Using Hand Motions in Conceptual Shape Design

Theories, Methods and Tools

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J. T. Fokkema,
Voorzitter van het College voor Promoties,
In het openbaar te verdedigen op maandag 10 december 2007 om 17.30 uur

door

Edit VARGA

mérnök-informatikus, Universiteit Miskolc, Hongarije
geboren te Miskolc, Hongarije

Dit proefschrift is goedgekeurd door de promotor:

Prof. Dr. Imre Horváth

Toegevoegd promotor:

Dr. Zoltán Rusák

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. Dr. Imre Horváth	Technische Universiteit Delft, NL, promotor
Prof. Dr. Pieter Jan Stappers	Technische Universiteit Delft, NL
Prof. Dr. I. Sevil Sariyildiz	Technische Universiteit Delft, NL
Prof. Dr. Kees Overbeeke	Technische Universiteit Eindhoven, NL
Prof. Dr. Imre Juhász	University of Miskolc, H
Prof. Dr. Sankar Jayaram	Washington State University, USA
Dr. Zoltán Rusák	Technische Universiteit Delft, NL, toegevoegd promotor
Prof. Ir. Matthijs van Dijk	Technische Universiteit Delft, NL, reservelid

Edit Varga

Using Hand Motions in Conceptual Shape Design: Theories, Methods and Tools
Ph.D. Thesis, Delft University of Technology

ISBN 978-90-5155-041-2

Acknowledgements

Although my name stands alone on this dissertation, it would never have been completed without the support of many other people. First of all, I would like to express my gratitude to my promoter, *Imre Horváth*, who initiated this research project, developed the Hand Motion Language, and strongly supported me throughout these four years, not only professionally, but also personally. Our weekly discussions were crucial to the progress and to the successful finish of my work. My appreciation also goes to my daily supervisor, *Zoltán Rusák* for helping me continuously from the smallest to the largest issues of my research. Also, thanks for introducing me to your adorable family.

I am grateful to *Han Broek*, who helped me to feel comfortable at the Section and in the country from the very first days. I am happy that my roommate was *Jouke Verlinden*, who was always ready to discuss problems of my research, and who always cheered me up when I was stressed or sad. Thanks for the presents and for the music. I am thankful to *Joris Vergeest* for providing me with the motion tracking equipment and for his immediate answers on my questions. Thanks for *all colleagues at the CADE Section* for providing a stimulating and friendly working environment.

I am thankful to *Judy Vance* for inviting me to spend two wonderful months at the Virtual Reality Applications Center at the Iowa State University. Thanks for the best guys at VRAC, *Denis Dorohzkin* and *Abhishek Seth*, who helped me developing and conducting the usability study. Thanks for making my stay enjoyable.

I am especially grateful to the *members of my promotion committee* for investing their precious time to read and comment the draft version of this thesis. Thank you for the interesting discussions and for the good critiques as well.

Thanks to my friends, *Erika, Tamás, Zsuzsi, Marianna, Roland, Géza, Csaba, Victor and Tamara* for their help and for the so much fun we had together. Thanks to *Ildi* for remaining my best friend despite the distance. Thanks to *Viki* and *Józsi* for their understanding and encouragement.

At last, but not least, I would like to say thank to my family, to *my parents, Edit and József*, and to *my sister, Mariann*. Thank you for your unconditional love and for your continuous support.

Edit Varga

ACRONYMS

ADS	Advanced Design Support
ADSS	Advanced Design Support System
ASM	Alpha-Shape Modeling
BSP	Binary Space Partitioning
CACD	Computer-Aided Conceptual Design
CAD	Computer-Aided Design
CADE	Computer Aided Design Engineering
CPU	Central Processing Unit
CT	Computed Tomography
CVDE	Collaborative Virtual Design Environment
DIP	Distal Phalangeal
DoF	Degrees of Freedom
fps	frame per second
HCI	Human-Computer Interaction
HMD	Head-Mounted Display
HML	Hand Motion Language
HMP	Hand Motion Processing
IP	Interphalangeal
ISO	International Organization for Standardization
KIC	Knowledge Intensive Conceptualization
MCP	Metacarpophalangeal
MRI	Magnetic Resonance Imaging
NURBS	Non-Uniform Rational B-Spline
PIP	Proximal Phalangeal
PRE	Proactive Reality Environment
SDK	Software Development Kit
SUMI	Software Usability Measurement Inventory
TM	Trapeziometacarpal
ULM	Upper Limb Model
VDIM	Vague Discrete Interval Modeler
VR	Virtual Reality

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

INTRODUCTION 1

Have you used your hands to describe the shape of the bottle of the perfume you are using to help your partner find it in the perfume shop? Have you used your hands (or fingers) to show how large a fish you caught during your holidays? Have you shown with your hands to your kid how to bend the plastic clay to arrive at a pretzel? Have you ever thought that you could also interact with your design system by using hand motions? This latter might be felt as some science-fiction at first, but after all you might also realize that it is technologically not impossible. In this thesis I show you that this can be implemented having the insights, supporting theories and necessary technologies. Moreover, I will also point out that using natural hand motions is actually a fast and fun way of externalizing shapes, such as your perfume bottle, to express shape characteristics, such as the size of your fish, and to manipulate shapes, such as that of a pretzel.

As a first step, I introduce the specific problem that I have dealt with in this promotion research, and clarify the context and goals of dealing with this problem. I will discuss the research questions and the set of hypotheses that guided me to a comprehensive explanatory theory, as well as to technical solutions that are beneficial for designers and can be used in future advanced design support systems. I finish this first chapter with an overview of the methodological framework and the practical goals of this research.

1.1 Current trends

The use of design support systems is proliferating, and from time to time new paradigms and technologies appear in practice. The target area of research and development of computer-based design support systems is gradually shifting from detailed design to conceptual design (Catledge and Potts, 1996). Naturally, the expectations for computer support of conceptual design are fundamentally different from the requirements for computer support of detailed design (Zheng et al., 2001). Efficient support of creativity, collaboration of designers (Jayaram et al., 2004), semantic integration of information and knowledge, and handling uncertainty and incompleteness in modeling (Leidner, 2003) can be mentioned as the major issues for design researchers, system developers and product designers. It can be also observed that graphical input and output are gradually becoming separated from the computers, as more efficient, truly three-dimensional visualization technologies emerge (Horváth and Opiyo, 2007).

Today's commercial computer-aided shape design software tools typically employ two-dimensional input devices, such as mice and tablets. As output devices, flat displays are used. Considering that most of the shapes to be designed are three-dimensional, designers face the problem of a dimensionality loss. With the abovementioned two-dimensional input means they can draw in two dimensions only, and the depth information should be specified by special features of the used modeling software. As a feedback, they have a two-dimensional projection of the shape, which can typically be viewed by continuous rotation. This procedure takes enormous amount of time to design complex shapes, and therefore is supposed to be inefficient in the early phases of design.

Recently, natural interaction methods are being studied and developed (Hummels and Stappers, 1998), with a focus on the integration of natural human interaction into design systems (Lin, 2003). Research has been going on for several years to explore different natural input methods, like speech and gesture recognition, or head-and eye tracking (Djenidi et al., 2002). With the help of these technologies development of multimodal and real-time systems is needed (Corradini et al., 2002), which not only improve the communication between designers and design systems, but also support collaboration of designers (Hummels and Overbeeke, 1999).

Conceptual shape design strongly relies on the imagination and creative skills of designers. On the other hand, computers impose various methods and structures on the creative actions. A general issue for research is how to create a bridge between the design thinking of humans and the systematic modeling process imposed by computers (Krause et al., 2004). As for now, there is a gap between the speed of design thinking and its computer support, due to the prevailing hardware, software and information processing paradigms in Computer-Aided Design (CAD) (Rauterberg, 2006). In a creative process speed and intuitiveness are closely related, since the designer need immediate visual feedback to be able to externalize his ideas, moreover, to be able to continue with the creation of a brand new shape.

Many researchers have made an effort to understand the human thinking and action processes that support design creativity and efficiency (Arciszewski et al., 1995). Several explanatory and predictive models have been developed. One of them, introduced by Horváth et al. (2003b), tries to explain these capacities with the concepts of cognitive theory (Figure 1.1). It identifies an inner cognitive loop (including ideation, reasoning and presentation), and an external loop (including reasoning, presentation and constructive model development). Typically, the timeframe of the mental actions forming the inner loop is 10^{-1} – 10^{-2} seconds, while the time frame for the modeling actions can be as large as 10-100 seconds. This scheme suggests that the speed of modeling actions and the mental actions should be as close as possible, in order to not become a burden on creative thinking. In other words, it is expected from the user interface of Advanced Design Support (ADS) systems that the time needed for creating a component of a product model (e.g. a surface) should be about the speed of human thinking, and not be longer than approximately a second.

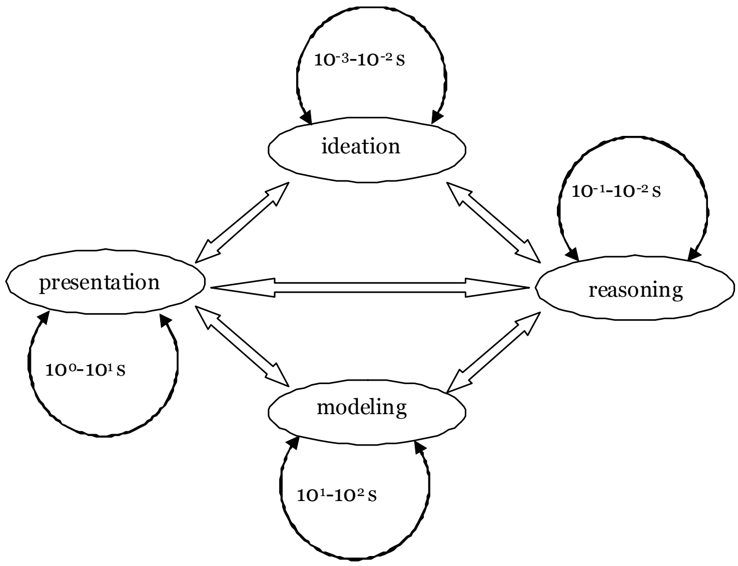


Figure 1.1 Cognitive scheme of conceptualization

This implies the use of bodily input issued naturally by humans, such as speech or gestures, which inherently fulfill the above expectations. *As a possible solution, hand motions are studied as input means for shape design systems in this research.* The advantage of hand motions over other input methods is the intensity of information they are capable to express. This intensity comes from the amount of information that can be obtained in a given time period. For instance, just imagine that defining a freeform surface takes many steps using two-dimensional input devices and even using three-dimensional input devices which are capable of defining points in three dimensions. A surface can also be generated using hand motions and by taking advantage of the changeability of the hand shape during the motion. The motion of the hand provides three-dimensional surface information directly, and using the three-dimensional space, surfaces can be generated at the required location immediately. As a consequence of the increase of information intensity, the modeling time is supposed to be decreased.

1.2 Background of research

This promotion research was conducted at the Faculty of Industrial Design Engineering as part of the *Methodology, Tools and Techniques* research program of the Faculty's research portfolio. The presented work belongs to the research sub-program called *Product Conceptualization in Collaborative Virtual Prototyping Environments,* running at the Section of Computer Aided Design Engineering (Figure 1.2). The title of the embedding research project is *Knowledge Intensive Conceptualization (KIC).* The work is also related to other research topics in KIC, such as vague discrete interval modeling (Rusák, 2003), verbal control of design systems (Kuczogi et al., 2002) and behavioral simulation (Van der Vegte and Rusák, 2007).

Research efforts in KIC have been made to explore human-centered methods and techniques for product conceptualization, including shape, structural and functional aspects. System frameworks and prototype tools are developed for knowledge-intensive computer support of the process of shape and functional conceptualization. In particular, advanced technologies for computer-aided conceptual design (CACD) are studied, for instance for multimodal system interfaces, highly interactive shape generation and manipulation, relations-based concept modeling, augmented prototyping (Verlinden et al., 2006), co- and dislocated collaboration support in virtual environments (Horváth et al., 2002) and information and knowledge management (Horváth and van der Vegte, 2003a). The KIC research program contributes to the scientific development of the discipline, and provides applicable solutions for product conceptualization and remote collaboration in the industry. A proof-of-concept implementation is under development, which contains hardware and software elements.

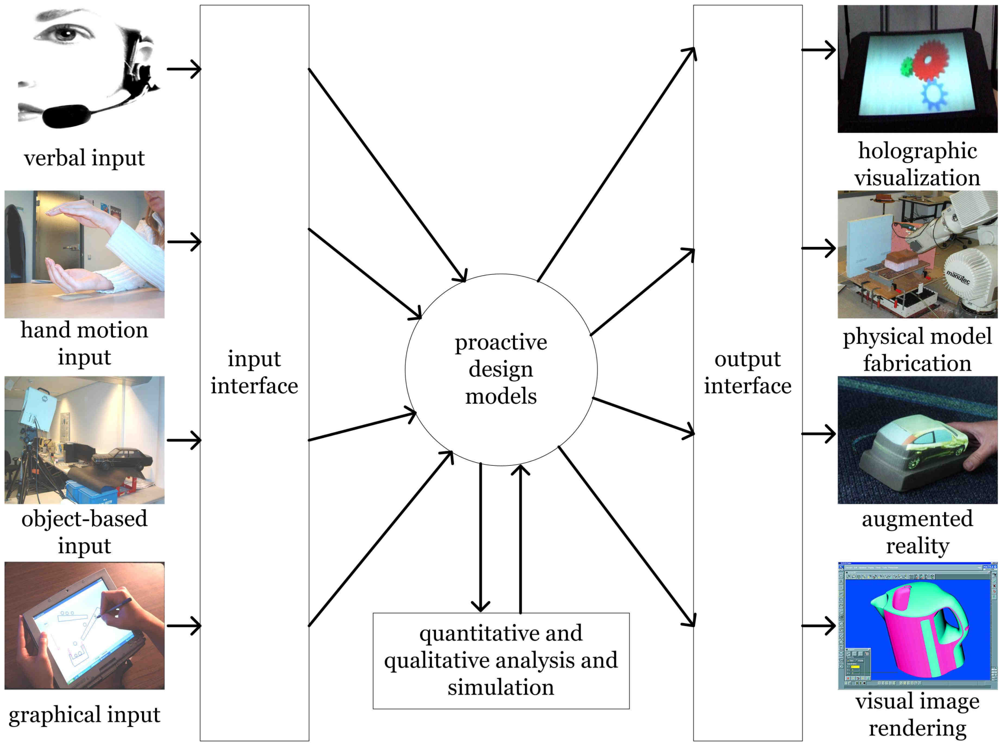


Figure 1.2 The concept of an advanced design support system, that is in the focus of the research at the Section of Computer Aided Design Engineering

The envisioned implementation of the advanced design support system (ADSS) enables interactive, truly three-dimensional design and simulation of products (Opiyo and Horváth, 2006). The main elements of this implementation are multimodal interaction, air-borne visualization (Opiyo et al., 2007), tactile/haptic sensing and interaction/behavior simulation. As primary interaction modalities, hand motions, verbal expressions and digital object scanning (Song et al., 2005) have been considered. The major hardware elements are a fast camera system for hand motion detection and a holographic imaging device for air-borne volumetric visualization (Figure 1.3). For haptic sensation, a string-suspended feedback device has been considered (Butnaru et al., 2007). Simulation of the interaction of humans with virtual objects, the interaction of virtual objects, and the behavior of virtual objects rely on sensor input and real time computation of multi-physical processes, respectively.

Figure 1.3 Proof-of-concept implementation of the advanced design support system

The common vision of the Section of CADE is a collaborative virtual design environment (CVDE), in which conceptualization and design of shapes are based on, among other things, a dedicated hand motion language. Groups of designers are assumed to work even at remote locations and to use hand motions in their collaboration (Horváth and Rusák, 2001), or alternatively verbal communication (Dorozhkin and Vance, 2002) and other advanced forms of conventional design and representation means (Lim et al., 2001) to externalize shape concepts. Designers jointly build and manipulate multiple shape variants in the distributed virtual environment, which provides true three-dimensional visualization and enables concurrent manipulation of shared shape models in real-time.

1.3 Problems of hand motion-based shape design

Looking at the problem of hand motion-based conceptual shape design, a vast amount of questions can be raised. Investigating the research problem, it can be seen that it involves multiple issues and relationships, and there are several possible points of departure. This section gives a broad view of the research problem, and briefly investigates the identified related fields of knowledge. The major fields and relationships and of knowledge are represented in Figure 1.4.

This figure shows that the problem can be approached from various fields, such as hardware technologies, information processing, computer modeling and simulation, human perception and cognition, and design methodology. The key to the successful completion of this research is to find a focus and the balance between the abovementioned disciplines, and to address the most relevant questions. Human, technological and application aspects were considered during the decomposition of the problem. Naturally, these aspects are related to each other and a single aspect cannot be discussed without taking into consideration the other ones. The decomposition of the problem can be seen in Figure 1.5.

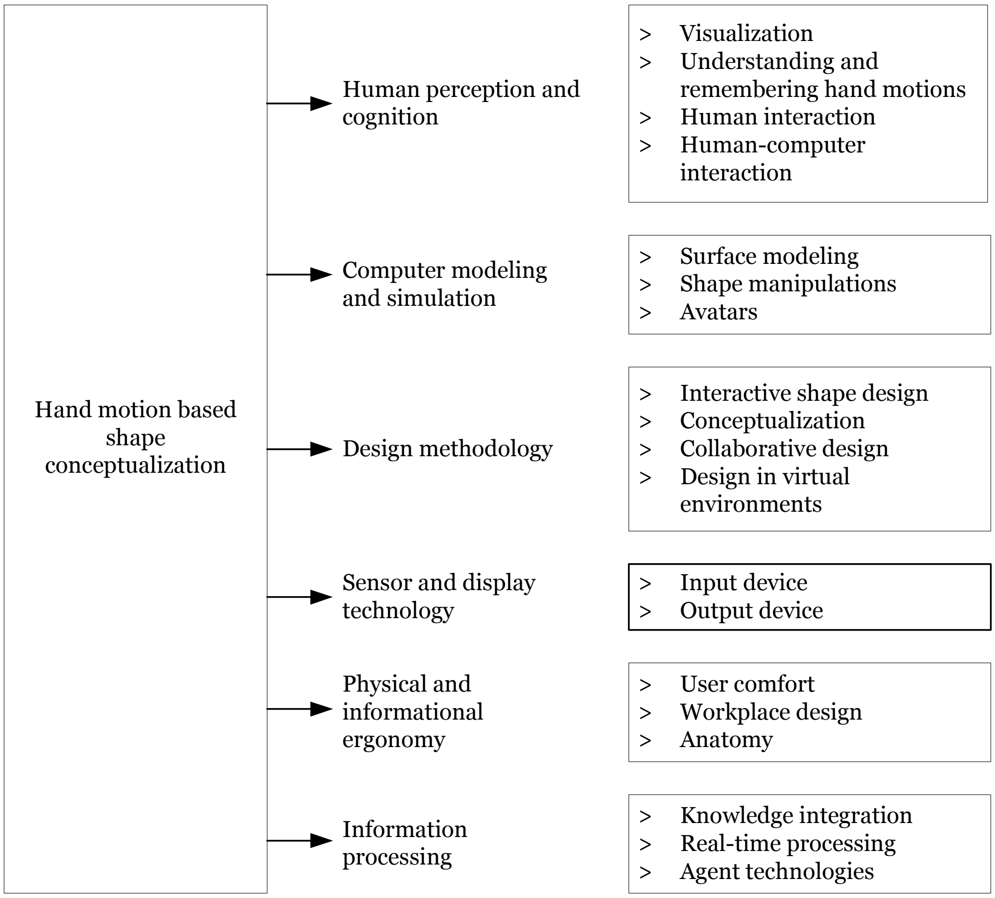


Figure 1.4 Multidisciplinarity: cognitive, technological and modeling aspects

Support of creativity and comfort can be mentioned as the main issues regarding the ***human aspect***. The human, as the user of the design support system, applies hand motions as a natural human capacity in order to indicate shape concepts. However, it is supposed that in order to facilitate creativity in the design process, the hand motions have to be specified in such a way, which is *intuitive* for the designers and at the same time unambiguous for computers. This can only be done by studying human perception and cognition in design. On the other hand, creativity might be also influenced by the technologies applied in the processing of hand motions. *Real-time processing of hand motions* is required to be able to provide the designer with immediate visual feedback. The design environment might also have an effect on the mental comfort of the user. If the designer is placed into an unnatural, technologically over-equipped environment, it might have influence on the feeling of comfort.

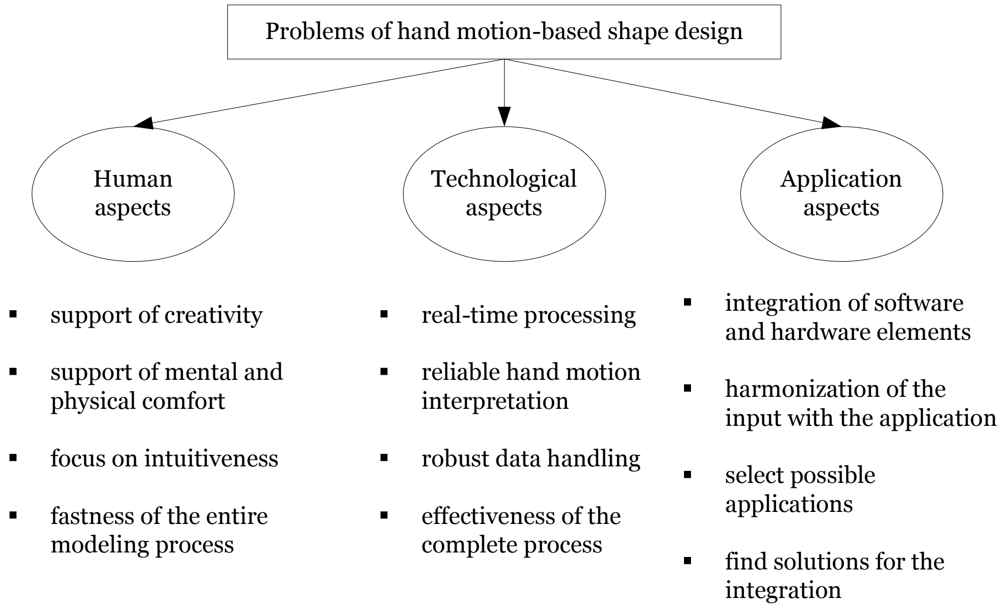


Figure 1.5 Sub-problems of research

When the *physical comfort of the designer* is considered, it also depends on the hardware, which might be attached to the designer and might put some limitations on his movement. The ideal design environment provides physical comfort, and its elements, such as tables, chairs or displays are arranged in a way that it is *ergonomically suitable* for designers. Most importantly, the motion envelope of the arms of the designer has to be taken into consideration. Anthropometry, *kinematical limitations of the arm- and hand movement* and speed of the hand motions can be mentioned as major issues to be considered.

Real-time, robust, reliable and effective processing of hand motions is the key ***technical aspects*** for the successful support of the design process. Hand motions go through a series of processing steps from the moment of commencing the very first modeling action with hand motions to the moment the intended shape element is visualized on the display. Speed was already mentioned as a key issue in the discussion of the human aspects. From a technological point of view, all the hardware and software elements are required to work in real time and the overall system is required to *provide immediate feedback* based on the integration of these elements.

It is worth mentioning that the hand motion input is supposed to be a powerful means in shape conceptualization. Nevertheless, what is useful for the designer, can be technologically problematic. *Maintaining uncertainty* is advantageous for the designer in several cases, but it is challenging to support with software tools.

At last, the problem of the development of a ***commercial version of the complex software tool*** should be mentioned, which *integrates all the elements* of hand motion processing. It is well-known that reaching a commercialized version of a software tool is a time-consuming and complex task going through several testing phases and improved versions. On the other hand, an analysis of *practical applications* is needed which may benefit from the integration of the new interface. Due to the fact that this research is highly explorative, I cannot and do not want to deal with these problems in this thesis, but the colleagues at the Section of CADE keep it in mind as an interesting and necessary topic for the future.

In summary, as it can be seen from the above description, even the specific scientific problems are far-reaching and diversified. With a view to the objectives of the promotion research and the available capacities, I focused on the information processing aspect of hand motion based shape design. Nevertheless, it was necessary to investigate some aspects of sensor and display technologies (to select the hand motion detection device) and computer modeling (to integrate hand motions into a shape design system which can exploit the capabilities of the hand motion-based input). The aspects of arm- and hand kinematics and ergonomics are only touched upon in this thesis and require further research in the future. The next section describes the actual focus and clarifies the bases and goals of research.

1.4 Focus of research

The problem of multimodal interfaces is complex (Latoschik, 2001), and in fact, little is known about the usability and utility of individual modalities in advanced design support systems. Therefore, in this research the emphasis was put on studying hand motions in conceptual shape design situations to be able to conclude about its merits and limits. To be more specific, this research focuses on

- the exploration of opportunities of detecting hand motions,
- the conversion of the detected geometric and kinematical information to representation of shape components and shape transformations, and
- on the testing of the proposed input against specific criteria of usability and utility.

