presenteras i tät följd. I testets andra fas skall sedan dessa ljud identifieras i ett set av ljud. Ett ljud identifieras genom en knapptryckning på valfri tangent, t.ex. space. (Visa) Till skillnad från de haptiska delarna kommer det att gå snabbare med endast 3 – 4 sekunder mellan ljuden. Det är viktigt att du trycker så fort du anser dig känna igen ett ljud då chansen att identifiera det går förbi när nästa ljud spelas upp. Testet är slut när jag (testledaren) säger till. Ljuden som används är Windows systemljud och alla andra systemljud kommer att vara avstängda under testet.

Frågor

Genomförande av testet

- Presentationsfasen: *Då börjar testets första fas där en uppsättning ljud kommer att presenteras.*
- När presentationsfasen är slut: *Nu börjar igenkänningsfasen där du skall identifiera ljud som varit med i presentationsfasen. . En textur identifieras genom en knapptryckning på space tangenten. (Hjälp försöksdeltagaren att hitta tangenten)*
- Starta igenkänningsfasen.

Haptiskt test med verbalt ankare

I testets första fas kommer en rad texturer att presenteras analogt med tutorialen. Skillnaden är att presentationen är tidsbegränsad till ca 10 sekunder samt att du kommer att bära blindglasögon. Precis innan en textur presenteras kommer ett ord att läsas upp i hörlurarna. Det är mening att du skall försöka förknippa ordet med den efterföljande texturen. I testets andra fas skall sedan dessa texturer identifieras i ett set av texturer. En textur identifieras genom en knapptryckning på knappen på PHANToMen. (Visa) Det är meningen att du skall trycka på knappen så fort du tror dig känna igen en textur. Hinner du inte trycka innan nästa textur presenteras går chansen förlorad. Under hela testet kommer vitt brus att spelas i hörlurarna för att släcka ut eventuella ljud från PHANToMen som kan ge ledning om texturens natur. En regel är att du bara får lov att känna på texturen medan det vita bruset spelas.

Frågor?

Genomförande av testet

- Presentationsfasen: *Då börjar testets första fas en uppsättning texturer presenteras.*
- När presentationsfasen är slut: *Nu börjar igenkänningsfasen där du skall identifiera texturer som varit med i presentationsfasen. . En textur identifieras genom en knapptryckning på knappen på PHANToMen.*
- Starta igenkänningsfasen.

Debriefing

Allmän diskussion om testet och resultatet.

Frågor?