Previous research indicated a trend, which is a stronger interest in the invention of new technologies for detecting and processing hand motions, than in the cognitive, semiotic and human aspects. In order to facilitate the effective utilization of the technologies, more efforts have to be devoted to human- and design-related issues. As far as the latter is concerned, there are several issues that are not at all or only partially studied until now. That is the reason why the development of a Hand Motion Language (HML) for shape conceptualization was put in the center of research (Horváth et al., 2003b). This promotion research focused on the investigation of its compliance to designers' expectations and to the tasks of conceptualization. Actually, the proposed hand motion-based interaction is supposed to support the fast input of three-dimensional shapes and shape transformations in advanced design support systems. To be able to conclude about the new interface in application, a proof-of-concept system was decided to be developed.

1.5 Initial set of criteria

As described in the previous section, a major goal was the development of a proof-of-concept system that enables hand motion-based shape design. Naturally, there has to be a set of criteria which the proposed system has to fulfill. The difficulty is that immediately an indispensable duality can be discovered regarding these criteria. As the goal was to support designers in the early phases of design, the following criteria were identified from a users' point of view. The system has to

- support fast input of three-dimensional shapes,
- provide immediate visual feedback on users' action,
- support intuitive interaction by hiding the mathematical description of shapes and shape manipulations from the user,
- provide an enjoyable interface for communicating shape ideas, and
- to allow the storage intermediate and vague shape ideas for later usage.

The above-listed criteria might sound simple or obvious from a users' perspective, however, from the perspective of the development of a working system, these criteria are sources of challenging research issues. For the sake of clarity, but without the complete analysis of these research issues, here are the criteria which have to be fulfilled – to be simple – just to make the system work. The system has to

- detect the motion of the users' hands,
- interpret the actions of users, and
- translate hand motions to shape descriptions or shape manipulation commands.

With these criteria in mind, the research questions are listed and discussed in the next section.

1.6 Research questions

Due to the complexity, novelty and multidisciplinarity of the problem, a large number of research questions could be formulated related to the abovementioned problems and intents. Summarizing the research problems, the main research question is:

- ***How can a working system for hand motion processing be implemented and tested for shape conceptualization?***

To be able to provide an answer, this complex research question had to be decomposed. In fact, two groups of research questions could be identified, namely theory-driven questions and practice-based questions. The first group raises questions about the underpinning theories and methods of hand motion processing, and the second group includes questions about the usability and practical utility of hand motions in conceptual shape design.

Theory-driven questions:

- ***What are the steps of data conversion that need to be done to arrive at a shape model from raw motion data?***

This research question addresses the complete process of information conversion and transfer in hand motion-based shape modeling. From the raw motion data, a virtual model of a shape has to be constructed. It was interesting to discover feasible solutions in the intermediate steps of hand motion processing, bearing in mind that the integration of these steps has to provide a real-time interface.

- ***Based on which decision mechanisms can the different hand motions be interpreted even if they are not always performed the same way?***

Even if a vocabulary has been prescribed, which was defined in the HML, the hand motions can be performed slightly differently every time. The hand motion interpretation has to resolve this problem of variability and offer reliable and robust interpretation of users' action.

- ***Is there a method which can be used to convert the hand motions to specifications of geometric entities and modeling actions in real-time?***

  Using the HML, modeling with hand motions is a continuous process of constructive (geometry-oriented) and manipulation hand motions. From the minimal motion information obtained during detection, a mathematical description of shape elements or information for the manipulation operation has to be constructed, respectively.

  Practice-based questions:

- ***How can information from the motion of human hands be obtained in real-time?***

  The importance of real time processing cannot be emphasized enough, as a major issue for the successful application of hand motions. Besides, the amount and the nature of the detected information is also a question.

- ***How usable is the developed proof-of-concept system in a conceptual design task?***

  It is interesting how the users react on the idea of hand motion-based interaction. A method needs to be found to collect data about the usability of the system developed for hand motion-based shape design. Specific criteria need to be studied and evaluated in order to conclude about the overall usability of the proof-of-concept system.

- ***How the generated shapes reflect the properties of the intended shape?***

  This question addresses the problem of comparing the actual generated shape and the shape that was intended to be created. Criteria for this comparison need to be found and tested in applications.

- ***What level of efficiency can be achieved by using hand motions in conceptual shape design in terms of the modeling process?***

  An interesting question is how the complexity of the shape influences the complexity of modeling. The types of shapes are needed to be identified which are particularly efficient or inefficient to be modeled using the current version of the HML.

  Many of the questions above can be covered by individual hypothesis; some of them however need to be further decomposed in order to provide matching answers.

1.7 Research hypotheses

Based on the research questions, the research hypotheses were set up, which can serve as starting points for explanatory and predictive theories for the addressed problems. The primary hypothesis is that *by extracting information from the physical motion trajectories of the human hands, information can be generated for the construction of the indicated shape elements of three-dimensional shapes as well as for shape transformation*. This projects ahead that the expression of shape elements by hand motions can provide sufficient information for shape conceptualization and manipulation in the process underpinned by the use of a predefined HML. The HML formalizes not only the input but also the outcome of the process and that of the shape conceptualization system, while leaving sufficient freedom for the designers. Figure 1.6 shows the hypothesized modules of the system for hand motion processing, and it is thought that, between each module, an appropriate information conversion method can be found taking into account the operational parameters of the hand motion detection, the recommended input of the hand motion interpretation process and the characteristics of the modeling engine.

As far as the information transfer from the motion trajectories to the computer modeler is concerned, it has been assumed that *obtaining motion information of some selected landmarks of the human hands provides sufficient information both for the interpretation of the HML words and for generating surfaces*. These landmarks are supposed to be selected by analyzing the words of HML in terms of those characteristic features, which provide information for interpretation and mathematical description of surfaces. It has been assumed that the minimum amount of information can be found by taking into account all of the abovementioned characteristic features and eliminating the redundancies.

Considering the interpretation of the HML words, it has been assumed that the *words of the HML can always be interpreted by applying a classification method, which based on the identified characteristic features of the hand motions, can find the matching one by searching in a hierarchical structure of the features*. The conjecture has been that based on a small number of characteristic features, all words of the HML can be identified in the continuous process of modeling in real time.

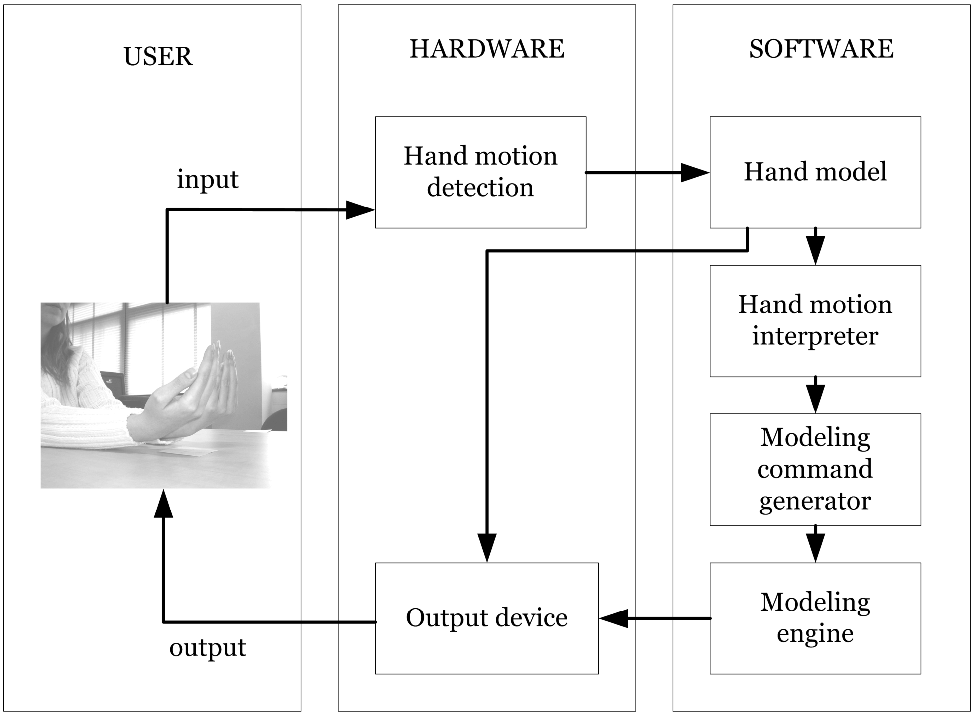


Figure 1.6 Modules of the hand motion processing system

Regarding the conversion of the detected information to a mathematical description of geometric entities and to a formal description of modeling actions, it has been assumed that *by studying the morphological characteristics of the detected motion trajectories, a method can be found by which point sets can be produced for the generation of surfaces*. In addition, it has been assumed that *by taking into account the mathematical descriptions of the modeling operations in the modeling engine, a sufficient amount of information can be constructed for each modeling operation by obtaining information from the detected motion trajectories*. Note that the information obtained should be in harmony with the data schemes describing the geometric entities or modeling operations in the modeling system, respectively.

To obtain information from the motion trajectories of the human hands, it has been assumed that *a passive detection technology provides the best-fit compromised solution in terms of the human and the technological aspects* listed in Chapter 1.3. This technology is also in harmony with the assumption above, that a small number of detected landmarks of the hand provides sufficient basis for further information processing. It has been also assumed that *by applying the selected hand motion detection and interpretation method a proof-of-concept system can be realized which is able to produce information for modeling* *arbitrary shapes by extracting information from the hand motion trajectories and by converting them either to geometric or to modeling command information (geometric information or to modeling commands).*

With regards to utility and usability of hand motions in shape design, it has been hypothesized that *utility can be evaluated by three factors, namely, fidelity of the generated shapes, accuracy of the manipulation operations and complexity of modeling. It has also been assumed that by analyzing the time spent on a design task, stimulation, cognitive load, operability, satisfaction, learnability and physical comfort, information for the usability of the overall system can be obtained.*

1.8 Research methods

The steps of the research in design context methodology were followed in this research. As described in (Horváth, 2007), the process of research in design context follows the scheme of the six-stage cycle of general research (Figure 1.7).

In a complex research project, the research steps may occur recurrently. Research in design context studies are revealing, empirical and analytical in nature. The goal is to explore, describe, understand, and explain design related phenomena, which occur naturally, or are partly or entirely related by design. The studied phenomenon is typically a set of behaviors of some entities such as humans, artifacts and surroundings. The context of research is defined by the goal of the research and the inherent relationships between entities, and is reflected in the selected research variables and the way of studying the relationships between these variables.

This research process involved two phases: an explorative and a confirmative part. The *explorative phase* was needed to discover current trends and available technologies. The *confirmative phase* of research addressed specific questions about usability and utility of hand motions in conceptual shape design, and answered them through the design and evaluation of experiments. As the proof of the pudding is in the eating, a proof-of-concept system also needed to be developed for hand motion processing. Actually, this system provided the platform for conducting the aforementioned experiments.

Six stages were carried out in the promotion research, as it is shown in Figure 1.7, the research process was decomposed to six stages, namely (i) exploratory study, (ii) forming the main hypothesis and sub-hypotheses based on the results of the exploratory study, (iii) establishing the theories of the HML based modeling interface, (iv) proof-of-concept implementation of the HML based modeling interface, (v) validation of the theories through the proof-of-concept implementation in application, and (vi) interpreting the results of research project.

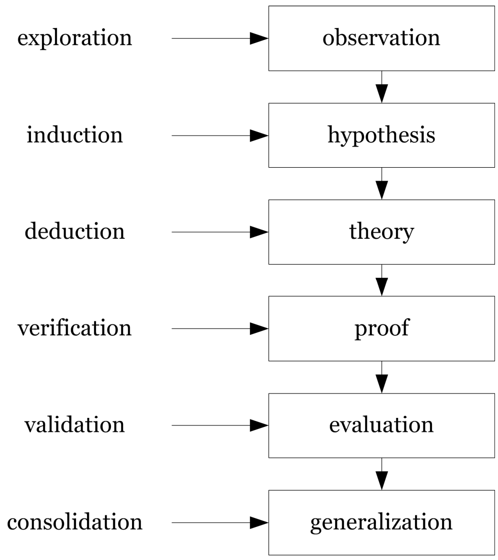


Figure 1.7 Major phases of research in design context

Literature review and search for adoptable technologies and commercially available and feasible solutions can be mentioned as the two applied methods in the explorative part of this research. They were done to get insight in the state-of-the-art technologies for hand motion processing. In the literature review the emphasis was put on the different approaches of hand motion processing, with regards to the amount of the detected information, the way of transferring this information to the modeling system and the relation of the human hands and the detection technology (if they are in contact with each other). In order to be able to build a foundational knowledge basis for the research and system development, I concluded to use an indirect processing of incomplete hand motions. However, further qualitative and quantitative analyses were necessary, with special attention on the applicable motion detection equipment. A comparative market search was performed to select the most fitting hand motion detection technology.

Logical induction was applied to establish the main research hypothesis, sub-hypotheses and questions. The main research question and its decomposition can be read in Section 1.3, and the main hypothesis and sub-hypotheses showing the direction and goal of the entire research project in Section 1.7.

Theory adaptation and brainstorming sessions can be mentioned as typical applied methods in the third phase of the research. *Concept synthesis* was used to integrate the applied theories. The fundamental theory and the functional framework of the HML based interface were conceptualized by brainstorming sessions. Various theories and software tools were investigated and analyzed about motion trajectory modeling and hand motion interpretation in order to have insight into the state of the art in these fields, and to adopt the applicable theories and tools.

The *development of the proof-of-concept system* for hand motion processing was realized to *test the established theories in an indirect way*. This included the design and implementation of the dedicated algorithms for each phase of hand motion processing. Existing software libraries and algorithms were studied for suitability for the intended application, to reduce the development time and to provide higher reliability and compatibility.

Experimental comparison of collected qualitative and quantitative data in real world-like tasks was the applied method for testing the usability and the practical utility of the theories and implementations. In the usability oriented experiment a *Likert-scale based questionnaire method* was used to collect information for a comparative statistical analysis of traditional CAD and hand motion-based modeling. In the experiment studying utility, the analysis of the technical parameters of the hand motion-based modeling system was done. Quantitative data were collected for statistical analysis. It has to be mentioned that again a *study of the related literature and standards* had to be performed to be able to derive the criteria and measures for studying usability and utility in the context of hand motion-based shape design.

The whole research was conducted and the obtained results were investigated logically in order to be able to *draw conclusions* about the values produced, and to point out its merits and limits. This serves as a good basis for planning future research.

1.9 Contents of Chapters

This thesis reports on the abovementioned topics in the following structure. Chapter 2 is an exhaustive literature review divided into three parts: study of the different functions of the hand, analysis of the state-of-the-art hand motion processing approaches with regards to information processing aspects and hardware detection technologies and assessment of applications using hand motion input.

Chapter 3 introduces the theoretical fundamentals, concepts and methods of using hand motions in shape conceptualization. The whole set of the HML is presented, and the theory of hand motion interpretation is discussed in details. Furthermore, the process of surface generation and manipulation is described.

Chapter 4 focuses on the realization of hand motion based shape conceptualization, with a special attention to requirements, functional specification, data structure and information flow. The connection of the HML to the Vague Discrete Interval Modeler (VDIM) as its user interface is also discussed.

Chapter 5 reports on the development of the proof-of-concept system, including the description of the applied hand motion detection technologies and the algorithms of hand motion interpretation.

Chapter 6 evaluates the usage of hand motions in conceptual shape design according to human and technical aspects. The first part of the chapter describes the details and evaluates the results of a usability study which was designed to gain information about user satisfaction and cognitive aspects. The second part of the chapter focuses on the evaluation of the quality of shape definition by hand motions. Quality is divided into fidelity, accuracy and complexity of modeling; and experiments were designed and conducted to quantify these measures.

Finally, Chapter 7 summarizes the research work, discusses the value of significance of the results, derives the main conclusions and recommends possible directions for future research.

LITERATURE REVIEW

A Study Of Hand Functions, Hand Motion Processing Approaches and Related Applications

2

I introduced the research problem in the previous chapter. It is diverse, it is multidisciplinary, but the focus was also clarified: information processing during hand motion-based shape conceptualization. More specifically, it is about the detection and interpretation of hand motions and the application and testing of hand motions in conceptual design. Now you might ask what has been already known in these fields. You might also think that there are some applicable solutions out there. Let's see! First please check back to Section 1.3, because I follow the structure of problem decomposition introduced there, and discuss the related literature concentrating on human, technological and application aspects of hand motion processing. The reasoning model for the literature study is seen in Figure 2.1. The sizes of the circles reflect the interest in research.

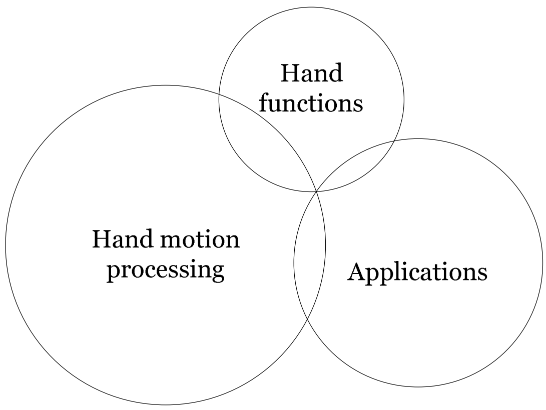


Figure 2.1 Aspects of literature study

I was mainly interested how other researchers approached the research questions I considered (Section 1.6), and I intended to find those operationally feasible solutions which could be adapted into my theories and methods. The goal of the literature study was to investigate and analyze the different opportunities offered by hand motion processing techniques. I found a large number of hand motion processing techniques, therefore I decided to group them into different categories, and analyze them accordingly. A reasoning model was built based on the main characteristics of hand motion processing technologies, and this served as a basis for classification. Each of the categories was analyzed taking into consideration *the requirements against a practical hand motion processing system for creative shape conceptualization.*

This chapter often contains the words "we" and "our", referring to the support of the colleagues at the CADE Section, and especially to the help of my promoter and my supervisor. This chapter was written based on the papers Varga et al. (2004a), Varga et al. (2004b) and Horváth et al. (2003b), by adding extra information that was necessary to be investigated during the progress of the research and also some relevant recent literature.

2.1 Survey on the knowledge about hand functions

Hand motions and gestures received a lot of attention in research in the last two decades because of the opportunities they offer for human-computer interaction (HCI). Many researchers believe that more natural and effective interfaces can be developed based on these resources for computer-aided design systems (Westeyn et al. 2003). One branch of research targets the technological platform, i.e., hardware and software systems for detection, recognition, interpretation and application of hand postures and signs (Pavlovic et al., 1997). Another branch of research deals with human aspects such as physiology, cognition, perception, and apprehension of hand motions and gestures (Wagner et al., 2004). Signs generated by hands have been studied as an individual instrument for communication between designers and design support systems, as well as part of a multimodal interface.

2.1.1 Understanding of hand postures, gestures and motions

The human hand is a biological mechanism, a versatile natural manipulator of the human body (Wu and Huang, 2001). As such, it exerts forces and produces motions that are used in controlled motor functions (Albrecht et al., 2003). Featuring polymorphism, human hand has generic characteristics based on which it is treated as a genotype in research, except when the amplitude of variation and the specific features of an actual phenotype are studied. In our terminology genotype means the generic constitution of what is called human hand, without taking into consideration any instance characteristics. On the other hand phenotype means that a given example of the human hand reflects individual or small group features, and they dominate its observable set of features. The possible changes and most of the observable features of the hand's shape are determined and constrained by its physical articulation (Elkoura and Singh, 2003).

Although often used as synonyms, the terms hand postures (Lee and Kunii, 1995), motions and gestures (Kim and Chien, 2001) have different meaning in our terminology. Hand postures are understood as individual formations of the hand without movement. Usually classified as one-handed, two-handed and double-handed; signs are manifestations of hand postures in various positions. Brought about by the arm, hand motion is a change in the spatial position of the hand and means a particular manner of moving the hand. While hand postures involve normal and hyperextension, flexion, palmer and radial abduction, ante- and retroposition, hand motions are combinations of hand postures and controlled movements of the hand in space. Motion of the hand enables a variety of activities, but each time obeys certain kinematical constraints. Human gesture is typically an action to convey certain indication and evoke a response. Hand gestures are combinations of hand postures and dynamic hand movements (Nam et al., 1999). They are used to express thoughts and emotions, emphasize speech, and indicate intent and attitude. In general, hand motions carry less semantic content than hand gestures (Cont et al., 2004), but they are more powerful in carrying out actions.

Human motions, including hand motions, are typically processed based on instrumented detection (Yu et al., 2000) or computer vision (Moeslund and Granum, 2001). One of the most challenging problems is to extract hand motions from complex views (Triesch and von der Malsburg, 1996). Instrumented detection can be enabled direct sensors, e.g. data gloves that must be worn by the user and attached to the computer. Alternatively, it can be done by indirect trackers and scanners that leave the hand naked but introduce difficulties in real-time recognition (Dourish, 2001).

The functions of the human hands have been sorted in *manipulation*, *indicative* and *descriptive* categories. Investigation of the hand in these functions has been in the focus of researchers for a long time (Aggarwal and Cai, 1999). Manipulation functions exploit the motor capabilities, that is, the ability of exerting force and adapting the shape in order to grasp objects. Indicative functions are for generating signs and gestures with the goal to communicate concepts and actions. Descriptive functions simulate an action of description by changing the form and position of the hand. A clear-cut separation of the functions is difficult to make since in general one of the functions is dominant, but it is accompanied by the other two functions. For instance, in the case of handwriting, the manipulation function dominates (grasping the pencil), but it is to enable the descriptive function (formation of lines). Writing can be recognized through its indicative function, in other words, based on the typical posture the hand takes while it writes.

2.1.2 Research in manipulator, indicative and descriptive functions of hands

Hand motions can be described on the basis of kinematics and kinetics (Kölsch et al., 2003). Kinematical description considers the geometry, position, orientation and deformation. Hand motions have been classified and described as rigid and non-rigid motions. Non-rigid motions are further classified as general and constrained motions (Kambhamettu et al., 1994). General motions are fluid and elastic motions, constrained motions are conformal, homothetic, iso-metric, quasi-rigid and articulated motions. Human hand motion is typically studied as articulated motion (Wu and Huang, 1999). Kinetics considers forces, moments and torques in generating movements. Based on visual investigations, (Gavrila, 1999), the human hand has been modeled as a multi-DoF rigid body system, (Huang, 1990), and deformable body system (Heap and Hogg, 1996). To consider the rules, constraint-based modeling, (Lee and Kunii, 1993), and knowledge-intensive animation of hand grasping (Rijpkema and Girard, 1991) have been proposed. Tracking of the positions and orientations of the hand can be by vision-based and non-vision-based methods such as magnetic, acoustic, and inertial tracking. Another branch of research are concerned with reconstruction of hand motions in virtual (animated), (Moccozet, 1996) and physical forms (Badler et al., 1991).

Focusing on semantic aspects, research in hand gestures studies the (i) formation of hand gestures (Eisenstein et al. 2003), (ii) recognition of hand gestures (Lamar, 2001), (iii) interpretation of hand gestures and (iv) conversion of hand gestures to commands of, for instance, shape modeling systems (Sturman, 1992). Hand gestures have been classified as (i) symbolic (hand posture indicating concept or object) or modalizing (following speech), (ii) pantomimic (representing interaction), (iii) iconic (representing object) (Sowa and Wachsmuth, 2002), (iv) deictic (expressing feeling or metaphor) and (v) self-adjuster (emphasizing significance, unimportance, or stimulation) gestures (La Viola, 1999). Two modes of gestures are delineated: gestures as a sign language and gestures as a spatial navigation.

A major field of research is sign language recognition. Human sign recognition means understanding intuitive signs (e.g. pointing finger) or professionally used signs (e.g. by signalmen at airports). Interpretation of the latter is easier due to its formalization. Gesture recognition is a wider field than sign recognition. The basis of extracting the meaning of gestures is the visual image of the hands. Recognition can be interactive or automated. Automated recognition of signs and gestures needs two processes: (i) an observation process based on sensors and (ii) feature classification for extracting gestures (Holden and Owens, 2001). Typical techniques are pattern matching (Tamura and Kawasaki, 1988), feature extraction (Imagawa et al., 2000), model matching (Shimada et al., 1995), and interactive learning (Lee and Xu, 1996). Tobely et al. (2000) applied a randomized self-organizing map algorithm for dynamic recognition of hand gestures with normal video rates. Hidden Markov functions have also been applied to recognize hand gestures (Nam and Wohn, 1996).

Descriptor functions are related to the use of hands to point at objects, indicate a point in space, designate a domain in space, emulate an analogy of something, or sweep following a trajectory. Researchers have been studying the nature and features of these hand motions, for instance, in two- and three-dimensional sketching. The characteristic motions are sensed, identified and the content describing shape- and shape-related information is extracted. For detection and recognition of hand motions both real-time and posterior technologies have been implemented and tested. Real-time hand motion recognition technologies involve a motion sensing process where the features of a motion are extracted from the input data. Both the principle of active signaling (e.g., data gloves) and direct detection (e.g., laser scanning), (Ahmad, 1995), have been used to obtain information from hand motions. Posterior technologies have been developed based on passive data extraction technologies such as image processing. Two-camera systems represent the conventional technology (Abe et al., 2000). Researchers have tried to take the advantage of having specific features in applications, and proposed dedicated solutions such as silhouettes-oriented multi-view tracking (Delamarre and Faugeras, 1999), visual tracking with occlusion handling (Lathuiliere and Herve, 2000), and processing in a contextual relaxation scheme (Chen and Huang, 2000).

2.2 Human-computer interfaces including hand motions

2.2.1 Multimodal interfaces

An evergreen topic of human-computer interaction research is multimodality (Arangarasan and Phillips, 2002). It has been also considered important for the user interface of future computer-aided conceptual design systems and what is hoped is that integration of the individual natural modalities can increase both the semantic level, and the efficiency of the interaction (Biermann et al., 2002). Several modalities are studied individually or as part of a multimodal interface, such as speech (Wolf and Bugmann, 2006), hand gesture- or motion processing (Kettebekov and Sharma, 2001), head-and eye movement (Tanriverdi and Jacob, 2000), tactile information processing (Bordegoni and Cugini, 2007), haptic interaction (Seth et al., 2006), facial expressions (Truong, 2007) or recently brain signal processing (Gnanayutham and George, 2007).

The applied modalities are selected based on the intended application. The most often used modalities are voice- and gesture control and control based on head- and eye movement. Most interfaces were designed to support impaired people in their communication both with other humans and computers. Sign language recognition can be mentioned as a typical application (Arendsen et al., 2007). Haptic interaction has been mainly proved to be useful to improve task performance in medical training, mechanical assembly in virtual prototyping (Galopp et al., 2007) and computer-aided styling. Up to now, brain-computer interfaces are meant for people with traumatic brain injury. Interfaces are often grouped as obtrusive or unobtrusive according to the relationship of the signal detection device and the human body. Brain-computer interfaces are of two types, invasive (signals obtained by surgically inserted probes inside the brain) and non-invasive (electrodes placed externally on the body).

Processing of multimodal information has four major steps, namely (i) signal detection, (ii) feature extraction, (iii) recognition/interpretation and (iv) fusion, where the individual modalities are combined. As far as information fusion is concerned, we can talk about feature-level and decision-level fusion (Russ et al., 2005). While feature-level fusion is considered to resemble human-like interpretation and combines features detected from individual modalities, decision-level fusion processes the different modalities separately and the information is fused when the separate decision-making processes give outputs.

In conceptual shape design speech, body- and hand gesture/motion processing and haptics has been considered lately (Arangarasan and Gadh, 2000), however, little is known about the usability and practical utility of the individual modalities. Therefore, we focus on the investigation of hand motion processing techniques as an effective form of shape externalization.

2.2.2 Hand motion detection with regards to model building

If real-time generation, visualization and manipulation of virtual surfaces by hand motions is concerned, our observation is that the most important efficiency issue is the quantity of information that should obtained from the detecting device and processed by the computer-based system. Less information typically goes with faster processing, but might reduce the fidelity of the generated shape and, hence, may need more work on the side of the designer. For instance, when the trajectory of a single point is tracked, creation of a surface needs two extra actions. First a curve defined should be defined by the user, which is called the generatrix of the surface. Then, a second curve, the directrix, should be defined, on which the generatrix moves along to generate the surface (Weimer, 1989). Another alternative is to specify a surface by a closure curve of the surface, but it cannot be filled in automatically due to the lack of morphological information. It seems obvious to obtain as much information from the hand motion detecting device as can be at all, but it (i) is limited by the functional capabilities of current motion detectors/scanners, (ii) extends the time of detection and scanning, and (iii) increases the amount of information to be processed for shape generation.

Usually, hand-held devices (Keefe, 2001), or a finger of the hand (Abe, 2000) is tracked to register three-dimensional points and to create three-dimensional curves this way. In this set-up, the designer first has to develop an idea of the surface to be created and then to decompose it into the abovementioned geometric entities. More information can be obtained if the motion of the fingers is also tracked and if the shape of the hand is also taken into consideration in the surface generation process. In this case surfaces can be generated by a sweeping movement of the hand (Dani, 1997). Modification of surfaces can be achieved by deforming them by means of changing the hand’s posture (Ma, 1997).

2.2.3 Hand motion recognition

Based on hand input, surfaces can not only be generated, but also manipulated by certain hand postures or by performing motions (Nishino, 1997). Obviously, the hand postures and/or motions have to be recognized and converted to modeling commands. Several methods were proposed to solve this problem (LaViola, 1999), however, most of them suffer from various limitations. For instance, some methods restrict the number of signs to a small set (Matsumiya, 2000). Others have problems with real-time computing (Pavlovic, 1997), or constrain the hand motion (Utsumi and Ohya, 1999). The abovementioned limitations spoil the naturalness and intuitiveness of shape generation and pose limitation on the user-system interaction.

In every-day communication, hand motions are used to give emphasis to the elements of verbal communication and/or to express concepts and information that is more straightforward this way than by words. Typically, a sequence of hand motions is performed and each hand motion represents a continuous unit of communication. However, this unit contains not only that part of motion which is expressing the information to be communicated, but also some transient starting and finishing phases which are actually not conveying useful information from the point of view of recognition. These phases should be identified and cut off in the recognition process. For this reason, the process of hand motion recognition has been extended with a sub-process, called segmentation. The segmentation sub-process identifies the limits of the starting and finishing phases of a specific hand motion, and provides this information for the recognition process. Various techniques have already been proposed for segmentation. Some of them analyze the entire trajectory of motion, by looking for changes in so-called motion descriptors, such as velocity or acceleration (Aggarwal and Cai, 1997). Other methods track the changes of some specific, body- or hand-related features e.g. position, orientation or posture (Nam and Wohn, 1996), (Liang and Ouhyoung, 1998). Most of the research done in motion segmentation relates to image processing, where motion segmentation aims at segmenting images into semantically significant parts (Borenstein and Ullman, 2002) (Konishi et al., 2003).

Based on studying the related work, we concluded that the hand motion interpretation method, which includes recognition and segmentation, has to meet two main requirements, namely it should be able (i) to recognize a moderately large set of hand motions and (ii) to complete data processing in real-time. As output means, those devices are ideal, which do not place the user into an immersive virtual space, and in case of collaborative work, those which give visual feedback to the users according to their position.

2.2.4 Visualization

When the designer generates and manipulates surfaces, he needs immediate visual feedback to be able to maintain spatial orientation. Taking into account, that the surfaces are generated by hand motions, it is the most advantageous if the visualization of the surfaces and the hand motions happen in the same workspace. As a result, users can see the image of the generated surface directly under their palms.

a. b.

Figure 2.2 (a) Stereo visualization (courtesy of McMains et al., 2003), (b) Volumetric display (courtesy of Grossman et al., 2004)

This can be effectively supported by the usage of three-dimensional displays (Van Orden and Broyles, 2000). For three-dimensional visualization, stereoscopic displays (Halle, 1997) or volumetric displays can be used. Stereoscopic displays provide two different images for the eye to generate a three-dimensional view (Figure 2.2 a). Head-mounted displays capable of three-dimensional visualization work in this way (Schkolne et al., 2001). One way of volumetric imaginary is the swept volume technique. The three-dimensional volumetric image is generated by sweeping a semi-transparent two-dimensional image plane around an axis.

In (Grossman et al., 2004) 198 two-dimensional images are uniformly displayed around the center axis, resulting in 116 million voxels. The display has a physical enclosure (Figure 2.2 b), which hinders the user to touch the virtual object and it can be viewed from any direction. Electroholography facilitates the computer-based generation of holograms in real-time (Lucente, 1997) and the interaction with them (Plesniak et al., 2003). Because of the large amount of data that has to be processed, interactive holographic displays have limitation in size and resolution at this moment (Bimber, 2004).

We concluded from the study of the related work, that the amount of the obtained information highly influences the method of surface generation. In this research, we only take into account those methods, which consider the whole hand shape, because these serve the intuitive creation of surfaces. When recognizing the hand motions, the method has to meet two major requirements, namely handling large set of hand motions and real-time processing. As an output, those devices are ideal, which do not place the user into a virtual space and give visual feedback to the users according to their viewpoint.

2.3 Analysis of hand motion processing approaches

2.3.1 Research related to the role of hand motion processing in multimodal interfaces

Human-computer interaction (HCI) is an evergreen, but still somewhat neuralgic issue of computer-aided design (Dix et al., 2001). For a rather long time, one-dimensional input devices (e.g., alphanumeric keyboards) and two-dimensional positioning devices (e.g. mouse) determined the way of entering shape information to design support systems. It has been recognized that these conventional input devices pose many, actually too many, constraints when applied in computer-aided conceptual design systems, in particular, in creative shape conceptualization systems. Through the past two decades, HCI has pursued a broad and ambitious scientific agenda, progressively integrating its research concerns with the contexts of system development and usage (Carroll, 1993). This has created an unprecedented opportunity to manage the emergence of new technologies so as to support socially responsive objectives.

Multimodality has been considered important for the user interface of future computer-aided conceptual design systems (Oviatt and Cohen, 2000), (Flippo et al., 2003), (Chu et al., 1997). A class of collaborative design systems seeks to provide multimodal input possibilities for the collaborating design participants to enable them to hold virtual shape conceptualization sessions. What is hoped is that integration of, for instance, voice control (Dorozhkin and Vance, 2002), (Kuczogi et al., 2001), (Grasso et al., 1998), hand gesture and motion processing (Rieger, 2003) (Aggarwal and Cai, 1999), physical object scanning and reverse engineering (Várady et al., 1997) can increase both the semantic level and the efficiency of the interaction and shape modeling.

In the area of computer mediated shape conceptualization, which features vagueness and incompleteness, human hand motion based shape externalization and modeling can play an important role. Hand motions might be a natural way for designers to express their shape concepts for computers, and to communicate with each other during shape conceptualization. Therefore, it is interesting to study it as a new descriptive input device.

Most survey papers published earlier addressed the problem of hand motion processing (HMP) from a pure technological point of view (Watson, 1993)

(LaViola, 1999). Others concentrated only on camera-based recognition (Pavlovic et al., 1997) or on contact methods (Hand, 1997). In our literature study, we investigate HMP from the aspect of real-time information extraction and conversion, which is required for creative shape conceptualization. Our aim was to analyze the available technologies and to compare them based on their operational parameters and their methods of information processing.

The next section introduces a reasoning model based on which the technologies have been sorted in four categories and the investigation of the conformance has been systematized. The following sections identify and analyze the most important technologies and processing approaches. Section 2.4 compares the various HMP techniques taking into consideration the requirements originating in shape conceptualization.

2.3.2 Categorization of hand motion processing technologies

The reasoning of our classification of hand motion processing technologies can be seen in Figure 2.3. In order to be able to reconstruct surfaces swept by hand motion in the three-dimensional space we have to extract sufficient amount of information from the posture and motion of the human hand. Consequently, extraction of information has been chosen as the primary aspect of HMP. It can be carried out in several ways. Human hands can be completely scanned, or some characteristic points (such as landmark points or silhouette points) can be detected. These two ways of obtaining shape information from the moving hands can be identified as complete and incomplete information extraction. The *way of information extraction* represents the first dimension in our reasoning about technologies.

A second aspect of HMP is the *way of transferring information* from the physical space, in which the hands are moving, to the virtual space, where the shape is modeled. It is also a dimension in our reasoning. The information transfer can be direct or indirect. Direct transfer means that the positional and motional information obtained from the hands is directly sent to a geometric modeling system, which generates the visual representation of the swept surfaces, for instance, in the form of point clouds. We talk about indirect transfer when the obtained information is first fed into an intermediate hand model with the aim to generate extra information, and an extended set of information is transferred to the geometric modeler and processed as before.

According to our reasoning, the third aspect of categorization is the *relationship between the hands and the information extracting devices*. Certain devices are mounted on or touch the hand, while other devices can extract information at a distance. These relationships have been described as contact or non-contact. This aspect has special importance in our research since our intention is to avoid the negative influences of the contact technologies on the comfort and creativity of the designers who interact with the hand motion based shape conceptualization system.

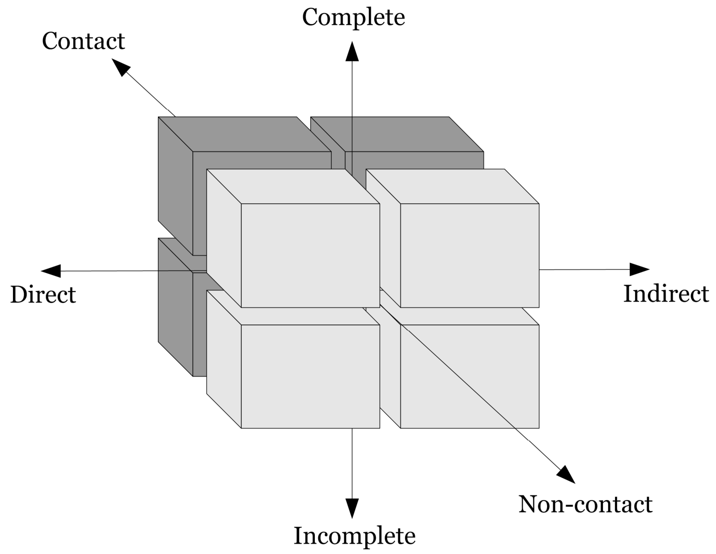


Figure 2.3 Categorization of hand motion processing technologies

Based on the above introduced framework of reasoning, we can sort the various HPM technologies into four processing categories, which have been called (i) direct incomplete, (ii) direct complete, (iii) indirect incomplete and (iv) indirect complete. Again, we emphasize that the major difference between the categories is whether an intermediate hand model is generated or not. We are talking about indirect data transfer, when an active hand model is used to extend the detected data for a better representation of the swept surfaces or for a better mapping of a manipulation action. A large quantity of contact and non-contact technologies belong to each of these categories.

In the next sections we focus only on those technologies, which can be applied in a hand motion based shape conceptualization system. In addition to the functionality and the operational parameters, we were also interested in the experiences related to the reported applications. More specifically, we investigated (i) which categories give the best opportunities for HMP and (ii) which technologies are the most advantageous for HMP within each category. In Section 2.4 we will investigate the compliance of the various technologies with hand motion based shape conceptualization.

The direct incomplete form of HMP technologies can extract or is used to extract only a limited amount of information from the moving hand, which is not sufficient to reconstruct a swept surface. The information from the moving hand is directly transferred to the geometric modeling system where the intended shape in virtually reconstructed. It is usually a curve. It is supported by both the contact technologies and non-contact technologies. The contact technologies can be divided in two subcategories on the basis of they are held in the hand, or they are mounted on the hand.

If sufficient information is collected, the swept surfaces and curves can be regenerated with high fidelity. If this extensive set of information is directly transferred to the modeling system, fast response times and real-time interaction can be achieved. This is the philosophy behind the direct complete hand motion processing technologies. It cannot be implemented by image processing based technologies, unless additional information is provided. At the same time, the technologies for scanning fast moving objects are still in their infancy. This is probably the reason why we could not find relevant non-contact technologies in this category. Nevertheless there are contact technologies, which can extract a large amount of data from the moving hand.

Indirect incomplete HMP processing technologies can extract only a limited amount of information from the moving hand due to the inherent technological limitations or due the need of a short cycle time. A hand model is used to generate the missing information that is not extracted from the moving hand and to support the construction of the intended shape by the geometric modeling system. We consider these hand models as active models, since they are activated and refreshed whenever new information from the moving hand is obtained.

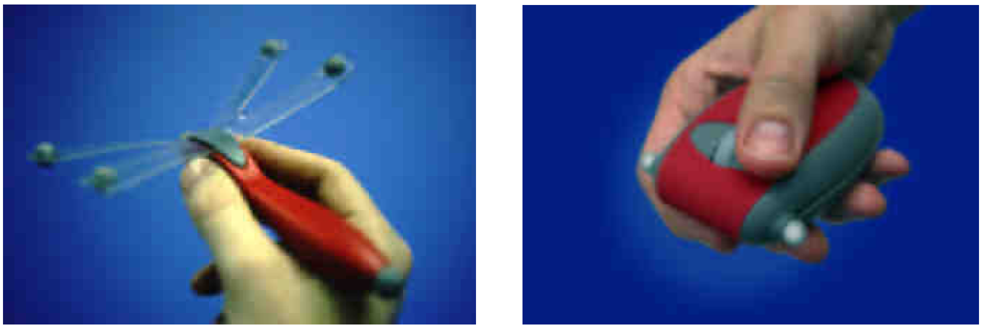
The indirect complete approach is based on technologies that are capable to extract sufficient information from the moving hands; nevertheless, it also uses a virtual hand model in the process of regenerating the swept surfaces. Obviously, the parallel use of these two information sources results in a redundancy in terms of data. On the other hand, some technologies are capable to reconstruct the whole hand surface. We consider these hand surfaces as passive hand models, since they are generated during the detection process. As we do not know anything about the configuration of the hand, another process is needed to extract the required information. For example, if we want to extract the points from the palm side of the hand, it requires further analysis.

The subsequent sections discuss the abovementioned categories of hand motion processing in the following order: (i) direct incomplete approach using contact technologies, (ii) direct incomplete approach using non-contact technologies, (iii) direct complete approach using contact technologies, (iv) indirect incomplete approach using contact technologies, (v) indirect of hand motions with non-contact technologies, (vi) indirect complete approach using contact technologies and (vii) indirect complete approach using non-contact technologies. The direct complete approach using non-contact technologies is missing from the discussion, because we could not find any references in this category.

2.3.3 Direct processing of incomplete hand motions with contact technologies

A typical representative of this category is a mouse, which supports two-dimensional positioning tasks. When included in a graphical user interface it facilitates interactive drawing in two dimensions. For this functionality other two-dimensional input devices have also been developed, for instance, tablets. Efforts have also been invested in technologies that are capable for three-dimensional drawing. They try to develop a digital three-dimensional equivalent of drawing on a piece of paper. First we analyze some of those proposals and implementations that convert the two-dimensional drawing functionality of a mouse to a three-dimensional drawing functionally. With these technologies the user can navigate in a three-dimensional virtual or physical modeling space and can generate or select points in three dimensions.

An early effort was made by (Clark, 1976) to design freeform surfaces in three dimensions. He intended to position the control points of a B-spline surface in the space using a three-dimensional wand, and to generate the surface accordingly. With the evolution of this type of technologies, the focus of research shifted from the functional aspects towards other aspects such as cognition, ergonomics and efficacy. The ergonomics considerations related to three-dimensional input devices put the technology development in a different context. The consideration of intuitiveness and creativity brought the issue of application of natural two-handed gestures (Leganchuk et al., 1998) in the focus of research. The experimental JDCAD (Liang and Green, 1993) and THRED (Shaw and Green, 1994) systems represent the first practical technologies. They apply two three-dimensional position and orientation trackers with three buttons, one for each hand. These devices, called 'Bat', support spatial geometric modeling tasks, such as modeling mechanical components or designing surfaces.



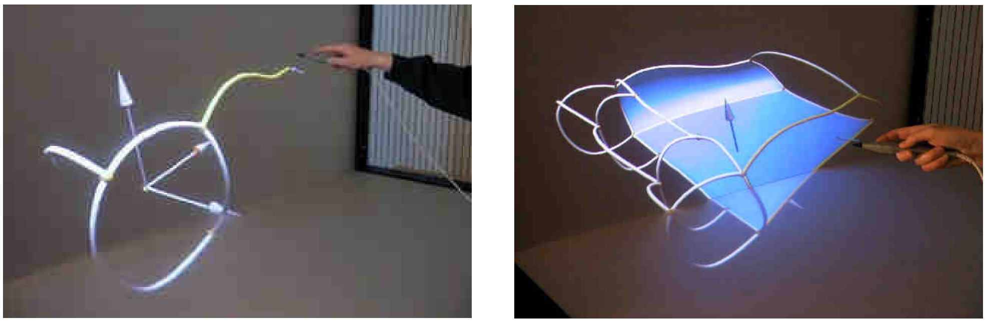
a. b.

Figure 2.4 (a) The "Dragonfly" in the dominant hand, (b) The "Bug" in the non-dominant hand (courtesy of (Stefani and Rauschenbach, 2003)

Based on anthropometrics and ergonomics considerations, left and right hand operations are defined and executed according to the importance of the tasks. That is, the most frequently used operations can be done by the dominant hand. The 'Frog', proposed by (Gribnau and Henessey, 1998), is a mouse-like input device with two buttons. A six DoF magnetic tracker was built inside to measure the Frog's location and orientation. The 'Bug' (Figure 2.4 b) and the 'Dragonfly' (Figure 2.4 a) are similar input devices using optical tracking (Stefani and Rauschenbach, 2003). The Bug has two retro-reflective spheres for the identification and the position detection of the device. The Dragonfly has six retro-reflective spheres to facilitate the detection of its position and orientation. The user can navigate in the virtual environment with two hands, one holds the Bug, and the other holds the Dragonfly. The latter has a virtual ray coming out of it, and the former has buttons. With this combination the problem of menu selection in three dimensions is solved. The user can point with the ray to a menu item and then press the button to select it. A virtual object can be connected to the Dragonfly to follow its motion while the button on the Bug is pressed.

With the help of these devices users can directly input three-dimensional data into computer systems in real-time, and the modeling time can also be reduced. However, these devices were developed following the navigation principles of traditional computer-aided design systems. Conventionally, the user controls the application with menus, and changes the shape of objects by clicking on menu items or icons, and the result of the operation is usually displayed on a monitor.

Freeform objects require the designers to define control points and then the system interpolates or approximates them to form a curve or a surface.



a. b.

Figure 2.5 The FreeDrawer system in use (courtesy of Wesche and Seidel, 2001) (a) Drawing a curve, (b) Filling in a surface

This approach however involves difficulties in the case of a three-dimensional design tasks, requiring split attention. The control points should be selected in three dimensions and the menu items and icons in two dimensions. The former is difficult since the users do not have appropriate spatial visual feedback on the two-dimensional monitor. The latter is also difficult because of the inherent three-dimensional characteristic of the input device. To solve the visual feedback problem, researchers started to apply head-mounted displays (HMDs), which place the user into a virtual environment (Clark, 1976). It has been observed, however, that psychologically and ergonomically this is not the best solution since the designers have no contact with the real world while wearing the HMDs (Wesche and Seidel, 2001) (Keefe et al., 2001). Research has also been done towards better solutions for control, and three-dimensional navigation elements such as the three-dimensional cursor and menu (Liang and Green, 1993) and the three-dimensional color picker (Keefe et al., 2001) have been proposed.

To come through the difficulty of continuous clicking when the user wants to draw a freeform curve, a new device was developed by (Wesche and Seidel, 2001) for their FreeDrawer system (Figure 2.5). It comprises a stylus by which the user can input points to the computer system by simply moving it in the three-dimensional space. The orientation of the stylus can be also measured. This solution is advantageous in artistic applications, likewise the CavePainting system (Figure 2.6), in which the user moves a tracked brush in the three-dimensional space and strokes are created according to its motion (Keefe et al., 2001).

a.

b.

c.

d.

Figure 2.6 Images of the CavePainting system (courtesy of Keefe et al., 2001) (a) Painting with the paint brush, (b) Painting with the bucket, (c) The painting table interface, (d) A result

In the CavePainting system the users can define so-called strokes by moving the brush by the hand, and it is similar to painting on a huge canvas. While the dominant hand holds the brush, the user can wear a pinch glove on the non-dominant hand to trigger some functionalities of the system.

The two technologies investigated above belong to the subcategory of technologies based on hand held devices. With this type of interaction many drawing tasks could be successfully realized in the three-dimensional space. The mouse-like devices and the stylus measure either the position, or the position and orientation of the global hand together, but do not care about the local finger motion. There must be a permanent grip and support provided, and, in addition, the shape of the hand can be hardly changed when holding these three-dimensional drawing devices. By tracking one point in the three-dimensional space a freeform three-dimensional curve can be created, but generation of a freeform surface is challenging. This is true for those approaches where drawing in three dimensions is realized based on detecting the motion of a fingertip. These technologies can support only simple navigation and manipulation tasks. Consequently, they cannot be considered in processing of hand motions in shape conceptualization.

Aligned with many more researchers we claim that a better solution can be achieved with non-contact technologies. Typical implementations of these technologies are based on computer image processing or computer vision. In these implementations image sequences are taken from camera(s) and analyzed by image processing algorithms. We investigate these technologies in the next section.

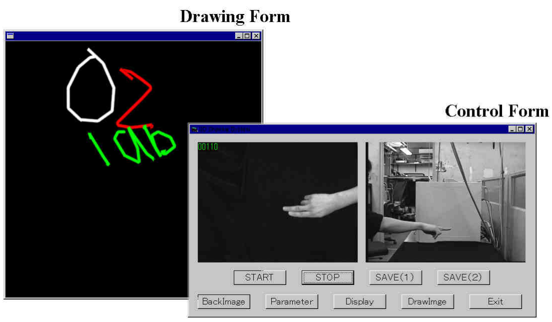
2.3.4 Direct processing of incomplete hand motions with non-contact technologies

Several competitive approaches have been proposed, but many of them suffer from the algorithmic complexity of image processing and the high computational time (Abe et al., 2000). Certain systems restrict the motion of the hand to avoid difficulties and to reduce the problems of occlusion. Many systems are sensitive to variable lighting conditions and backgrounds, and moving in front of changing backgrounds. The pre-calibration of the camera(s) also takes time and presume some knowledge about the position of the hand. Since with one camera the calculation of three-dimensional positions is not possible, researchers apply multiple cameras, typically two, (Segen and Kumar, 1998), or use other additional information, such as shadows (Segen and Kumar, 1999).

We start our investigation with a system, in which only two-dimensional data are considered and a single camera is used. This system, developed by Iannizzotto et al. (2001), substituted the mouse with a vision-based input device (Figure 2.7 b). As the user communicate with the computer through the monitor, two-dimensional information is sufficient. The thumb and the index finger are used to emulate the functions of an ordinary mouse. The necessary information is obtained from the motions and contacts of the fingertips. After filtering the image, the portion where the fingers are detected becomes small (app. 60x100 pixel). Consequently, the pattern-matching algorithm can work in real-time. They applied a 6 frame per second (fps) tracking speed, considering that their camera was able to capture 25 fps and they approximated the speed of dynamic gestures as 8 Hz.

User tests showed that the use of this system could be learnt easily, although the accurate selection of a menu item proved to be difficult. In a similar approach, in the Finger Mouse System, which was developed by Ko and Yang (1997), the position of a fingertip was tracked and the different functions were detected by three pre-defined gestures. Their Finger Draw application is driven by gestures, but these gestures are detected as sequences of points described by the moving fingertip. Based of these point sequences, primitive objects can be drawn and simple operational task can be realized.

In order to make three-dimensional freeform curve design without hand-attached devices possible, Abe et al. (2000) developed a 3D drawing system (Figure 2.7 a). Actually what this system does is using hand motions to complete simple drawing tasks. They used two cameras to detect the moving hand, from which one took top-view and the other took side-view. The restrictions of this system are that the designer must use his right hand, and the back of the designer's hand must always face up. The designer indicates three-dimensional points by positioning the tip of its finger and gives a drawing command by hand pose. The system extracts a pair of fingertips on both images and estimates the three-dimensional position of points by comparing the coordinates and a pair of camera parameters. The average time of hand pose recognition is 10 fps. They reported that they could draw images without feeling delay because of the short processing time.



a.

b.

Figure 2.7 (a) A 3-D drawing system using two cameras for hand motion recognition (courtesy of Abe et al., 2000), (b) A user interacting with the computer using hand gestures (courtesy of Iannizzotto et al. 2001)

Segen and Kumar (1998) first applied two cameras to estimate the three-dimensional position of the fingertip of the thumb and index finger. Then, they implemented a pilot system based on one camera and a light source, most probably to reduce the processing time (Segen and Kumar, 1999). In the latter case, features derived from the projections of the hand and its shadow was used to compute three-dimensional position and orientation. Both systems operate at the rate of 60 Hz. They tested it with a Fly-through application, where the position of the pointing finger along an axis was used to control the velocity. With the information of two fingertips, picking and moving tasks can be realized. In their Scene Composer application, objects can be picked and moved in a three-dimensional scene.

As it has been shown in this section, many efforts have been made to use hand motion as an input means. Striving after natural ways of describing shapes, researchers studied the opportunities of obtaining sufficient amount of information directly from the hand, without any hand-held device. Our observation is that in the investigated cases only some dedicated points were detected. That is, these technologies extract less information than that is needed to generate surfaces or complete boundary of objects. They transfer the obtained data either directly to the modeling subsystem, or to the image processing subsystem and then to the modeling subsystem. They do not apply active hand model to generate extra information. In the next section we investigate those approaches, which try to collect as much information from the hands as possible.

2.3.5 Direct processing of complete hand motions with contact technologies

Murakami and Nakajima (2000) developed a hand-held input device for intuitive three-dimensional shape deformation. It has a cubic shape and pieces of conductive foam are embedded on the surface and the inside of it to form a sensor network to cover the cube. It contains 24 sensors on the cube edges, 48 on the faces and 18 inside the cube to measure deformation. A three-dimensional virtual control volume is defined on the screen for deforming the objective shape. By measuring the deformation of the tool electrically in real-time, the system computes the corresponding deformation of the control volume, and then that deformation is mapped to the objective shape.

The conclusion we can derive based on the investigation in this section is that the currently available contact technologies can be used to obtain large amount of data that is needed to generate complex hand shapes and/or surfaces. However, if an incomplete detection is complemented with the use of active hand models, they can also be considered. Below we investigate the technologies, which have been sorted into this category.

2.3.6 Indirect processing of incomplete hand motions with contact technologies

Data gloves offer the most possibility to capture information about human hand motions, as they were designed especially for this task. The most equipped model contains 22 sensors. It has three flexion sensors per finger, four abduction sensors, a palm-arch sensor, and sensors to measure flexion and abduction. The sensor resolution is 0.5 degrees and the raw sensor data rate is 150 records/s and 112 records/s when it is filtered. With this information hand motion can be captured and a hand model can be driven in real-time. However, position and orientation data can be obtained only with additional tracking devices.

The amount of degrees of freedom tracked can vary. Pure point tracking results in positional (X, Y and Z) information. A number of systems can track position and orientation. (X, Y, Z, A, B and C). In order for the system to be able to follow a point it must be recognizable. To achieve it there are three methods. First, it can be recognized from a geometrical feature. In this case the surroundings of the point must be scanned and interpreted. Second, a passive marker can be attached. This is a piece of material that can be recognized by the system because, for example, it is magnetic. The third option is to attach an active marker. This is a small device that sends out some signal that can be received by the tracking system, for example light pulses. The number of tracked points varies from 1 to 256 in the case of commercially available systems. Different types of media, like light or magnetic fields, can be used for measuring.

The complexity of the tracking task increases if more points have to be followed simultaneously. There must be for example a way to distinguish the different points so that they cannot be mixed up. Depending on the speed of the motions to be tracked one will need a faster or slower system. The speed of the tracking system is determined by the amount of measurements that can be made per second, which is about 100-240 Hz. The accuracy of detection varies from 0.1 mm to 2 mm. Most measuring systems can be influenced by external factors, like light sources or magnetic fields. It depends on the application whether this is a problem or not. A tracking system can influence its surrounding. This can cause unsafe situations. For example high-energy lasers can damage the eye; magnetic field may disturb for example pacemakers. By building up this setup, positional, orientation and also hand shape data can be obtained. Now we analyze systems that use data gloves to obtain hand information for communicating shapes.

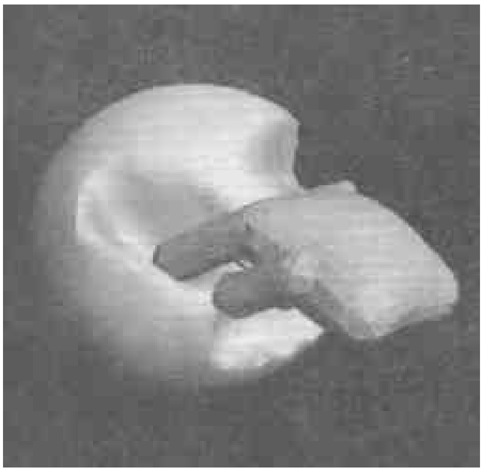
A typical technology was developed by Xu et al. (2000), which is able to support the creation and manipulation of freeform curves by hand motion. They detect the tip of the user's index finger with the data glove and reconstruct its motion curve from these points. They developed algorithms for filtering the points, for instance, to delete unintentional point and to compress redundant points, in order to create a high fidelity curve from the hand motion.

As an early effort, Weimer and Ganapathy (1989) developed a three-dimensional modeling system, which uses hand gestures to interact with a virtual environment. A data glove detects the gestures and it is used to control a computer model of the human hand. The hand model can be parameterized as either a left or a right hand, and it provides continuous visual feedback, showing the hand's relationship to the virtual world. The selection, viewpoint modification and surface modeling functions are implemented by combining hand motion with voice input. Although the usage of the hand model offers the possibility to define swept surfaces by hand motion data, in this system surfaces are built from swept curves.

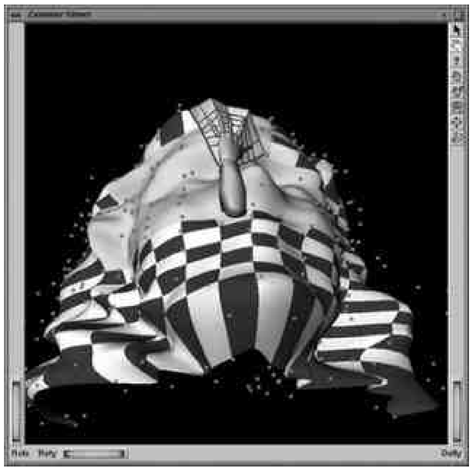
To take more advantage from the hand model, Matsumiya et al. (2000) presented an interactive technique for modeling three-dimensional freeform surfaces in real-time. In this system users can design three-dimensional objects by using their fingers, which are detected by a data glove (Figure 2.8 a). A primitive object is given, which can be manipulated by two operations, denting and pinching. Predefined finger angles switch the types of manipulations. The forefinger, the shape of which is given by a mathematical formula, produces dent deformation. Actually, its shape is subtracted from the primitive object. The pinch deformation adds a cylindrical shape to the primitive object, which is generated among the forefinger, middle finger and thumb.

In the above technology, the shape of only one or two fingers was used. However, it is natural to use the whole hand to deform shapes. The next effort provides a solution for this problem. As a development of the surface deformation technique (Ma et al., 1997), the authors proposed a new method (Wong et al., 1998). They addressed problems with the planar projection method. Before manipulation, users were required to set their hand in a coplanar state to create a base plane of the hand. Deviations of finger joints from the base plane lead to errors in the initial mapping. On the other hand, the amount of hand movement could be small, because the mapping was done via the base plane. Because of these reasons, they changed the mapping method from planar projection to ray projection. To achieve this, they constructed a three-dimensional triangulated hand model; therefore the users do not have to set their hand in a coplanar state.

The hand model was built upon the points measured by the glove and some additional points, which were defined by simple distance assumptions. Then each point was connected to the neighboring points and a surface was created from triangles and the projection was realized through them. The whole deformation procedure took less than 0.5s. The system was further developed to a distributed virtual sculpting environment (Li, 2003) to support shape communication among geographically separated designers (Figure 2.8 b). The main disadvantage of this method is that users got tired during using the glove. They also commented that as they are editing a model with both hands, they can not look around in the virtual environment.



a.



b.

Figure 2.8 (a) Dent deformation of an object (courtesy of Matsumiya et al., 2000), (b) Virtual sculpturing of a human head model (courtesy of Li, 2003)

Nishino et al. (1998) proposed a method for modeling three-dimensional objects based on hand motion detection by a data glove. Spatial and pictographic bimanual gestures were used to create and modify three-dimensional objects. An object was defined by a process, in which the user created and combined primitives, and deformed this rough shape to achieve the wanted shape. These steps were iterated until the final shape is obtained. In order to get an adaptable gesture interface, they implemented a gesture learning and recognition algorithm, which allows the users to register their preferred gestures before using the system. The main findings of their experiments were that dynamic adjustment of the quality of visualization and speed of drawing are critical issues for an efficient modeling. The users of this system spent about 20 minutes to produce complex objects, e.g. a teapot.

While in the previous approach users created and deformed primitive objects, Surface Drawing was developed for creating freeform three-dimensional shapes by hand motion (Schkolne, 2001) (Figure 2.9). As the hand is moving in the three-dimensional space, the path of the hand is directly realized as geometry. Their goal was not to create numerically precise models, but to support shape expression and communication. Although surfaces are defined by hand motions detected by a data glove, modifications of the shape should be solved using tangible tools. To give a solution for this problem, Ma et al. (1997) developed a technique, which allows the users to deform surfaces by hand motion. The angles of finger joints and the local coordinate of the palm were measured by a data glove. To calculate the joint positions, they assumed that the distances between the finger joints are given. These points were further used to interpolate a surface, called hand surface. In their method surface deformation was realized as a planar mapping of the hand surface to the surface of the object.

a. b.

Figure 2.9 The SurfaceDrawing system (courtesy of Schkolne, 2001)
(a) Hand motions creating a shape, (b) The "magnet" tool between the fingers is used to deform geometry

The usage of the hand model also proved to be efficient in the next effort, which is a different application. However, in this case the glove is not equipped with electric wires and sensors, it is in contact with the hand. In the Hand Motion Understanding system, developed by Holden and Owens (2001), a 21 DoF kinematical three-dimensional hand model is used to detect hand motion. The model is a combination of 5-finger mechanism (total 15 DoF) each attached to the wrist base of 6 DoF. They used a color-coded glove in order to locate the joint positions on the images captured by a single camera. The joints are separated by different colors; thus, a color segmentation algorithm can be used to determine markers. Then a correspondence is established between the marker locations and the joints in the model. Computing the size of the markers on the image and predicting its location can solve the occlusion problems. The five previous frames are used to predict the three-dimensional model state and the possible states are projected onto the image. The model is gradually updated until its projection fits the image feature. In order to classify the hand signs, a vector is built from the angles of finger joints, that is, according to the 15 Dofs. A sign is represented by its starting/ending posture and motion information. The starting and ending postures are defined by using basic hand postures of the Australian Sign Language. The number of directional changes in the movement of fingers is used to represents the motions.

From the point of view of our intended application, the technologies that extract information from the bare hand, and use neither active nor passive markers, are even more interesting.

2.3.7 Indirect processing of incomplete hand motions with non-contact technologies

Basically, recognition of hand postures and gestures can be categorized into two approaches considering camera-based systems: model-based approaches and appearance-based approaches. The latter takes advantage of some features extracted from the image of the hand and classifies it into gestures (Utsumi and Ohya, 1999), (Wu and Huang, 2000), (Lamar et al., 2000), (Schlattmann et al., 1997). These methods are usually based on the assumptions that the appearances are much different among different gestures, but small among different people, so the estimation of the hand configuration does not need to be accurate. It can be true in some applications, such as recognition of a set of predefined gestures, but it is not sufficient in applications, which require multi-DoF input, like in our case. Because of the previous reason, we do not give detailed description of the appearance-based techniques and we focus on the model-based approaches. As our aim is to obtain three-dimensional positional data from the hand, we mainly consider approaches with three-dimensional hand models. We do not analyze approaches with two-dimensional hand models (Heap and Samaria, 1995) (Rehg and Kanade, 1994).

First we demonstrate some examples which use a single video camera to capture the hand. Although these methods cannot provide us with three-dimensional positional information, we analyze them because of the hand models. Wu and Huang (1999) proposed a camera-based non-contact method for capturing articulated human hand motion, which was decoupled to global hand motion and local finger motion (Figure 2.10). They employed a kinematical hand model, where each finger was modeled as a kinematical chain with the palm as its reference frame. They treated the hand model as a set of 16 rigid objects, in this case sticks. The configuration of the hand was defined by the length of the sticks and the kinematical relation among them. A generic three-dimensional hand model was automatically calibrated to each person in order to derive user-specific models. Feature points were observed from the images taken by the camera. Based on these points and the estimated motion parameters the three-dimensional hand model could be regenerated. The algorithm worked accurately even if the local finger motion between two consecutive frames is large. However, the algorithm failed, when one of the fingertips was occluded.

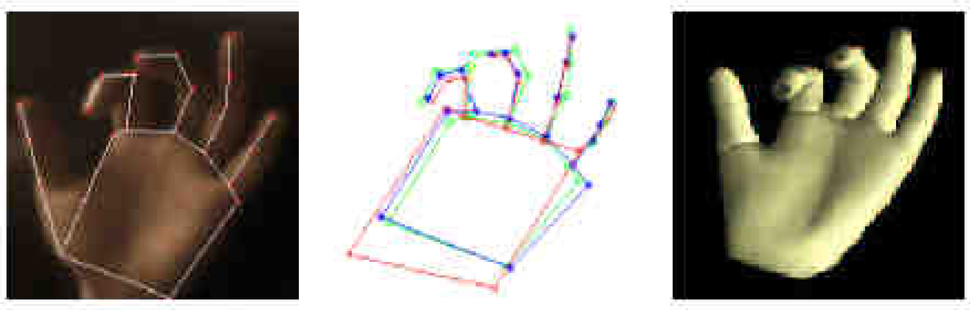


Figure 2.10 Modeling articulation of the human hand (courtesy of Wu and Huang, 1999)

Heap and Hogg (1996) applied a three-dimensional deformable Point Distribution model of the human hand in order to track the hand moving with 6 DoF. The hand model was constructed from real life examples of hands and it is modeled as a surface mesh from which large amount of position information can be derived. The model is used to track a hand in real-time, 10 fps on a standard 134 MHz Silicon Graphics Indy workstation. The model is projected onto the input images and three-dimensional edge detection is used to move and deform the model. Their system assumes a homogeneous dark background and the user must initialize the hand model. They addressed problems like scale and rotation confusion, planar rotation ambiguities, occlusions and implausible model shapes.

Although positional information is not available in these cases, the hand shape is successfully reconstructed. In the following part of this section some examples are shown using multiple cameras.

Stenger et al. (2001) used an anatomically accurate three-dimensional hand model for tracking the hand on two-dimensional images (Figure 2.11). The hand model was created from truncated quadrics, which provides a practical method for generating contours of the model. The hand model was projected and compared with the edges of the hand detected on the images. Their system is scalable from single to multiple cameras and from rigid to articulated model. In their experiments they used two cameras and a 7 DoF hand model. This method can handle self-occlusion.

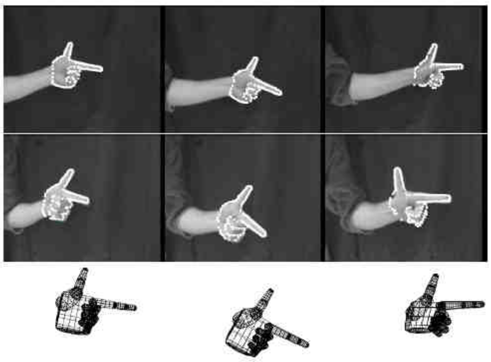


Figure 2.11 Model-based 3D hand tracking (courtesy of Stenger et al., 2001)

They captured images with 360x288 pixel in front of a dark background. The parameters of the hand model were manually set to match the pose of the hand in the first frame. The global hand motion was tracked with 6 DoF and 1 DoF was given to the configuration of the thumb. They intended to use this model in simulating a point and click interface, using positional, velocity and acceleration data. They addressed that the computational complexity grows linearly with the number of cameras. They reported that the system can not operate in real-time, and they proposed some methods to improve the speed, such as optimizing the code, using a faster machine (in this experiment a Celeron 433MHz machine was used) or a distributed system.

Ogarawa et al. (2003) captured the hand by three infrared cameras. With this technology, silhouette images of the hand can be extracted easily. Then a volumetric representation of the hand is generated from the three silhouette images. In this approach, a kinematical bone hand model is used with 20 DoF of 15 joints, and extra 6 DoF for the entire hand, 3 for translation and 3 for rotation.

A surface mesh model of the hand is constructed by measuring a real hand and fitted to the bone model. The position and shape of the reconstructed hand volume is estimated by fitting the articulated hand model in the three-dimensional space. They reported that the accuracy of thumb pose recognition is less than other fingers due to the complex kinematical structure.

In these cases they used multiple cameras because of the accurate reconstruction of the hand configuration. With this technology positional data also can be obtained as we could see in Section 2.5.2.

2.3.8 Indirect processing of complete hand motions with contact technologies

CT, MRI and ultrasound technologies can capture internal images of three-dimensional volumes and scanners can detect points on the surface of objects. With this information the surface of the three-dimensional object can be reconstructed. However, if we want to measure the three-dimensional position and orientation of the object, in our case the hand, additional equipment is needed. Typically, three-dimensional sensors are attached to the object to be measured, that is the reason we categorized them into the contact technologies.

In medical examinations 3D ultrasound, CT and MRI are a widespread technique. Recently researchers deal with surface reconstruction based on images provided by these technologies (Estépar et al., 2003), (Cheng and Dey, 1999).

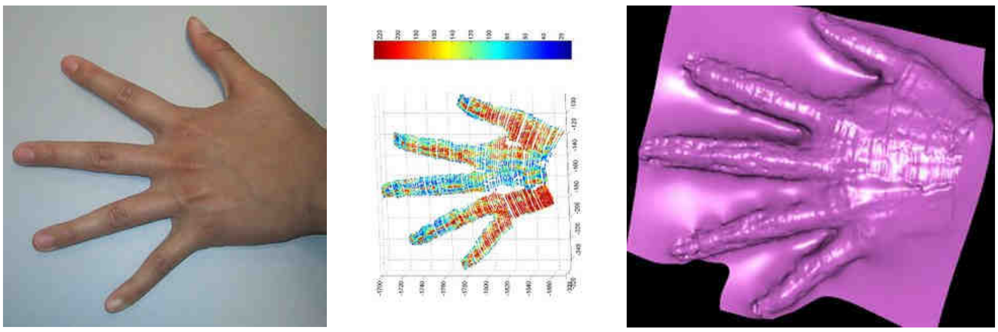


Figure 2.12 Direct surface extraction from 3D ultrasound images (courtesy of Zhang et al., 2002)

Zhang et al. (2002) proposed a technique for extracting surfaces from 3D ultrasound data (Figure 2.12). They used an open architecture ultrasound machine for measurement and markers were attached to the probe to be able to track its position and motion optically. The accuracy of tracking is high enough (~0.1 mm), but the overall system accuracy (~1mm) depends on the calibration of the probe. During the calibration process the transformation matrix from 2D ultrasound images to the three-dimensional probe is determined. The measured surface is implicitly described as a set of points with three coordinates. Then the surface is defined by a single function, whose parameters are these points. As the number of the obtained points was more than 100000, and it just highlighted the noise, point deletion algorithm was needed. Depending on the application, the amount of remained point can be chosen. With almost 10000 points they achieved 50s fitting time and 73s surfacing time, by using a 1GHz Intel P3 computer with 1GB RAM.

Scanners provide us with large amount of point data. Like in case of the previous techniques, the whole shape of the hand is obtained, but it is not known which points belong to the different parts of the hands. Most of the research is on surface reconstruction upon scanned data (Várady et al., 1997), (Vergeest et al., 2001). However, in our case this large amount of point information is sufficient, but the problem is making correspondence between the detected points and the parts of the hands.

2.3.9 Indirect processing of complete hand motions with non-contact technologies

Deawele et al. (2004) proposed a method for tracking hand motion from 3D point trajectories and a smooth surface model. They developed an articulated hand model, where the skin surface is defined as an implicit surface, and the skin motion is described by skinning techniques used in computer animation. The basic configuration of the hand is a 27 DoF kinematical chain. A sequence of images was taken with the restriction, that the palm should face the cameras. About 500 points of interest were extracted and matched between the left and right view, and the resulting 3D point set served as input for hand tracking. Then the hand structure was modeled interactively, and the third view was used for the validation of tracking results. In our case, this large amount of point information is sufficient, and we do not need an additional hand model.

2.4 Discussion

In this section we further investigate the various technologies from the aspect of using in a hand motion based shape conceptualization system. To support the discussion, first we characterize the technologies for the information content that they provide for generation of swept virtual surfaces according to the given requirements. Let I_H be the amount of information obtained from the moving hand; I_M the information that is derived from a virtual hand model, and I_S the information needed to reconstruct the swept surface.

Figure 2.13 shows the characterization of the discussed HMP approaches for the kind and amount information they process. All direct incomplete approaches provide less information than needed due to the partial scanning of the hand. Conversely, all indirect complete approaches result in more, actually redundant, information since they completely scan the hands and also manipulate hand models. These features do not promote the application of these two approaches in hand motion based shape conceptualization. Therefore, we excluded them in our research from the further investigations.

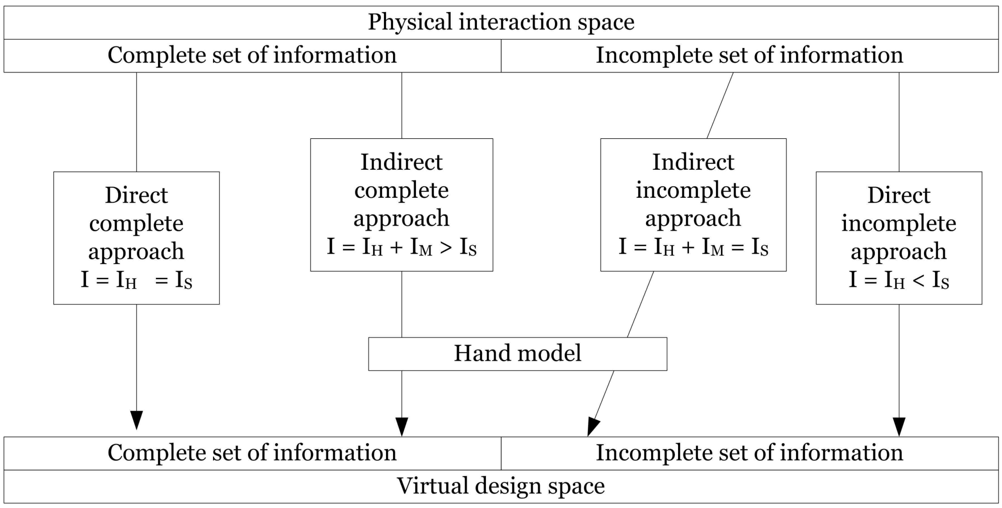


Figure 2.13 Comparison of hand motion processing approaches

The indirect incomplete HMP approach seems to make sense in our specific application since the information obtained by partial scanning of the hand can be extended by the information obtained from a hand model, and thus the swept surfaces can be reconstructed with high fidelity. The direct complete approach is also appropriate, since it is supposed to obtain and transfer sufficient amount of information to the virtual modeling space at once. However, the number of actions involved in the two processes is different. In the first case, (i) the hands should be detected and scanned, (ii) the information obtained by partial scanning should be transferred to the hand model, (iii) the hand model should be actualized, (iv) the required additional information should be derived, (v) the extended set of information should be sent to the shape modeling system, and (vi) the surface should be displayed. In the second approach, (i) the hands should be detected and fully scanned, (ii) the information should be transferred to the shape modeling system, (iii) the information should be preprocessed, and (iv) the surface should be displayed. A process involving less number of processing can be in principle better, but in our case we have to take into consideration the capabilities of the current technologies. For this reason it is not obvious which approach is finally better. In any case, we give preference to non-contact technologies for the comfort of the designers.

The possibility of real-time processing strongly depends on the time elapsed by the execution of the actions of processing. Therefore the speed of detection, scanning, and computation is also considered as a technology selection criterion. Real-time processing is crucial in a conceptual design system, where ideas may come rapidly after each other and designers need fast visual feedback. Based on the cognitive model of shape conceptualization, the typical cycle time is between 1 and 10 s. It means that the hardware and software platform of the system should be able to provide us with visual feedback in at least ten seconds. The speed of the hand motion (5-8 m/s) is also a challenge for the HMP technology. Our analysis showed that while contact technologies like data gloves could work in almost real-time, camera-based detection systems need more time due to image processing. On the other hand, direct complete HMP approaches elapse more time at scanning the hand than the indirect incomplete approaches, but the latter require additional time to process the virtual hand model.

When designers use their hands to conceptualize shapes in the three-dimensional space, the free movement of the hands is a basic requirement. The designers should not be limited by the applied detection and scanning technologies. Intuitiveness of motion suffers a lot under restrictions such as 'user's hands must always face up'. Furthermore, if the hand movement is constrained by heavy and uncomfortable equipments, or by cables that connect the user to the computer, the space of motion is limited and the comfort is demolished. It requires significant adaptability from the HMP technology that can currently be achieved only with limitations. Specific technologies such as color-based gloves only slightly restrict the hand motion, though they are in a direct contact with the hand. Obviously, the non-contact technologies meet the comfort and adaptability requirements much more. However, data gloves can work properly in different hand positions and orientations, camera-based systems usually restrict the position and orientation of the hand due to the difficulty of handling occlusion problems. In the case of gloves, for instance, also the different hand sizes can cause problems. It is requested that the quality of HMP should not be influenced by the trajectory and speed of hand motion. Some of the low-scale image-based systems have intrinsic limitations to fulfill this requirement.

The constructed surface should properly reflect the details and characteristics of the intended surface - a fact that introduces the requirement of fidelity. The typical magnitude of the macro-geometry of the human hand is 10 – 100 mm, and of the micro-geometry is 0.1 – 1 mm. However, the hand moving in the space sweeps a vague domain rather than a crisp surface. The reasons of this are (i) the multiplicity of points on the hand that generate the surface, (ii) the uncertainty of the best fitting motion trajectories, (iii) shaking and imperfect forming of the hands while moving, and (iv) the interaction of macro- and micro-geometry information at scanning the hand. In general, the typical magnitude of the characteristic uncertain movements is 1 – 10 mm. The HMP technology and the geometric modeler should jointly take care of these. Some progress has been achieved with fuzzy sensing technologies, but much more can be expected from non-nominal shape modeling, such as vague discrete interval modeling (VDIM) (Rusák, 2003) and alpha-shape modeling (ASM) (Gerritsen, 2001). Unfortunately, in many of the current research projects, hand motion based shape input and shape concept modeling have not yet been addressed concurrently.

2.5 Conclusions

Hand motion is regarded as a prospective input mechanism for computer-aided conceptual design systems for initial shape design of consumer durables. Its success or failure in this application largely depends on the enabling detection and processing technologies. With the findings of the literature study, we answered our research question regarding the real-time information extraction from human hand motions (Section 1.6). We learnt that a lot of research and development effort has already been invested in this area. We introduced a classification scheme to systematize the survey and our investigations. This chapter reported on the findings about the state of the art and analyzed the technologies applicable to hand motion processing.

The duality of the human and technological aspects of hand motion detection has been already discussed in Section 1.3. Regarding the technologies found, we want to mention that on the one hand, we always give privilege to the aspects of ergonomics, but on the other hand the selected solution must be feasible from the technological aspect. Unfortunately (or naturally), the inconvenient devices (gloves, markers) offer easier detection, and the convenient devices (camera detection without aids) result in a much more difficult detection, not only because of the complexity of image processing algorithms, but also in terms of computational speed. As the goal of this research was the development and testing of a proof-of-concept system in conceptual design situations (Section 1.4), we focused on the theories and implementation of a hand motion processing system with a minimal effort on detection.

We learnt that direct complete and indirect incomplete HMP technologies have the potential to support hand motion-based shape conceptualization. In this chapter we compared these two categories of technologies from four aspects: contact, speed, adaptability and fidelity. The advantage of the direct complete technologies is that they do not need extra time to process a virtual hand model, but they do need it to scan the moving hands. If sufficient amount of information is obtained from the hand, fidelity of the generated surface can be high. Indirect incomplete HMP technologies target landmarks or other characteristic points only. They require less scanning time as well as less sensitive and powerful technology, which is an important issue from a technological point of view. After this study, our conclusion was that based on the related literature the HMP technologies can only qualitatively be assessed for applicability in hand motion based shape conceptualization. Consequently, more research was needed regarding the technological in order to decide on a concrete technology, which meets the requirements the best. The results of a market search with a focus on the operational parameters of tracking systems are reported in Section 4.5.

Based on studying the related work, we concluded that the characteristics of the input data, such as the amount of the detected data or the flexibility of the input means, strongly influence the method of surface generation. We learnt that the hand motion interpretation requires two separate processes, namely, detection and recognition. Because the proposed hand motion-based input strongly relies on visual feedback, we also studied the different visualization techniques. We concluded that those output devices are ideal, which do not place the user into an immersive virtual space, and in case of collaborative work, those which give visual feedback to the users according to their viewpoints.

For an effective shape conceptualization, a fast motion detection process is crucial. The main finding of the study reported in this chapter is that fast detection can be achieved by detecting small amounts of points in space and using a kinematical model of the hand to extend this initial set of information to the required set of surface modeling information. It has to be noted that hand motion based shape generation actually requires a complex upper limb model, rather than just a hand model. The reason is that the designer should have the freedom to perform hand motions and to identify spatial positions in any arbitrary region of the virtual modeling space.

With a view to the findings above I discuss the theories (Chapter 3), the applied methods (Chapter 4) and the implementation details (Chapter 5) of the proposed hand motion processing system. This chapter showed that two criteria have to be definitely met, namely, (i) fast detection of hand motions and (ii) real-time recognition of the signs of a moderately large set of hand motions (HML). In the next chapter I consider these criteria and present methods for hand motion detection and recognition.

THEORY OF HAND MOTION-BASED SHAPE CONCEPTUALIZATION

3

The previous chapter showed that a lot of research has been already done in the related fields. I learnt that there is no out-of-the-box solution for the problems discussed in Chapter 1, but the study of the related literature showed some starting points. It gave directions for establishing theories for the real-time detection and recognition of hand motions, which are discussed in this chapter.

I start with the introduction of the Hand Motion Language, which is the set of signs I applied for shape generation and manipulation. This part is an excerpt of the paper (Horváth et al., 2003b), in order to understand the proposed method of shape modeling and to discover fundamentals, more specifically, the enabling features for the recognition of the signs. The definitions and theories for hand motion interpretation were published in (Varga et al., 2006b). The basics of the kinematical hand model were published in (Varga et al., 2005a). This chapter is mostly the reworked and extended version of these two publications. In addition, the theories of surface generation and manipulation are presented as well.

3.1 Theoretical fundamentals of the Hand Motion Language

(Horváth et al., 2003b) defined a modeling language, the Hand Motion Language (HML). This is the set of hand motions that I used in my research. One of the major objectives in defining the HML was to minimize its dependence on temporal, morphological and spatial variances. Likewise in conventional computer-aided drawing and geometric modeling, macro-level operations offer speed advantage over elementary operations. The situation is the same with hand motions. Combinations of simple words make the shape modeling process not only more effective, but also more expressive.

The HML was defined as a formal modeling language comprising three types of constituents that were formally specified: (i) the set of symbols, (ii) the rules of concatenating the symbols into words and (iii) the grammar for composing sentences from the words. As a formal language, HML is based on the postures and movements of the human hands. It was developed using the analogy of a verbal-textual language, which consists of an alphabet and a grammar that interweave in the language.

Called terminal symbols in the theory of formal languages, the letters of the HML are signs produced as postures of the hand. A purposeful combination of sequences of changing postures of the hand or the hands is a word of the HML. Thus, words represent the lowest semantic level of constructive actions of shape conceptualization. A sentence is composed of words that are needed to express the intended semantics. A sentence of the HML is a sequence of words (i.e., hand motions) that is needed to generate the components of a simple shape or to manipulate it. A paragraph is a sequence of sentences that are needed to describe a simple shape. Finally, a chapter is a set of sentences constructing one-piece compound shapes or multi-piece shape assemblies.

Let the letters, words, sentences, paragraphs and chapters of the HML be denoted by l, w, s, p and c in general, respectively, and the sets of them by the same capitalized letters. Let the grammatical rules for words be denoted by q and for sentences by r. Let Σ denote the predefined set of the Σ_i unique hand motions, where i is the number of the different predefined hand motions. Thus, s is a sequence of hand motions such that $s_k = \left\{\Sigma_i \mid \Sigma_i \in \left\{\Sigma_0, \Sigma_1, \ldots, \Sigma_n\right\}\right\}$ where Σ_i is a unique hand motion, and $n \neq 0$. A word of the HML is a construct as $w = \left(l_k, q_j\right)$, where $k = 0, \ldots, m, m \neq \infty$ and $j = 0, \ldots, n, n \neq \infty$.

According to its meaning, a hand motion can be either procedural or constructive. A word consisting of (letters of) procedural hand motions is a *procedural word*, and those of constructive hand motions are *constructive words*. A procedural word provides information for the process of shape conceptualization, while the constructive words provide information for the shape model. Hence, the total set of words consists of the subsets of procedural and constructive words: $W = \left\{w_j \mid w_j \in \left(W^P, W^C\right)\right\}$.

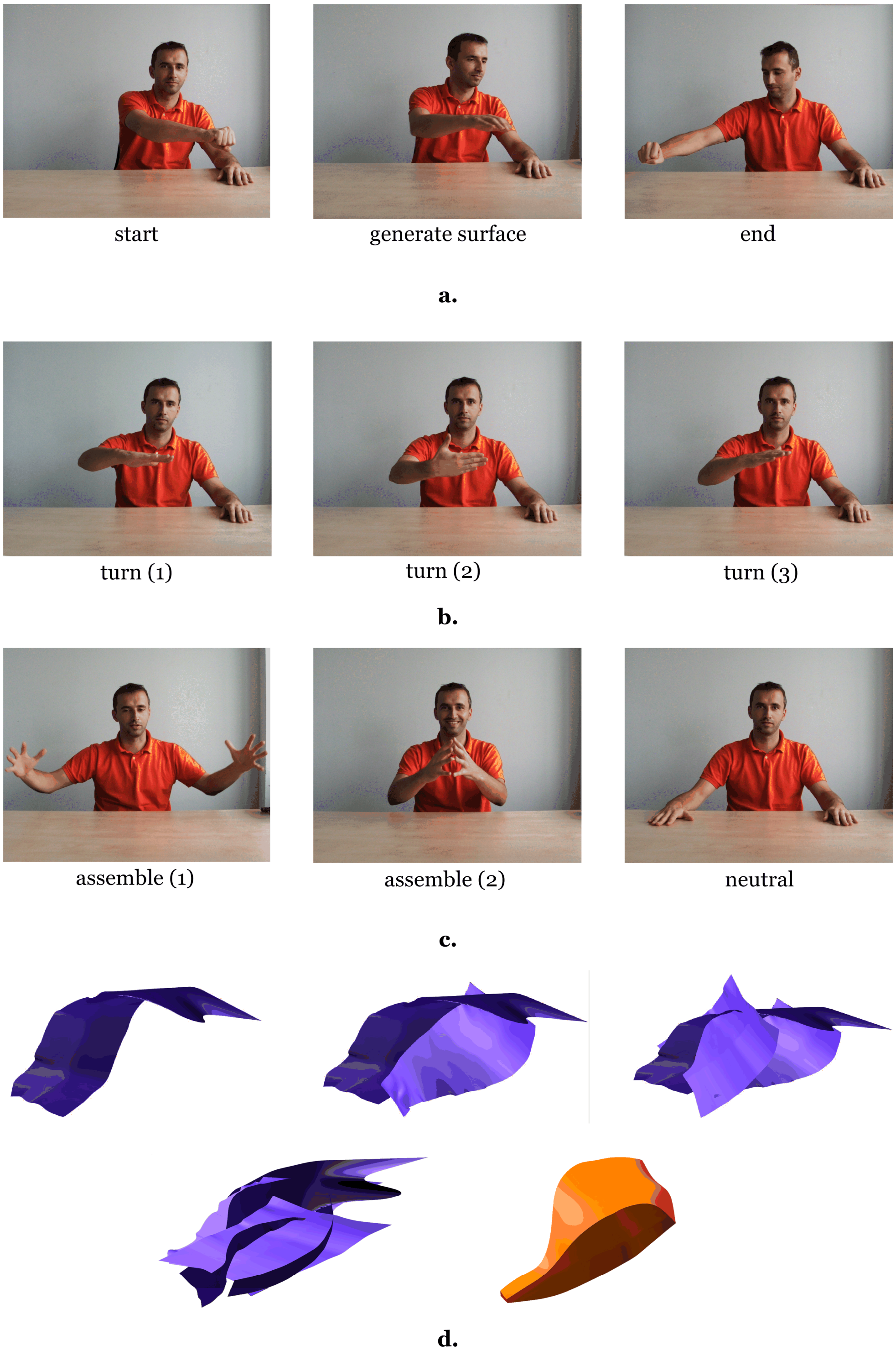


Figure 3.1 An example of using HML

Typical procedural words are the words *start*, *stop*, *share* and *obtain*. The main reason of defining procedural words is that computer controlled detection, scanning and conversion of hand motions require well recognizable starting and end postures that enclose the constructive hand motions. The constructive words can be grouped according to the semantics of the information they convey to the shape model. Based on the information contents they can be geometric, identification, connectivity, positioning, scaling and assembling words (see Appendix B for the complete vocabulary of the words of HML).

A sentence, s, of the HML is a composition of words obeying to the rules of the language: $s = \left(w_i^P, w_j^C, r\right)$, where: $i = 0,\ldots,m,\ m \neq \infty$ and $j = 0,\ldots,n,\ n \neq \infty$. Less formally, a sentence is a shape formation oriented arrangement of procedural and constructive words. A paragraph is a composition of sentences, i.e., $p = \left(\bigcup s_i\right)$, where $i = 0,\ldots,m,\ m \neq \infty$. Finally, a chapter is a composition so as: $c = \bigcup p_i$, where $i = 0,\ldots,m,\ m \neq \infty$.

An example for the usage of HML can be seen in Figure 3.1. In this example, an initial shape of a mouse was created by generating four surfaces, turning them respectively and finally assembling them to get the inner volume of the four surfaces. Figure 3.1 a shows a surface generation process, which constitutes from a *start* HML word in the beginning, a sweeping movement of the hand which expresses the shape of the surface to be generated and a *stop* HML word, which indicates that the surface generation process is ended. Figure 3.1 b shows a turning procedure of a surface, performed by the *turn* HML word. The *assemble* HML word is seen in Figure 3.1 c. Figure 3.1 d shows the results, i.e. surface patches in each step and Figure 3.1 e depicts the final shape.

The next section elaborates our proposed method for the computer interpretation of the HML. First the complete process of hand motion interpretation is discussed and then the related terms are defined. Based on the definitions, the theory of segmentation and recognition of the words of HML are introduced, for which algorithms are defined in Chapter 5.

3.2 Complete process of trajectory segmentation and processing

It has to be emphasized, that the HML is dedicated to the shape conceptualization phase of product design, when the speed and intuitiveness of expressing the design concepts, rather that the precision of the model representation play an important role. From a user point of view, a basic requirement is that the design process needs to be *continuous*, and in addition to supporting the generation of virtual surfaces, the shape conceptualization system should provide immediate *visual feedback* for the designers. Continuity means that the designers do not have to stop with their hand motions each time they want to express the next modeling command. Immediate visual feedback is necessary for the designers in order to be able to see and reason about the actual status of the shape model, to rapidly decide on the next design action. On the other hand, delayed feedback may destroy intuitiveness and comfort of using the shape conceptualization system.

From a system development point of view, these requirements mean that the HML words have to be recognized during the *continuous motions* of the hands. More precisely, two processes are needed: (i) segmentation and (ii) recognition. The segmentation process is used to find and detach the meaningless parts of the hand motion. In other words, it has to find the beginning and the end of the segments of the hand motion trajectories that express an HML word. The recognition process identifies which modeling command was performed by the user and associates it with the standard definition of that command.

Figure 3.2 is the basis of discussing the information transfers in the process of HML based shape conceptualization. First, the designers indicate that they are ready for starting the design process. They set the equipment into standby state, which shows that it is ready for detection. After *motion is sensed* in the environment, the hands are located and tracked. A series of positional data is converted into a motion trajectory, which is used to generate surface patches in case of geometric commands. As far as the manipulation commands are concerned, the retrieved positional data is converted to features used in posture recognition, and these *features are continuously compared* with the stored postures. When the best match is found during a motion sequence, it is *identified as the performed HML word*. When this selection cannot be done, the motion is scanned once more. Afterwards a decision is made if the HML word was a geometric or a manipulation one. If geometric, the motion trajectory information is converted to a *surface patch* representing the swept surface. If it is manipulation, the information is converted into a *modeling command*. Then in both cases the information is sent to the modeler in order to visualize the results of the indicated action.

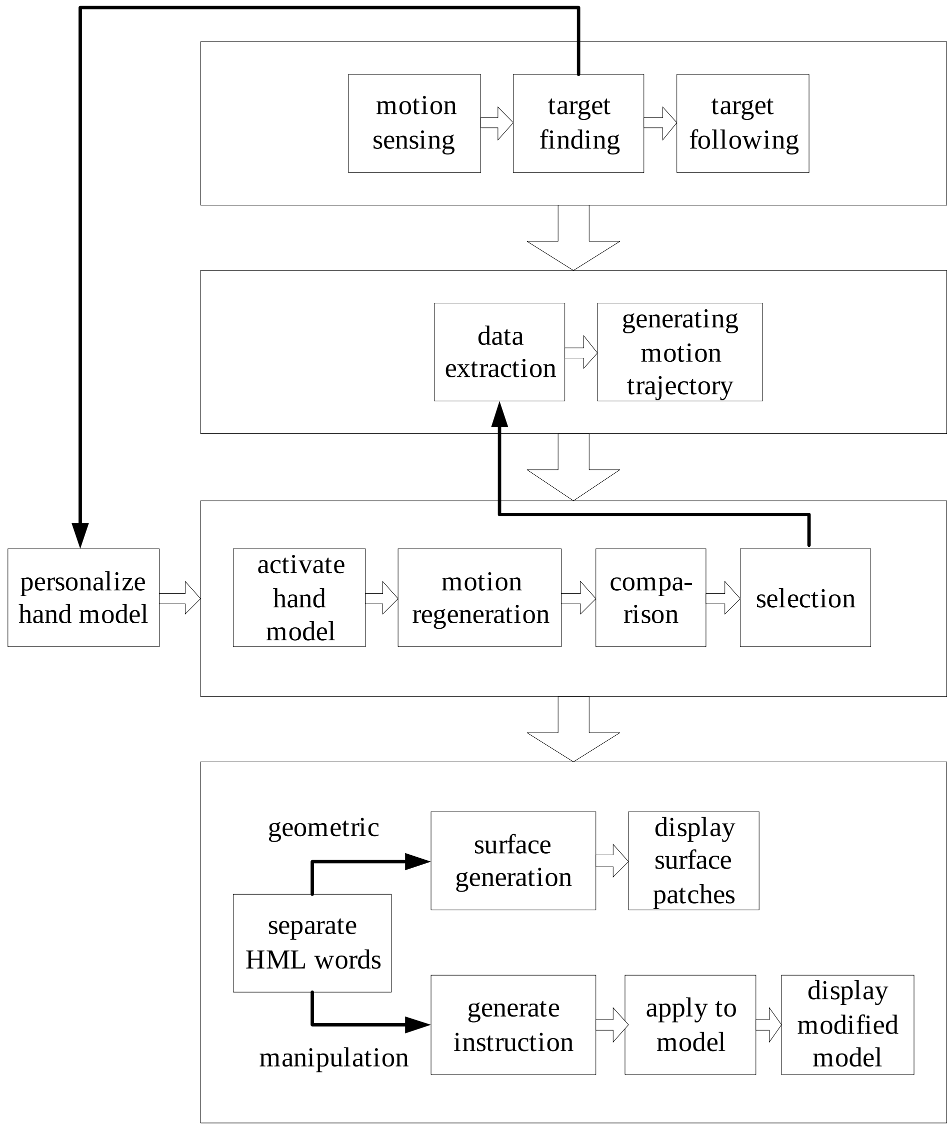


Figure 3.2 Process of hand motion processing

From the side of the HML words, there are some features which facilitate the implementation of the segmentation and recognition processes. A HML word has two states: a *transition state* and a *steady state*. In the transition state the postures of the hands are continuously changing, while in the steady state they are changing according to a pattern, or do not change at all. The steady state is actually the useful part, which allows the recognition of HML words. Therefore, the transition state refers to a command change, and the steady state conveys the meaning of the intended modeling command. Considering these features of the HML, the basic idea behind the segmentation and recognition processes is the continuous tracking and quantification of the postural changes.

3.3 Definitions related to hand motion interpretation

The process of hand motion interpretation is built around some key concepts that are logically and procedurally interrelated. These key concepts are represented in Figure 3.3. A hand motion is processed as a composition of a motion trajectory and a series of postures. A trajectory is composed of the incremental sequence of displacements and a point set describing the spatial positions of the landmarks of the hand in a given posture. A posture is expressed as the composition of this point set and the features of the posture that is represented by a set of quantitative parameters. Recognition of a segment uses the information conveyed by the point set, the displacement and the features. If the useful segment of the hand motion trajectory is successfully extracted, then it can be converted to a HML word, either expressing a geometric command, or a constructive command. The former type of commands enable the generation of surface entities, the latter type of commands are needed to manipulate the shape and control the procedure. To be able to introduce a formal theory for hand motion interpretation, first I formally define the key concepts shown in Figure 3.3.

*Hand motion*s are sequences of two-handed hand and arm postures changing continuously over time, expressing either constructive (for geometry generation) or procedural word (for geometry manipulation) of the HML, and conveying motion trajectory and three-dimensional geometric information in real-time. Therefore, a hand motion HM is defined as a combination of postures (C) on different trajectories (T): $HM = C \times T$.

A *posture* is a specific configuration of selected landmarks of the hand and arm at a given moment in time. Consequently, a posture is described as a combination of landmark positions (P) and relations of landmarks (F). That is, $C = P \times F$.

A *trajectory* is the motion of one landmark point of the hand and its displacement over time as defined below, and can be described formally by: $\tau_i = \bigcup_{t \in \mathbb{N}/\{0\}} \vec{p}_{i,t}$; $\vec{p}_{i,t} = \vec{p}_{i,t-1} + \vec{\delta}_{i,t}$, where $i \in \mathbb{N}$. Trajectories of multiple points are defined as: $T = \bigcup_{i \in \mathbb{N}} \tau_i$.

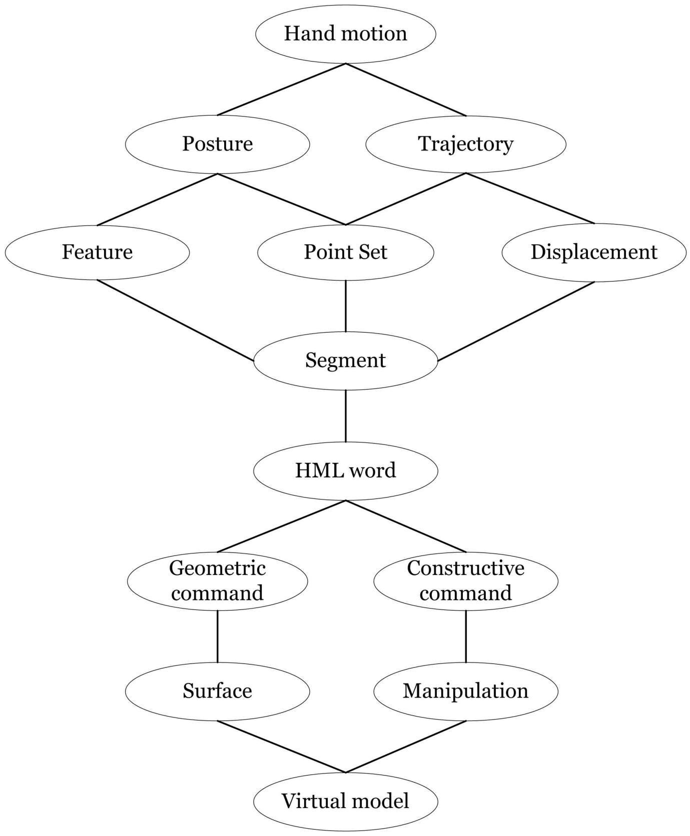


Figure 3.3 Elements of hand motion interpretation

A *feature* is a metric relation of two landmark points at a given moment: $f_{i,j} = \left(\vec{p}_i, \vec{p}_j\right)\big|t_k; i, j, k \in \mathbb{N}; i \neq j$ and a feature set is defined as: $F = \bigcup\limits_{\substack{i,j,k\in\mathbb{N}\\ i\neq j}} f_{i,j}\big|t_k$.

A feature set, together with a P point set, represents a $C_t,\ t \in \mathbb{N}$ posture.

A *point set* (P) is a set of selected landmark points (i) of the hand and arm at a given moment in time. A point set is defined as: $P = \bigcup\limits_{i\in\mathbb{N}} \vec{p}_i$; $\vec{p}_i \in \mathbb{R}^3$; $\vec{p}_i = \left(x_i, y_i, z_i\right)$.

A *displacement* is defined as the translation of one point over a period of time: $\vec{\delta}_{i,t} = \vec{p}_{i,t} - \vec{p}_{i,t-1}; i \in \mathbb{N}; t \in \mathbb{N}/\{0\}$. Displacement of a point set is defined as: $\Delta_P = \bigcup\limits_{\substack{i\in\mathbb{N}\\ t\in\mathbb{N}/\{0\}}} \vec{\delta}_{i,t}$. Therefore, the set of trajectories of the P point set is defined as a combination of specific points and their displacements: $T = P \times \Delta_P$.

A *segment* (E) is a sequence of $C_t, t \in \mathbb{N}$ postures in time, that is, $E = \bigcup_{t \in N} C_t \mid C_t = C_{t-1}$, where $t = [i,k], i,k \in \mathbb{N}$. This means that in a given segment the hands form the same posture, and a segment is period of the whole motion. The segment is a part of the hand motion trajectory: $T = \bigcup_{i \in \mathbb{N}} E_i$.

Let us assume that a segment corresponds to an *HML word* (w). The whole HML vocabulary is the set of individual HML words, and can be denoted as: $W = \bigcup_{i \in \mathbb{N}} w_i$.

A w HML word can be either of *geometric* nature, $W_G = \bigcup_{i \in \mathbb{N}} w_i$, of *constructive* nature: $W_C = \bigcup_{j \in \mathbb{N}} w_j$ or of procedural nature: $W_P = \bigcup_{k \in \mathbb{N}} w_k$.

Geometric words generate s*urface*s, which are represented as point clouds. The points of the point cloud are provided by the motion of a P_S subset of the P points and constructed by a certain function f : $S = f(T_S) = f\left(P_S \times \Delta_{P_S}\right)$.

Constructive words are to express manipulation operations. A *manipulation* operation can be identification, connection, positioning of surfaces or surface elements, scaling surfaces or assembling of surfaces. Let's denote the group of manipulation operations by M, which consists of a finite number of operations, therefore $M = \bigcup_{i \in \mathbb{N}} m_i$. A manipulation operation corresponds to a HML word, and contains the related geometric information.

A *virtual model* is generated as a composition of multiple surfaces on which multiple manipulation operations are performed: $VM = \prod_{i,j \in \mathbb{N}} m_j(S_i)$. The generation of the virtual model is a sequence of surface generation and manipulation processes.

The definitions introduced in this section provide the basis for the discussion of the proposed theories for hand motion interpretation.

3.4 Theory of trajectory segmentation

The basis of both HML word recognition and trajectory segmentation is the previously described two-handed posture recognition method. In each frame of motion, the C posture is defined and compared with the previously recognized postures. Let's denote the posture recognized at a certain moment by C_1. With this in mind, the theory of segmentation can be defined as follows: if $C_1 = C_{1-i}, i = 1, 2, \ldots, n$ and $C_{1-i-1} \neq C_{1-i}$, then it means the beginning of a segment. The i number refers to the previous frames, which are compared to the current frame. The n number, which is the number of frames to be compared, can be set according to the frame rate of the measurement device and the speed of the hand motion.

Accordingly, $t = 0; \forall \left(C_i \mid C_{1-i} \neq C_1, i = 1, 2, \ldots, k \right) \Rightarrow j = t + 1, \mathit{and\ if}\ t > m$ means the end of a segment, and refers to a posture change. k is the number of frames in which the postures are compared, and m is the number of different postures during these k frames. The reason for counting the non-equivalent postures is that there could be errors during the measurement, e.g. because of occlusions. In these cases the posture recognition is not successful; nevertheless, it does not mean the ending of an HML word. Momentary measurement errors can be filtered with this method. When a segment starts, the process of HML word recognition is applied to recognize the command performed by the user.

3.5 Theory of recognition of the words of the Hand Motion Language

The whole set of the words of HML is defined in Appendix B. This section explains the proposed theory of recognizing the words of HML, which is necessary in order to identify which word was indicated by the user. The recognition of HML words is a multi-step process, which comprises the following sub-processes: (i) personalization, (ii) feature definition, (iii) one-handed posture recognition, (iv) two-handed posture recognition and (v) handling of sign groups. These processes are explained below.

After a careful analysis of descriptive features, I selected six of them (Figure 3.4), which describe and distinguish each of the postures which occur in the HML. These features are defined for the left hand as follows: $F_L = \left(f_{L1}, f_{L2}, f_{L3}, f_{L4}, f_{L5}, f_{L6} \right)$; where f_{L1} is defined by the distance $d_{L1} = \left| \overrightarrow{P_3 P_5} \right|$, f_{L2} by $d_{L2} = \left| \overrightarrow{P_2 P_4} \right|$, f_{L3} by $d_{L3} = \left| \overrightarrow{P_5 P_7} \right|$, f_{L4} by $d_{L4} = \left| \overrightarrow{P_1 P_5} \right|$, f_{L5} by $d_{L5} = \left| \overrightarrow{P_1 P_7} \right|$, and f_{L6} by $d_{L6} = \left| \overrightarrow{P_1 P_8} \right|$. The feature set F_R is defined for the right hand likewise. The set of measured points can be seen in Figure 3.4.

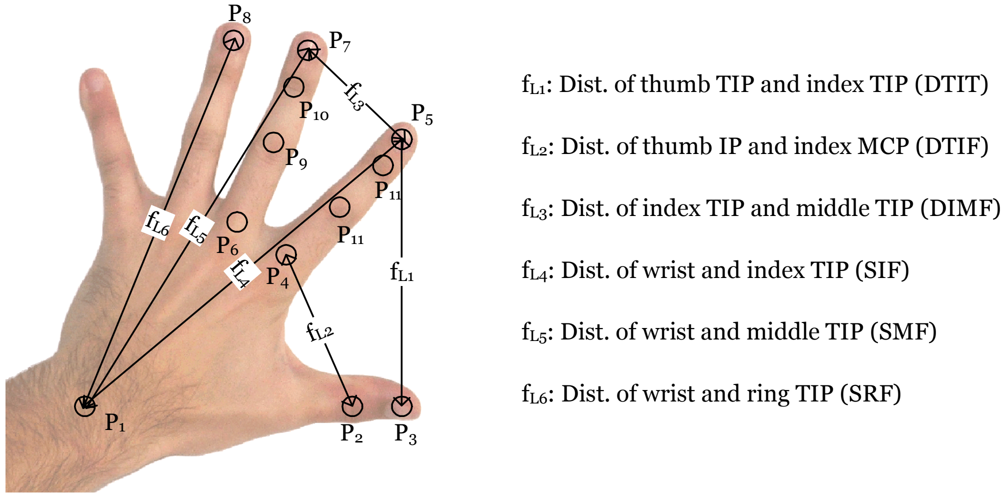


Figure 3.4 Features for posture recognition

The features f_{L1}, f_{R1}, f_{L2}, f_{R2}, f_{L3}, f_{R3} can take up two values, either *close* or *far*; and the f_{L4}, f_{R4}, f_{L5}, f_{R5}, f_{L6}, f_{R6} features either MF/E or TF values, which mean moderately-flexed/extended and totally-flexed. This set allows for modeling 64 posture variances for one hand, out of which 13 are meaningful, 17 meaningless and 34 impossible. The meaningful postures occur in the HML vocabulary, the meaningless do not, and the impossible configurations refer to postures which can not be shown by the hand.

The HML word recognition starts with a *personalization* process, in which relevant distances are obtained from postures shown by the user. These distances are related to the higher - *far* and *MF/E*, accordingly – values of all features. Let us denote these distances as $D_{LP} = (d_{LP1}, d_{LP2}, d_{LP3}, d_{LP4}, d_{LP5}, d_{LP6})$ for the left hand. For the right hand, D_{RP} is defined accordingly.

During the detection process, the distances D_L and D_R, of the feature sets F_L and F_R, are obtained in each frame of motion. In the *feature definition* process these measured distances are compared to the distances in D_{LP} and D_{RP}, in order to define the corresponding feature values. The values which are used to compare the distance quotients come from experimental measurements. The same algorithm is used for the right hand accordingly.

The feature sets F_L and F_R serve as bases for *one-handed posture recognition*. A specific combination of features defines a certain posture. These one-handed postures are recognized by a decision tree-based classifier method. The decision tree is built by considering the features. A certain feature is the root and the tree is extended to the leaves by involving more and more features. The feature values give the direction of the search in the tree, and in the leaves of the tree the postures are found. The decision tree classifier is applied both for the left and right hand individually.

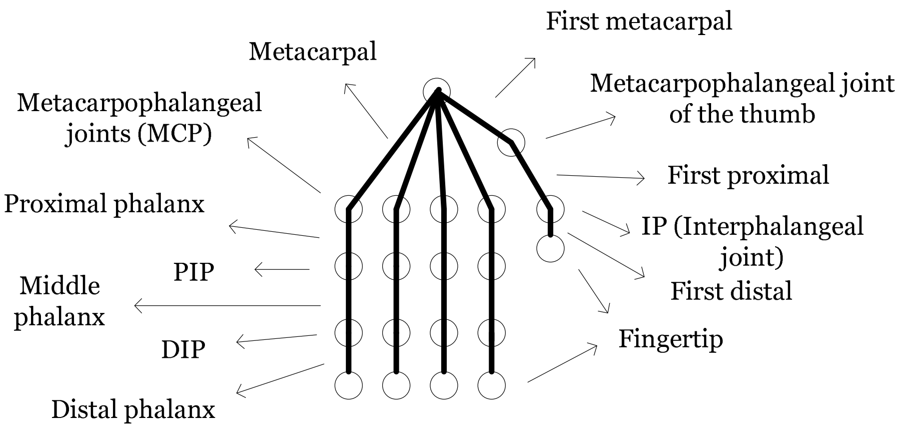


Figure 3.5 Bones and joints of the hand

The postures of the left hand (C_L) and the right hand (C_R) are combined with the help of a set of rules, which contains the meaningful posture combinations. This is the process of *two-handed posture recognition*. A two-handed posture is therefore defined as $C = (C_L, C_R)$. The values compared to C_L and C_R are the names of the postures, and the values assigned to C are the names of the HML words or HML word groups.

When an HML word group is found, it requires further processing. An identifier is assigned to each group, which enables the triggering of different functions related to groups. This is the last step of HML word recognition, called *handling of sign groups*. For each sign group additional parameters are obtained and/or the modeling context provides the information for HML word selection.

3.6 Principles for information completion

Information incompleteness of the hand postures and the hand motion trajectories is typically caused by measurement problems of optical sensing devices. Occlusion of certain landmarks of the hand occurs frequently when the two hands overlap each other from the camera point of view or when a given hand posture is shown (e.g. fist, which occludes the fingertips). In this section I analyze these occlusion problems, and propose principles for completing information about hand postures and hand motion trajectories.

Table 3.1 Cases for information incompleteness of the index finger

Occluded landmark of the thumb	Influence on HML	Completability
MCP	semantic information	no
IP	semantic and geometric information	yes
fingertip	semantic and geometric information	yes

Table 3.1, Table 3.2, Table 3.3 and Table 3.4 list the influences of occluded joints for the thumb, index finger, middle finger and ring finger. The little finger does not provide any information for the recognition of HML words or for surface generation and therefore it is not considered. In the tables, influence on semantic information means that a missing landmark position causes the HML interpreter fail and the word of HML cannot be recognized. Influence on geometric information either means that the intended surface cannot be reconstructed completely or that the data for the manipulation command corresponding to the performed HML word cannot be extracted.

Table 3.2 Cases for information incompleteness of the index finger

Occluded landmark of the index finger	Influence on HML	Completability
MCP	semantic information	no
PIP	geometric information	yes
DIP	-	yes
fingertip	semantic and geometric information	yes

In Table 3.1, Table 3.2, Table 3.3 and Table 3.4 a situation is assumed in which all landmark points are detected (meaning a marker is placed on the landmark points) and what was interesting for us whether in case of the occlusion of one marker the related information can be computed or not. This is indicated in the last column of the tables.

I rely on the kinematical structure of the hand in order to reconstruct the missing information. Basically, there is no way to accurately reconstruct the position of the MCP joints because of the adductive movement of the hand.

However, due to the simple kinematical structure of the fingers, the positions of the DIP and PIP joints and the position of the fingertips can be reconstructed with the assumption that the lengths of the phalanges are known. Knowing that the angle at the DIP joint is the 2/3 of the angle at the PIP joint, the position of the fingertip can be calculated using forward kinematics and the position of the PIP and DIP joint using inverse kinematics.

Table 3.3 Cases for information incompleteness of the middle finger

Occluded landmark of the middle finger	Influence on HML	Completability
MCP	geometric information	no
PIP	geometric information	yes
DIP	geometric information	yes
fingertip	semantic and geometric information	yes

I focused on minimizing the number of the detected landmarks of the hand, and built the model according to this goal, by eliminating the need of measuring the positions of the DIP and PIP joints. The hand model stores information about the kinematical structure of the hand and its main purpose is to extend the set of detected information. This extra information is generated in order to be able to reconstruct the shape of the hands, which is the basis of surface generation. On the basis of the detected data, missing joint positions can be calculated.

Table 3.4 Cases for information incompleteness of the ring finger

Occluded landmark of the ring finger	Influence on HML	Completability
MCP	-	no
PIP	-	yes
DIP	-	yes
fingertip	semantic information	yes

I grouped the upper limb-related kinematical constrains into three groups, namely, macro-level constraints, posture-level constraints and micro-level constraints. Actually, macro-level constraints deal with arm movements and are mainly important to be able to design a comfortable workplace for the designers applying the words of HML. Posture-level constraints consider those postures which are impossible or uncomfortable to be formed in order to be able to avoid them in the construction of the set of hand motions. Micro-level constraints affect the movement of the fingers and play an important role in the surface generation process. These constraints will be discussed in this top-down order in the following sections.

3.6.1 Macro-level kinematical constraints

It has to be emphasized that the words of HML are symmetric, which means that they can be performed either with the left hand or with the right hand. In case of double-handed words, again, the postures forming the word can be taken up by either hand. Therefore, the description for the hand model in Section 3.6.3 is valid for both hands and the two hands and arms are constructed to form a complete Upper Limb Model (ULM).

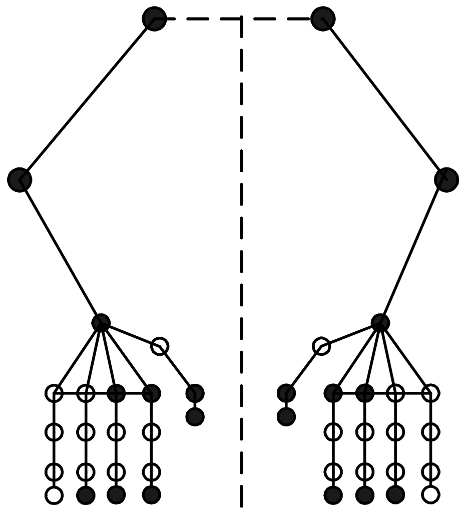


Figure 3.6 A simplified representation indicating the main components of the upper limb model (without parameterization)

Figure 3.6 shows the landmarks, which were selected to be detected, with black dots. It is assumed that the shoulders of the designer do not move (actually a minimal uncontrolled movement is assumed) during the hand motions. The motion envelope of the two arms defines the working space of the designer.

The human hand is highly articulated. The bones and joints of the hand were presented in Figure 3.5. Table 3.5 analyzes the degrees of freedom the human arm and hand have. It is assumed that the shoulder does not have translational movement, only 3 DoF rotational movements. The elbow has 2 DoF and the wrist has 2 DoF rotational movements. Each of the four fingers has 4 DoF. The DIP joint and the PIP joint each has 1 DoF and the MCP joint has 2 DoF due to flexion and abduction. The thumb has a different kinematical structure from the fingers. Its IP joint has 1 DoF and its MCP joint has 2 DoF. The thumb also has a special joint, called trapeziometacarpal (TM), which connects the thumb to the wrist and allows flexion and abduction. Since I do not need to consider the sophisticated motion of the thumb, I eliminate this joint in the model.

Table 3.5 Degrees of freedom of the human arm and hand

Joint	DoF
shoulder	3
elbow	2
wrist	2
Thumb MCP	2
Thumb IP	1
Index, middle, ring and pinky MCP	2
Index, middle, ring and pinky PIP	1
Index, middle, ring and pinky DIP	1
Thumb, index, middle, ring and pinky tip	0
17 joints + 5 fingertips	**26 DoF**

Workplace design is thoroughly studied in the field of ergonomics. As a group of designers is assumed to sit around a table, I studied the ergonomic rules, which are created to avoid fatigue of muscles, of sitting workplaces. The ergonomic rules were adopted from (Eggleton, 1983) with keeping in mind that in this case the graspable objects are virtual, therefore no weight constraints have to be considered. The basic rules for a sitting workplace are:

- Any object that is to be frequently grasped should be located within 15-36 cm of the front of the work surface. These ranges are the distances from which small objects can be procured without requiring the operator to bend forward. This rule is to be applied for the active workspace, where the surface generation and manipulation happens.

- It is permissible to have the working person occasionally lean forward to procure something outside the work area, but such reaches should not be made a regularly occurring part of a brief work cycle. Such an area can be used for storing passive objects after applying the *put aside* word of HML on them. By using the *bring in* word of HML, they can be set active again.

- As far as the working height is concerned, a majority of manual tasks are most easily performed if the work is at elbow height. This implies that seated workplaces should be provided with adjustable chairs to enable collaboration of designers with different heights.

3.6.2 Posture-level constraints

As it is stated in (Lin et al., 2000), hand motion is constrained in a way that it cannot take up arbitrary hand postures. For instance, the pinky finger cannot be bent without bending the ring finger, or the bending of the MCP joint of the index finger implies the bending of the MCP joint of the middle finger as well. Figure 3.7 illustrates some hand postures that are unlikely to be formed by most people, without applying forces on the hand.

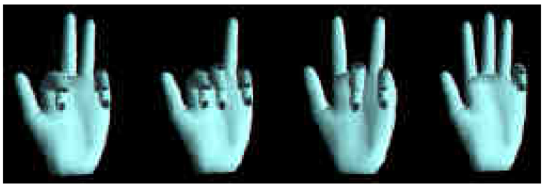


Figure 3.7 Some impossible hand postures (courtesy of Lin, W. et al., 2000)

These postures are important in the hand motion interpretation process. First of all, extremely unlikely postures show that something might have gone wrong during the detection. For instance, it might help to discover that a marker was incidentally misplaced, which would have consequences on the recognition of the words of HML. Secondly, this knowledge can help during the recognition of HML words. As it was described in Section 3.5, in order to recognize the words of the HML, a small number of landmark positions is enough to be detected. Therefore, I did not have to deal with all of the DoFs and all the impossible postures.

However, knowing those impossible postures, which can be discovered by the small number of detected features that were used, improved the recognition process. More specifically, postures are described by a combination of features, and some specific combinations refer to postures that cannot be formed by the human hand. I analyzed all feature combinations systematically, and excluded the impossible ones from the recognition process.

Table 3.6 Impossible feature combinations

DTIT	DTIF	DIMF	SIF	SMF	SRF
close	close	?	?	?	?
close	?	?	TF	?	?
?	?	close	MF/E	TF	?
?	?	close	TF	MF/E	?

Table 3.6 shows the abovementioned impossible feature combinations. The question marks in the table can be replaced by any value of the specific feature.

More precisely, the specified features already define that a number of combinations are impossible. Actually, out of 64 variances of feature values 34 are impossible and the rest are those postures, which can be formed by the human hand, but are not elements of the HML. At last, but not at least, these latter formations of the hand are those postures, which most typically indicate the change between two words of HML, and therefore provide important information for segmentation.

3.6.3 Micro-level kinematical constraints

Micro-level constraints refer to the limits of the motion of the fingers. Commonly these constraints are categorized as static and dynamic ones. As defined in (Lin et al., 2000), static constraints can be expressed as inequalities:

$$0° \le \alpha_{MCP,F} \le 90°$$
$$0° \le \alpha_{PIP} \le 110°$$
$$0° \le \alpha_{DIP} \le 90°$$
$$-15° \le \alpha_{MCP,AA} \le 15°$$

F refers to flexion and AA refers to abduction-adduction in the inequalities. Another commonly applied approximation is that the abduction-adduction movement of the MCP joint of the middle finger is so small that it can be eliminated. There is one major dynamic constraint regarding finger motion, namely, $\alpha_{DIP} = \frac{2}{3}\alpha_{PIP}$. This approximation will be used in the following description of the hand model.

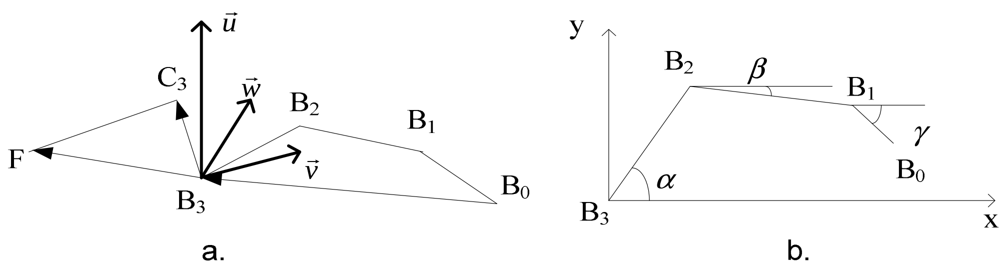


Figure 3.8 Basic settings for calculation

The process of calculating the positions of the PIP and DIP joints of the index finger is described (Figure 3.5). Let's denote the wrist by F, the MCP joint of the index finger by B_3, the PIP joint by B_2, the DIP joint by B_1, its fingertip by B_0 and the MCP joint of the middle finger by C_3. The basic settings for calculating B_2 and B_1 are shown in Figure 3.8.

First a reference frame is defined with origin B_3 (Figure 3.8 a): $\vec{u} = \overrightarrow{B_3C_3} \times \overrightarrow{B_3F}$; $\vec{w} = \overrightarrow{B_0B_3} \times \vec{u}$; $\vec{v} = \vec{u} \times \vec{w}$. The point B_0 is denoted as B_0' in the reference frame $\vec{u}, \vec{w}, \vec{v}$ and its coordinates are computed as $\overrightarrow{B_0'} = \overrightarrow{B_3B_0} \cdot \underline{\underline{M}}$.

The system of equations to be solved is:

$$\begin{aligned} &\left|\overrightarrow{B_3B_2}\right| \cdot \cos\alpha + \left|\overrightarrow{B_2B_1}\right| \cdot \cos\beta + \left|\overrightarrow{B_0B_1}\right| \cdot \cos\gamma = x_{B_0} \\ &\left|\overrightarrow{B_3B_2}\right| \cdot \sin\alpha - \left|\overrightarrow{B_2B_1}\right| \cdot \sin\beta - \left|\overrightarrow{B_0B_1}\right| \cdot \sin\gamma = y_{B_0} \end{aligned} ;$$

$$\gamma = \frac{3}{2}\beta$$

The computed points B_1 and B_2 are used in the surface generation and surface manipulation processes presented in the following sections.

3.7 Theory of surface generation

Surfaces are generated based on the segmentation of hand motions. When a segment corresponds to a $w_i \in W_G$ geometric word of HML, a surface is created based on the T motion trajectories of the hand. Actually, surfaces are formed by using a $T_S = P_S \times \Delta_{P_S}$ subset of the T trajectories, where $P_S = \{P_1, P_6, P_9, P_{10}, P_7\}$ and therefore $T_S = \{\tau_1, \tau_6, \tau_9, \tau_{10}, \tau_7\}$ (see Figure 3.4).

Six types of three-dimensional surfaces can be generated using the HML. These surfaces correspond to a unique word of HML and therefore the type of the surface is identified by the semantics of the HML, more precisely, the recognized word of HML. The geometric words of HML are: *plane*, *cylinder*, *cone*, *sphere*, *ellipsoid* and *freeform*.

3.7.1 Surface identification

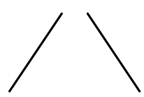
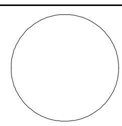
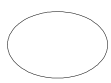
In the process of modeling with the HML, the surface generation process starts with the *start* and ends with the *end* word of the HML. Between these two words, a series of points of multiple trajectories are collected without the indication that the objective of the generation is a cylinder or regular shape or a freeform surface.

Therefore, the type of the surface has to be recognized, for which I could only rely on the abovementioned collected points.

To enable the identification of the intended shape, the deviation of the generated surface and a best-fit surface is searched. If the deviation is below a certain limit, the corresponding best-fit shape is visualized. The limit can only be defined by user experiments which study the maximum deviation of the generated and the best-fit shape in a statistical analysis. This study is reported in Section 6.2.

The moment of comparing the generated and the best-fit shapes is always the end of the performed word of HML, after the collection of measured points. There are five best-fit shapes to be compared as it is listed above. Best-fit planes and cylinders are constructed based on the $T_S = \{\tau_1, \tau_6, \tau_9, \tau_{10}, \tau_7\}$ trajectories and best-fit lines, circles and ellipses based on the τ_6 trajectory. The reason for comparing three-dimensional shapes in the case of plane generation is the fact that due to the kinematics of the arm, the hand moves in a circular pattern unless it is purposefully controlled by an unnatural movement. In the case of the cylinder the reason is simply the curvedness of the shape which does not let us to confuse the cylinder with two parallel planes.

Table 3.7 Description of best-fit shapes

Shape name	Mathematical formula	Shape by the MCP joints
plane	$(\vec{n} \cdot \vec{p} + d)/\lvert\vec{n}\rvert$	
cylinder	$\lvert\vec{p} - \vec{p}_0 - \lvert\vec{a}\rvert \cdot \vec{u}\rvert - r$ $\vec{a} = \vec{u}(\vec{p} - \vec{p}_0)\,;\, \vec{u} = \vec{v}/\lvert\vec{v}\rvert$	
cone (2 lines)	$\lvert\vec{p} - \vec{p}_0 - \vec{a} \cdot \vec{u}\rvert$ $\vec{a} = \vec{u}(\vec{p} - \vec{p}_0)\,;\, \vec{u} = \vec{v}/\lvert\vec{v}\rvert$	
sphere (circle)	$\left(r - \lVert\vec{p}_a - (\vec{p}_a \cdot \vec{n}) \cdot \vec{n}\rVert\right)^2 - (\vec{p}_a \cdot \vec{n})^2$ $\vec{p}_a = \vec{p} - \vec{p}_c$	
ellipsoid (ellipse)	$(\vec{n} \cdot \vec{p}_{3D} + d)/\lvert\vec{n}\rvert$ $\lvert p_{2D} - p_c\rvert - \lvert\vec{p}_d\rvert / \sqrt{(X_d \cdot \sin\alpha)^2 + (Y_d \cdot \cos\alpha)^2}$ $\alpha = \measuredangle(\vec{p}, \vec{p}_c) - \alpha_T$	

In the other three cases, it is enough to compare the pattern of one motion trajectory formed by the MCP joints of the middle fingers. Table 3.7 shows the mathematical formulas regarding each shape which have to be minimized in order to get the corresponding best-fit shape. In the formulas $\vec{p}$ is the actual measured point, $\vec{n}$ is the normal vector of the two-dimensional shapes, $\vec{p}_0$ is a point on the line or on the center line of the cylinder, r is the radius of the cylinder or circle, $\vec{v}$ is the direction vector of the line or center line of the cylinder, $\vec{p}_c$ is the center of the circle or ellipse, $\vec{p}_{2D}$ is the projected point on the best-fit plane of the actual measured point, $\vec{p}_d\left(X_d, Y_d\right)$ and X_d or Y_d is the distance from the center to the perimeter of the ellipse along the X or Y axis and α_T is the rotation of the ellipse in counter-clockwise direction.

For the measured points all of the best-fit shapes are generated and their deviation is computed. If none of the deviation of the generated and the computed best fit shapes is below under the required limit, a freeform surface is generated. Regular shapes are generated based on their specific parameters, such as center, radius or height, which are obtained during the minimization process.

3.7.2 Surface generation

Freeform surfaces are generally created based on a general sweep, by moving a curve which can change its shape during the surface generation process (Juhász, 1995). I used the middle finger as generatrix and its motion as directrix. The measured landmark points are the wrist ($\vec{p}_1$), the MCP joint ($\vec{p}_6$) and the fingertip ($\vec{p}_7$) of the middle finger. These points are extended with the computed points of the PIP joint ($\vec{p}_9$) and of the DIP joint ($\vec{p}_{10}$) with the help of the hand model, according to the description in the previous section. This procedure is done for each frame of the motion sequence defined by the given segment.

Based on the collected points a Bézier-surface is generated. Let's assume that we move a Bézier-curve which is given by its $\vec{p}_i; i = 1, \ldots, 5$ points and that the control points of the Bézier-curve are also moving on Bézier-curves. These $\vec{p}_i$ points are actually the measured and computer landmark points of a finger as it is described above. Let's denote the control points of the Bézier-curve which define the trajectories of the $\vec{p}_i$ points by $\vec{p}_{ij}; j = 0, \ldots, n$, where $n+1$ is the number of the measured frames of motion. This means that our Bézier-surface is defined as

$$\vec{b}(u,v)=\sum_{i=1}^{5}\left(\sum_{j=0}^{n}\vec{p}_{ij}B_j^n(v)\right)B_i^4(u)=\sum_{i=1}^{5}\sum_{j=0}^{n}\vec{p}_{ij}B_i^4(u)B_j^n(v),$$

where $B_k^m(t)=\binom{m}{k}t^k(1-t)^{m-k}$ are the Berstein polynomials as defined in (Juhász, 1995).

Complicated surfaces need several sweeps to be composed of. Let's assume we have two Bézier-surfaces: $\vec{c}(u,v)=\sum_{i=1}^{5}\sum_{j=0}^{n}\vec{c}_{ij}B_i^4(u)B_j^n(v);u\in[u_0,u_1];v\in[v_0,v_1]$ and $\vec{d}(u,v)=\sum_{i=1}^{5}\sum_{j=0}^{n}\vec{d}_{ij}B_i^4(u)B_j^n(v);u\in[u_1,u_2];v\in[v_1,v_2]$. If we want them to be connected along the $u=u_1$ parameter line with an order of continuity r, the following equations should be realized: $\left(\frac{1}{\Delta u_0}\right)^i\Delta^{i,0}\vec{c}_{4-i,j}=\left(\frac{1}{\Delta u_1}\right)^i\Delta^{i,0}\vec{d}_{i,j};(i=0,1,\ldots,r;\ j=0,1,\ldots,n)$, where $\Delta u_i=u_{i+1}-u_i$ as it is defined in (Juhász, 1995).

3.7.3 Surface generation by VDIM

The triangulation method is used to generate meshes on the point set representation in order to be able to compute the approximated surface normal for each point. Once the surface normal of the points are known, the model can be imported to the VDIM system, and a nominal or a vague shape is generated and visualized (Rusák, 2003). As it was explained in Chapter 1, this modeling system was selected because of its capability of generating and visualizing imprecise and incomplete shapes.

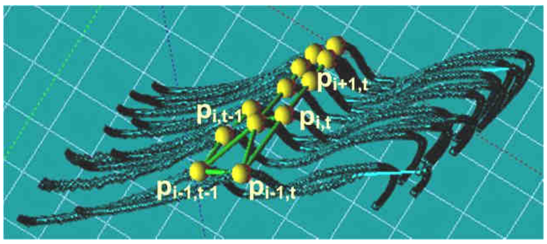


Figure 3.9 Triangulation of motion trajectories

In the VDIM, two methods are offered to define vague shapes. If a surface is defined by a single hand motion, a nominal surface is generated in the VDIM system. By positioning two nominal surfaces together in the modeling space, a vague discrete interval model is defined. If a surface is defined by alternating hand motions, for instance by continuous sweeping, a different approach is followed. When there is a sudden change in the direction of sweeping, the swept trajectories are segmented into pieces. In order to identify the change in the direction of the sweeping, the position and the approximated surface normal of the swept points are investigated for each frame of motion (Rusák et al., 2006).

Let $P_{i,t}$ denote the point of the i^{th} trajectory for the t^{th} frame of motion. Let $n_{i,t}$ be the approximated surface normal vector of $P_{i,t}$, which can be calculated based on the triangulation of the set of eight close neighbor points $\{P_{i-1,t-1}, P_{i-1,t}, P_{i,t-1}, P_{i+1,t}, P_{i-1,t}, P_{i-1,t+1}, P_{i,t+1}, P_{i+1,t+1}\}$ of $P_{i,t}$. $n_{i,t}$ is determined as the average of the surface normal vectors of each triangle, for which $P_{i,t}$ is defined as a vertex.

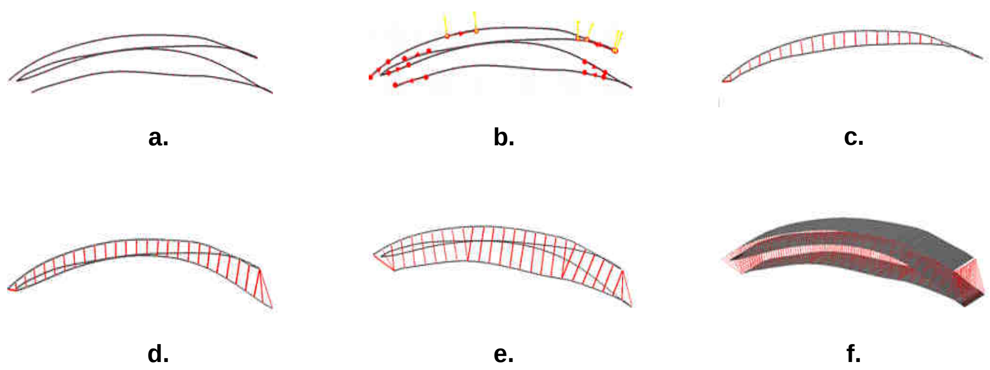


Figure 3.10 Generation of vague discrete model by alternating hand motions

Figure 3.9 illustrates the principle of triangulating motion trajectories. The location of motion trajectory segmentation is defined in the frames where: $\operatorname{sgn}\left(\left(P_{i,t+k} - P_{i,t}\right) * \left(P_{i,t} - P_{i,t-k}\right)\right) = -1$ or $\operatorname{sgn}\left(n_{i,t} * n_{i,t+k}\right) = 1$. It means that if the direction of the i^{th} trajectory in frame t changes more than $\pi/2$ radian compared to the previous frame, and the surface normal vector does not change more than $\pi/2$ radian then the i^{th} trajectory is segmented in frame t.

When segmentation occurs in frame t, a new point-set is created and all points are aggregated into it until the next segmentation occurs in frame $t+m$. When the motion trajectory has at least two segments, a vague interval model can be generated, which defines a discretized interval between the two segments. Figure 3.10 c illustrates the vague model that has been generated between the first two segments of a surface sweeping action. As sweeping goes on, additional surface segments are generated in the modeling space and added to the previously created vague model. Figure 3.10 e shows the accumulative increase of the vague interval model. When the sweeping process stops, the surface generated last is added to the vague model.

3.8 Theory of surface manipulation

As well as surfaces, manipulation commands are also generated based on the segmentation of hand motions. When a segment corresponds to a $w_i \in W_C$ constructive word of HML, a manipulation command is created based on the T motion trajectories of the hand. Manipulation commands are formed as $m_i = \left(w_{C,i}, P_S, F_S\right)$, where $w_{C,i}$ is the HML word which corresponds to the detected E_i segment in the hand motion, P_S is a subset of the P set of detected landmark points of the hand and F_S is a subset of the F set of features. P_S and F_S are unique for each manipulation word in the HML vocabulary. For each manipulation word the corresponding data can be found in Appendix C.

Table 3.8 HML words grouped according to the time of geometric data extraction

Continuous	End of HML word
Turn, Distance by points, Distance by curves, Distance by surfaces, Size by surfaces, Size by curves, Size by points, Angle by surfaces, Angle by edges, Zoom in/out, Put aside/Bring in	Identify point, Identify curve, Identify surface, Identify object, Surface to surface, Surface to curve, or vice versa, Surface to point, or vice versa, Curve to curve, Curve to point, or vice versa, Point to point, Construct/Deconstruct, Compose/Decompose, Assemble/Disassemble, Cut through/Cut out

Surface manipulation happens in the modeling engine by specific algorithms for each manipulation. The input parameters, which are actually represented by P_S and F_S, of these algorithms is collected from the hand motions according to the currently performed HML word ($w_{C,i}$). As the words are different in their nature, different data have to be extracted at different moments in time.

Table 3.8 groups the words of HML according to the time when the related geometric data have to be extracted. It can be either the end of a specific word, or it can continuous during the entire performance of the word. The former means that data have to be extracted in one frame of motion in the end of performing the

HML word, and the latter means that data have to be extracted in each frame of motion during performing the given HML word to be able to provide continuous visual feedback for the designer on his action. Formally it means that a manipulation command corresponding to a segment is $m_i \mid E_j$, and the t_k moment of time in the end of the HML word relates to the last frame in E_j, and when the data extraction is continuous, $\forall t_k \in E_j$ are considered.

Tables in Appendix C show the geometric data to be extracted for all the words of HML and the operation that has to be done in the modeling engine. For the understanding of the notions in these tables refer to Figure 3.5. Section 5.7 specifies algorithms defined for the manipulation operations presented in this Section.

The theoretical fundamentals of hand motion processing have been laid down in this chapter. First, the HML has been formalized and then the complete process of hand motion processing has been defined. The definitions for hand motion interpretation were given in a mathematical form. Actually, hand motion interpretation involves several processes, these are trajectory segmentation, HML word recognition, and surface generation and manipulation. Theories for the implementation of these processes were introduced based on the definitions. The theories defined in this chapter serve as basis for the development of the algorithms of the proof-of-concept system, which are presented in Chapter 5.

FUNCTIONS AND COMPUTATIONAL RESOURCES FOR HAND MOTION PROCESSING

4

Imagine a collaborating team of designers using the hand motion-based system to conceptualize shapes of various products. It is assumed that the hand motions and the visualization of the shapes happen in the same workspace, i.e. in the design space, simultaneously. Imagine that you see the surfaces appear directly "under the hand" of the designer. With this imagination in mind, this chapter discusses the requirements for shape conceptualization and presents the functional specification of the proposed hand motion processing system. The data sources, flows and structures of the system are defined. The processes of modeling and visualization are discussed with a view to the applied modeling engine, the Vague Discrete Interval Modeler. A feasibility study of the applicable tracking technologies was performed to be able to select the best tools for the proof-of-concept development of the system. The results of this study are presented and discussed in the end of this chapter.

This chapter is based on the paper (Varga et al., 2005b) with the necessary adjustments according to the differences between the initial ideas and the final implementation.

4.1 Identified functions of the proposed system

Based on the research hypotheses presented in Section 1.7 and on the conceptualized system shown in Figure 1.6, the set of functions which are needed to be implemented was identified.

The first three functions are related to the *capturing of hand motions* are

- tracking of hand motions,
- adaption to different hand sizes of users and
- supporting interaction using both hands.

The functions concerning the *identification of the intention of the user* are

- recognition of hand motions and

- adaption to differences of hand motions, because different users may perform the same command in a slightly different way.

The identified functions regarding the *surface generation and model building* processes are

- generation of additional information to be able to create surfaces, because I decided to track only selected landmarks of the hands,
- conversion of hand motions to surface patches and
- conversion of hand motions to shape modeling operations.

In the next section, these functions are elaborated and structured in a hierarchical form in order to provide a basis for identifying the data flows in the proposed system.

4.2 Functional specification

Based on the functions presented in the previous section, the hierarchical structure of system functionalities was established (Figure 4.1). As a possible scenario, I propose the usage of dataglove as input device and monitor as output device, however, according to my imagination the ideal scenario uses optical tracking as input device and holographic display for truly three-dimensional visualization. Four main groups of functions were identified, these are (i) information acquisition, (ii) information completion, (iii) information conversion and (iv) modeling and visualization.

The first main group of functions is ***information acquisition***. Before starting the modeling session, the users need to be *prepared*. Preparation depends on the type of the input device. In case of using a dataglove, the glove is put on the hands, and when optical tracking is used, the markers need to be placed to the right positions of the user's hands. To start the tracking process, the detection device needs to be set to *alert* state. When the device is ready to be used, the actual *sensing* starts. Two types of sensors are available in the market. Datagloves measure specific angles at the joints of the hands. Other sensors, such as the ones used in magnetic and optical tracking, measure the three-dimensional position and sometimes orientation of markers attached to specific landmarks of the hands. *Post-processing* of the information is needed, if the measured data are angles. In this case the calculation of the position of the landmarks needs to be done by applying forward kinematics algorithms. Finally, the information has to be *transferred* to the next stage of processing, which generates extra information.

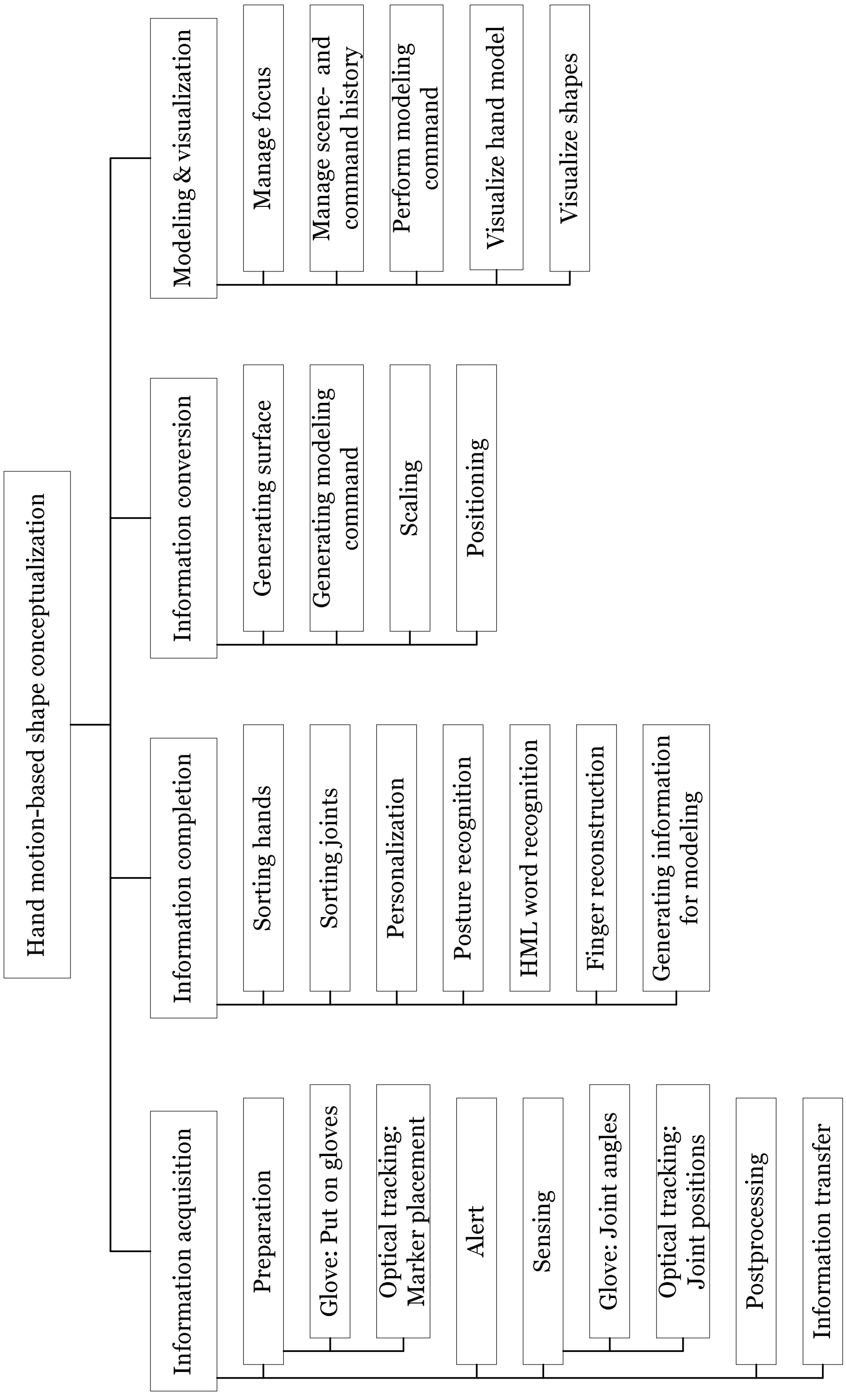


Figure 4.1 Functions of the proposed system

The next group of functions is called ***information completion***. The data need to be *sorted* according to the left and right hand, and the data coming from the detection device need to be assigned to specific *joints* of the hand model. Hand models need to be *personalized* by setting the sizes of the phalanges according to the hand of the current user. *HML words* performed by the user need to be *recognized* in order to react on the intention of the user. An important part of the HML word recognition process is *posture recognition*, because postures do not change during a single HML word. The *reconstruction of the shape of the fingers* is based on anatomical rules and involves the calculation of positions of joints that are not measured. Furthermore, additional information needs to be generated to be able to perform the recognized modeling command. This additional information comes from the current state of modeling, e.g. which objects are in the scene and what the last command was.

The third group of functions is named ***information conversion***. Based on the recognition of the HML word, either a *surface* or a *modeling command* is generated. When the performed HML word is of constructive nature, the command is the generation of a surface, either freeform or regular, such as a sphere, cylinder or plane. Other commands are related to selection or manipulation of surfaces by setting their sizes, positions, spatial relations or modifying their structure. *Scaling* and *positioning* are needed according to the used visualization device. Finally, the generated surface is *transferred* to the modeling space for visualization and further manipulation.

The last group of functions is ***modeling and visualization.*** Modeling refers to the management of information in the system, in terms of storage and transfer between modules. *Focus, scene information and command history* need to be obtained and stored to facilitate the realization of modeling commands. If the output device is not a truly three-dimensional one, the hand model of the user needs to be visualized as avatar for virtual presence. The identified modeling command needs to be *performed* by the modeling system (Section 4.4), and the result, more precisely, the generated surface has to be *visualized* on the output device as feedback for the user.

4.3 Data sources, flows and structures

4.3.1 Data sources

The source of motion information is the *user*. The designer’s hand motions – more specifically the HML words - serve as input for shape generation. Hand motions are scanned by a special device, which is able to track the motion and provide us with not only the trajectory of the global hand motion, but also with the *motion trajectories of specified parts* of the human hands. Actually, *three-dimensional coordinates* of points are the outcome of the scanning process.

The hand model and the surface creation process generate *extra points*. Some *characteristic features* of the hand were identified, which can be defined by calculating the *distance of specific points*. These distance values provide the input for hand motion interpretation, which identifies the beginning and end of the words of HML and assigns the indicated word with a *numeric identifier*. This way the indicated modeling action can be defined and transferred to the modeling engine. The corresponding shape modeling command is constructed and performed by the Vague Discrete Interval Modeling (VDIM).

4.3.2 Description of the Vague Discrete Interval Modeler

For a better understanding of the applied modeling engine, a short description of the VDIM is given in this section (Rusák, 2003). Figure 4.2 illustrates the VDIM method of shape conceptualization through an application example. The VDIM method takes one or two point sets and attaches some uncertainty to the location of the points to define particles. These particles describe a probable interval, where the shape of a concrete object may exist. A finite set of discrete particles form a particle cloud that defines either a region of an uncertain global shape, or an uncertain shape feature. Particle clouds are generated either by describing the uncertainty of the contained points, or by deriving distribution trajectories between the corresponding points of two (or more) non-coincident discrete point-sets.

Vague discrete interval models represent a cluster or a family of shapes. From a design point of view it usually gives the global shape of objects. Whenever an individual shape with specific morphological properties is needed, an instance must be generated from the interval representation. The principle of shape instantiation is that the semantic content of certain shape formation rules are converted to information about the placement and necessary local adjustments of the derived instance shape. In order to capture the semantics and convert it into computable functions, a three-step mapping procedure has been developed. On the highest level, shape formation principles provide non-numeric knowledge about how to derive instances from the shape interval. From these shape formation principles the types and number of instantiation operators can be derived based on the structural representation of the model. Also the shape formation rules, which are linguistic expressions of shape modification, are extracted from the shape formation principles.

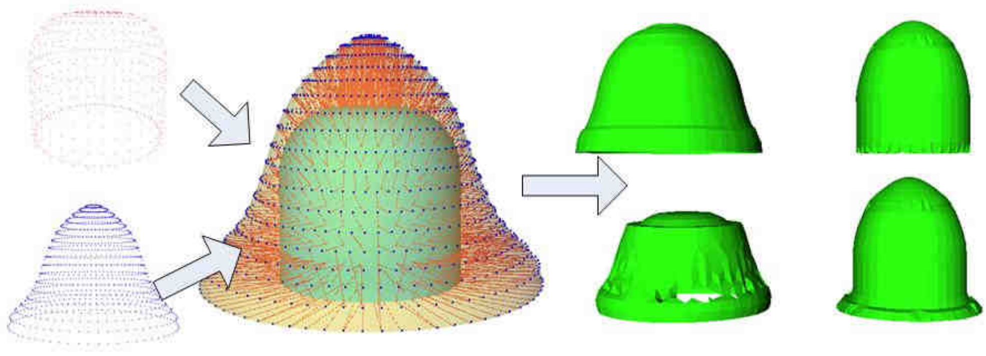


Figure 4.2 Shapes created in the VDIM

To map a shape formation rule to an instantiation function, the morphological changes that the effect parameters are able to introduce to a region of a shape were analyzed. The ultimate goal was to express these morphological changes with simple linguistic expressions that can be intuitively understood and applied by designers. The formal representations of the propositional structures of these linguistic terms are called shape formation rules. In practice, eleven basic linguistic expressions (e.g. offsetting, tilting on corner, symmetric warping, symmetric scalloping, symmetric saddling, curving, etc.) have been defined, that are mapped onto effect parameters introducing changes in the shape. By applying logical compositions or extra constraints on these rules, new rules can be generated in the VDIM. From shape formation rules, the parameters of the instantiation operators are derived that are specific to a given part of the shape. Unique features of a shape have to be identified by the user, who can select regions on the shape. In the last step, the instantiation operators are applied to the shape with the defined values of the parameters of the instantiation functions.

There are two methods to define vague shapes. If a surface is defined by a single hand motion, a nominal surface is generated in the VDIM system. By positioning two nominal surfaces together in the modeling space, a vague discrete interval model can be defined as it was illustrated in Figure 4.2. If a surface is defined by alternating hand motions, for instance by continuous and waving sweeping, a different approach is followed. A vague model is constructed using the lower and upper boundaries of the waving motion as described earlier in Section 3.7. The instantiation of the shapes is through the functions of the VDIM using keyboard and mouse control, or alternatively by hand motions as defined in (Rusák et al., 2006).

4.3.3 Data flows

In Figure 4.3 the dataflow diagram shows the information flows of functions, according to Figure 4.1.

When capturing hand motions with a *tracking* device, the result is raw data of measured spatial positions in each frame of motion, which in most cases contains some momentary measurement errors. The most common error is, when a marker is occluded and its position cannot be measured. The removal of these errors is what I call *post-processing*. The cleaned motion information is used to *activate the hand model*. In this process, the unmeasured landmark points of the hands are calculated. They are mapped to the joints of the hand model, which is activated, and reconstructs the motion of the real hand. The hand model is *personalized* by setting the sizes of the phalanges of the current user. The hand model receives the motion trajectories of the wrists, head of metacarpal bones and fingertips. This is enough information to calculate the positions of missing joins using inverse kinematical algorithms.

I built the hand model in a way that it provides descriptive features for the interpretation of the words of HML. Interpretation is a complex process and described in Section 5.3. When the HML word is interpreted, the corresponding modeling action is selected. The modeling action is either *surface generation*, if the command is of constructive nature, or surface manipulation, if the command is of manipulative nature. The generation of the *modeling command* happens based on obtaining information from the modeling scene and from the command history, with an attention to the format of the command applied by the modeling engine. Finally, the generated surface is *scaled and positioned* according to the operational parameters of the output device, and *visualized* to the user.

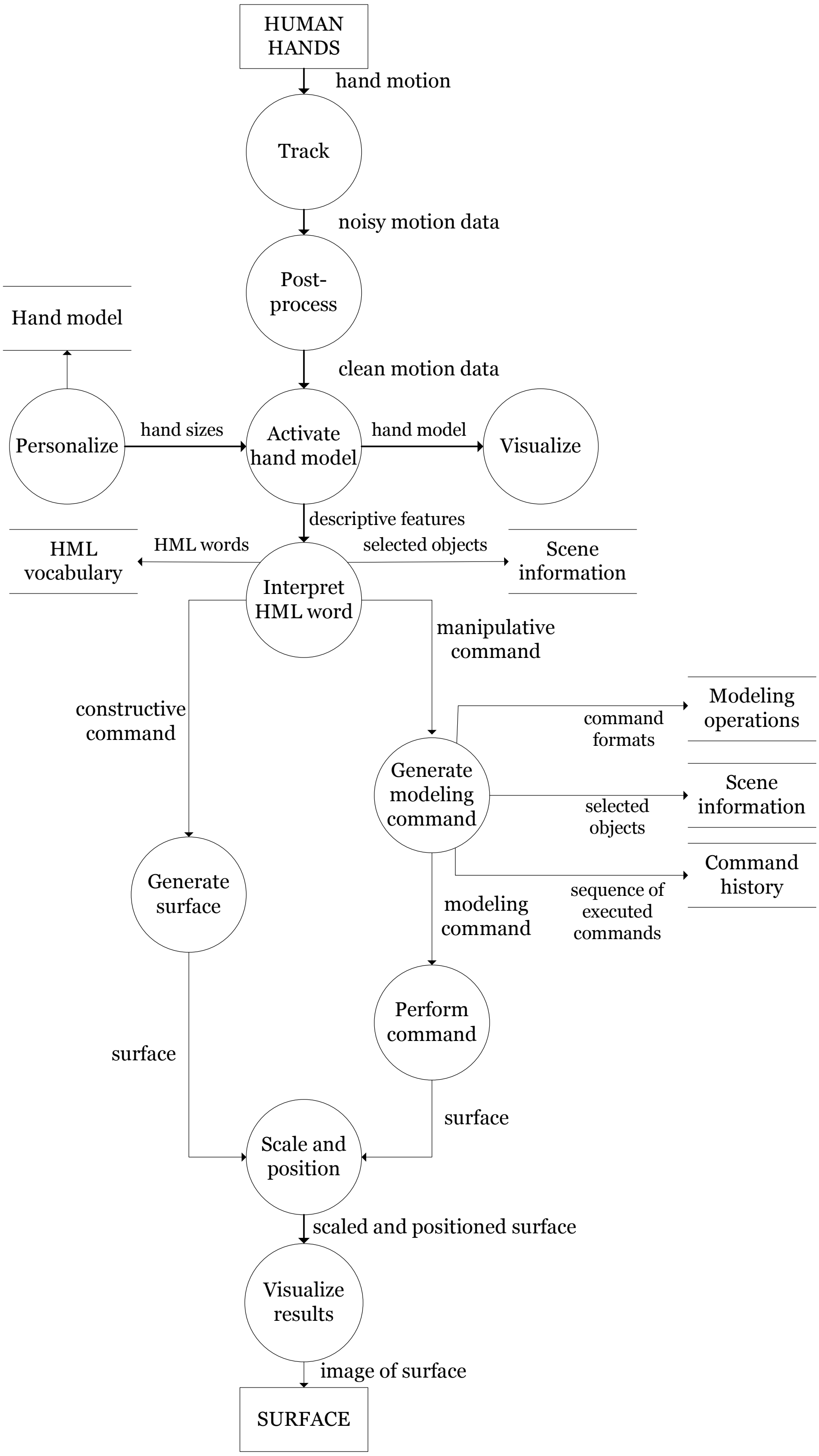


Figure 4.3 Dataflow diagram of hand motion processing

4.3.4 Specific data structures

The objective of this section is to summarize the types of data applied during hand motion processing. Figure 4.4 summarizes the pieces of data that is needed to operate the HML-based interface. I identified three major parts in the complete process of hand motion processing where different data structures have to be handled. These are detection and tracking, hand modeling and shape modeling.

The *detection and tracking* part deals with the information about the measures of the space, where the hand motions occur. Another important aspect is the operational parameters of the detection equipment, such as speed or accuracy. These different types of data have an influence on each other, e.g. the measures of the space to be detected affects the accuracy of detection. Obviously the input for the detection and scanning part are the hand motions, more precisely the words of HML, and the obtained outcome is the motion trajectory information of the hand, which in this case involves multiple motion trajectories of specific landmarks of the hand as it is fully described in Section 5.2.

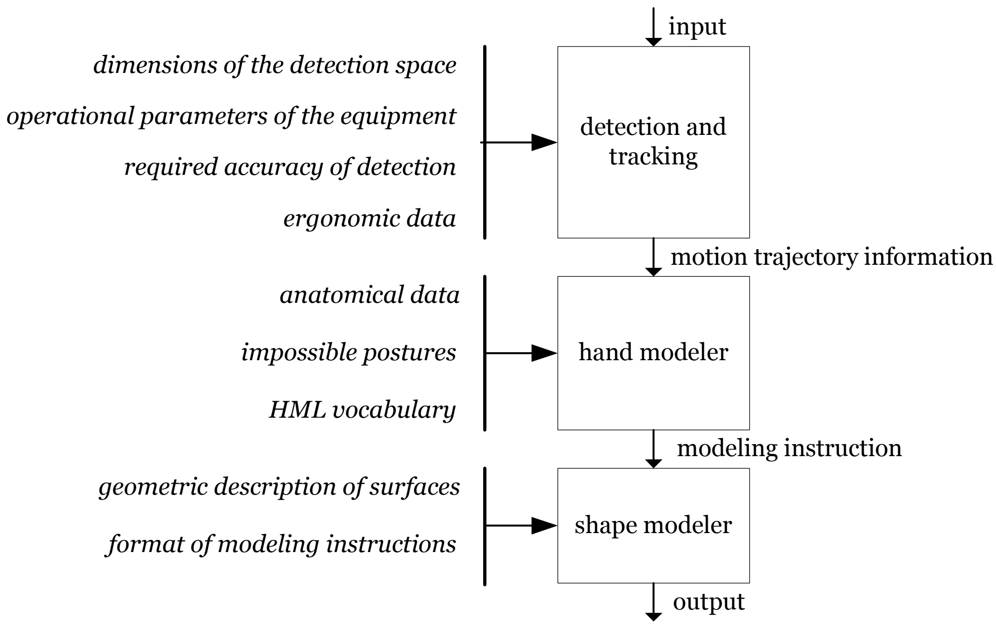


Figure 4.4 Data structure of the system

The *hand modeler* handles different types of data. (i) Anatomical data are used to generate the unmeasured trajectories of the fingers. (ii) Impossible postures are built in the hand motion interpretation process to improve the interpretation. (iii) Descriptive features of the hand model define the words of HML. Additional pieces of information are (ii) the theory based on which the detected motion trajectories are segmented to meaningful and meaningless parts (Section 3.4), (iii) the theory based on which the words of HML are modeled and identified (Section 3.5), (iv) the theory based on which surfaces are generated (Section 3.7), (v) the theory based on which the motion trajectories are converted to modeling instructions (Section 3.8). Based on the interpreted word of HML a surface or a modeling instruction is formed with respect to the format of the command used by the *modeling engine*.

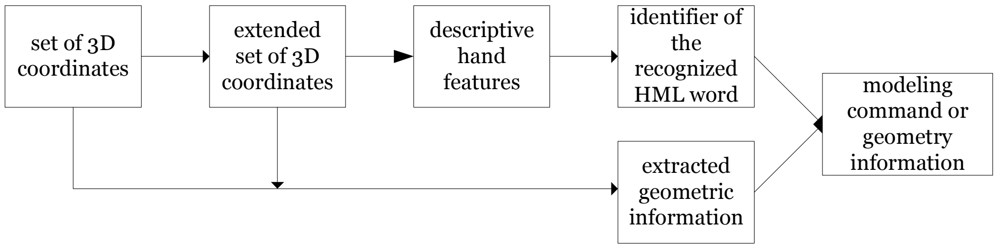


Figure 4.5 Different types of data in the complete process of hand motion based modeling

Figure 4.5 shows the main data types which occur in the complete process of modeling by hand motions. Three-dimensional coordinates are tracked by detection device from the markers which are attached to the designer's hand before the tracking process. This set of coordinates is extended by the kinematical model of the hand in order to generate surfaces according to the shape of the hand and to be able to interpret the action of the designer. Based on these positional values, some characteristic features of the hand are calculated in the form of distance values.

These distance values provide the input for the hand motion interpretation process, which identifies the word of HML showed by the designer. In the end of this process I get the identifier of the HML word either in textual or numerical form. Based on the identified HML word, the related geometric information is extracted from the motion trajectories of the hand. The identifier of the performed HML word and the extracted geometric information together provide the input for the modeling engine, which constructs the modeling operation out of these data. As it was described earlier, the applied modeling engine is the VDIM. The next section explains the connection of the hand motion-based interface to the VDIM.

4.4 Modeling and visual processing by the Vague Discrete Interval Modeler

To create a modeling environment, the HML interpreter was developed and connected to VDIM as input means. The main issues concerning the realization of this connection are (i) the adaptation of the HML to the VDIM commands, (ii) the data communication between the HML interpretation software and the VDIM, (iii) processing of the system commands which govern the model building process, (iv) processing of the geometric commands enabling the generation and composition of surface patches and (v) visualization of the geometric components with special attention to positioning and scaling.

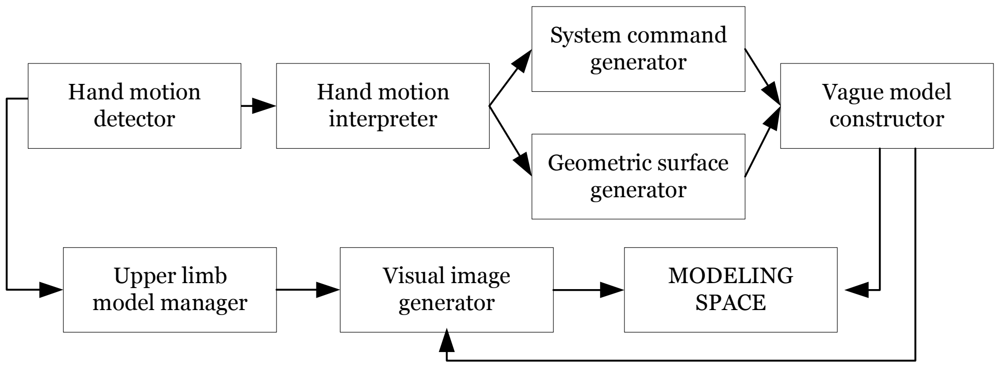


Figure 4.6 Connection of HML and VDIM

Figure 4.6 illustrates the connection of the hand motions to the VDIM. The user performs hand motions in the modeling space. The hand motions are measured by the hand motion detector, and the outcome of the detector serves as input for the HML interpreter. The HML interpreter either produces a geometric or a manipulation command, which goes to the geometric surface generator or to the manipulation command generator accordingly. The geometric and manipulation information provides the input for the vague model constructor. Ideally, the visualization of the results happens in the modeling space, e.g. using truly three-dimensional visualization devices. If this is not the case, the hand motion detector produces information for the upper limb model manager as well, and the upper limb model, as an avatar, and the generated shape model are visualized in the visual image generator.

4.5 Testing of the framework of the proof-of-concept system

The feasibility study was designed with the goal to (i) find solutions from implementation point of view and (ii) qualify the solutions for operational criteria. In each part of the system different realization possibilities in terms of methods, tools and hardware elements are available.

Table 4.1 Criteria defined for the feasibility study

Function	Subfunction	Mathematical model	Data structure	Procedure	Algorithm
Information acquisition	preparation			usability (25)	
	alert	sensitivity (1)		usability (26)	
	sensing	adaptability (2) fidelity (3) sufficiency (4) real-time ability (5) usability (6) extendibility (7)		reliability (27)	complexity (32) robustness (33) autonomy (34) collaboration (35)
	postprocessing	accuracy (8)			
Information completion	filtering	fidelity (9)	consistency (22)		complexity robustness autonomy intelligence (36) collaboration
	generating trajectories	fidelity (10) sensitivity (11)			
	personalization	fidelity (12)		usability (28)	
	completing trajectories	fidelity (13)			
	comparing trajectories	extendibility (14) modifiability (15) accuracy (16)	flexibility (23)	usability (29)	
Information conversion	generating surfaces	fidelity (17) adaptability (18)	flexibility (24)	usability (30)	
Visualization	display	fidelity (19) usability (20) real-time ability (21)		usability (31)	complexity robustness autonomy collaboration

Feasibility issues are studied with a view to the functions of the system and to the main phases of converting the information from the working environment to processing software, namely mathematical models, data structure, procedures and algorithms. The transition phases and the main functions were arranged in a matrix, and their corresponding criteria were specified to their relationships (Table 4.1). For example, the mathematical model in information acquisition is the tracking method which is used to obtain hand motion information or in case of information completion a hand model, which describes the kinematics of the human hand. As overall criteria cannot be defined for the entire system, individual criteria were selected for each function. These criteria characterize system properties which on the one hand, tell about the conditions of the implementation, on the other hand about the functioning of the system. Although the names of criteria are the same in Table 4.1, they have a different meaning in different contexts. For instance, usability in (24) means, that the better the system the least time has to be spent with preparation. In (25) the size and position of the button is defined, since the user can have problems with pushing a very small button with a hand fully instrumented with markers, or reaching a button which is placed further than the user can move around with the cables.

Table 4.2 A comparison of commercially available tracking methods

Tracking method	Max. number of sensors	Size of Sensors (mm)	Output	Measurement rate (1/s)	Environmental constraints
Electromagnetic	4-120	~23x28x15	position& orientation	100-240	metallic objects affect performance
Acoustic	4-32	~27x39x15	orientation	180-500	background noise
Glove	22	One size glove	joint angles	150	
Active optical	512	~∅3-5	position or position & orientation	4600/number of sensors	line of sight
Passive optical	150	~∅2-3	position	160-484	line of sight

In the next step, measures were defined to describe the criteria. These measures are to compare the alternatives qualitatively or quantitatively. Let's analyze this by comparing the tracking methods listed in Table 4.2. The criteria of sufficiency (4) means that if a surfaces with high fidelity needs to be generated (17), at least three positions of a single finger has to be measured: at the wrist, at the head of metacarpal and at the fingertip. This brings in the next criteria, the usability (6), which shows that tracking methods, which have markers bigger than ~ 1 cm, can not be applied if tracking of the motion of multiple fingers is needed.

As the system is planned to use in collaboration, it is very important to analyze the extendibility of the tracking method. The optical tracking systems can work with a large number of markers, so it can be easily extended to use it with a small group of people. In case of an instrumented glove, a new pair of glove has to be purchased for each user, which results in extra costs. However, the number of markers can influence the measurement rate, which results in lower performance of real-time ability (5). Two conclusions can be derived from this short analysis. Firstly, the criteria of each function should be fulfilled with regards to the final goal, namely, the generation of surfaces. Secondly, criteria of an individual function may have an effect on each other; therefore solutions should be optimized according to their relations.

To make the comparison easier, indices were created. For instance, the tracking method requires adaptability (2) to differences in hand sizes. Ergonomic tables show the hand sizes of the population. To analyze the goodness of the method, it is defined how many percent of the adult population can use the system. The higher the number is, the more adaptive the system is. In case of accuracy (16), a percentage shows how many of the user performed HML words were recognized correctly. The higher the number is, the more accurate the classification method is.

In the following chapter I propose algorithms for the system functions introduced in this chapter, and elaborate on the development of the proof-of-concept system for hand motion processing.

5 ALGORITHMS AND RESOURCE INTEGRATION FOR THE PROOF-OF-CONCEPT SYSTEM

In Chapter 3 I introduced theories for hand motion processing, and in the previous chapter I discussed the functional specification of the proposed hand motion processing system. Basically, now the question is how to make the system work. Therefore, this chapter discusses the algorithms as practical realization of the theories. In fact, these algorithms provided the basis for the implementation of the proof-of-concept hand motion processing system. Requirements for system operations are discussed and the setup of the hand motion detection environment is presented. Algorithms for trajectory segmentation and hand motion recognition are explained as the major elements of the complete hand motion interpretation process. The algorithms for surface generation and manipulation, which actually define the output of hand motion processing and at the same time the input for the modeling engine, are discussed.

This chapter is based on the papers (Varga et al., 2006a) and (Varga et al., 2006b).

5.1 Requirements for system operation

First of all, it should be mentioned that I did not intend to develop a new dedicated measuring technology, since this research concentrated on the creation of a proactive virtual design environment. The forerunning searches revealed that many technologies are already available on the market for hand motion detection.

I identified those basic requirements, which should be met by the preferred hand motion detection technique. In short, these requirements can be sorted to ergonomic and to technical requirements. Fulfillment of ergonomic requirements provides a comfortable and easy-to-use working environment for the designers. The use of cables which connect the equipment worn by the user to the computer was undesirable. Heavy equipment on the designers' hand was also unwanted in order to avoid being tired after a short period of time. In fact, technical requirements came from the nature of three-dimensional shape conceptualization. Three-dimensional data should be obtained in real-time, without restricting the designer's motion and the largest flexibility should be achieved in terms of shape manipulation.

Techniques, like magnetic tracking, active optical tracking and data-gloves fulfill the technical requirements, but not the ergonomic ones. The other way round, passive optical tracking fulfills the ergonomic requirements, but not the technical ones. It restricts the motion of the hand in two ways: (i) limits the working environment of the users to a small space and (ii) often forces the users to hold their hands in a certain orientation (e.g. requires them to always show the dorsal part of the hand to the camera). Passive optical tracking should be accompanied by image processing, therefore requires a vast amount of computation. To decrease the complexity, most systems introduce certain restrictions in terms of the background and illumination of the working environment. In addition, passive optical tracking requires a vast amount of computation.

The assumption is that a proactive design support environment must encompass all instruments that are needed to detect the postural changes of the human body, by means of the designers express their intents and execute the shape construction operations. In the case of conceptualization of freeform shapes in space, the free movement of the hands is important. I learnt that detection of some carefully selected landmarks of the hands is sufficient either for designing surfaces or recognizing hand motions. What actually is needed is a technique, which is able to (i) detect motions from a distance, (ii) measure three-dimensional coordinates of the selected landmarks, and (iii) perform real-time detection.

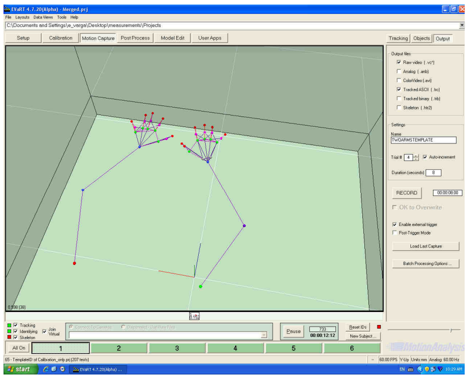
For these functions, active optical tracking with wireless markers and multiple cameras seemed to be the best available technologies. However, I believe that future technologies will enable non-mounted instrumentation, which means, that there will be no sensors or signal generators attached or placed on human body parts. The applied passive optical tracking method is discussed in the next section with a view to hand motion interpretation.

5.2 Passive optical tracking of hand motions

In order to be able to interpret the instructions of the designer, first the hand motions have to be detected. For this purpose, a passive optical tracking environment was used. In this environment, a camera system using infrared light measures the position of so-called retro-reflective markers, which are attached to specific landmarks of the designer's hand. The tracking system enables the free movement of the user, since there are no cables to connect the user to the detection device or to the computer. Previous approaches put several limitations on the hand postures and motions due to the insufficient technical capabilities of the tracking system. These limitations forced the user to use simple, but - from a shape modeling point of view - unnatural hand postures and motions. The six camera based system (Figure 5.1 a) enables the recognition of complex two-handed postures and motions, which were specifically designed to be intuitive for the designer. The positional data measured by the tracking device served as the input for hand motion interpretation and for geometry generation.

The camera system used was purchased from the Motion Analysis Corporation (www.motionanalysis.com). The Hawk Digital System consists of Hawk Digital Cameras, the EagleHub, and EVa Real-Time (EVaRT) software that can capture complex motion with extreme accuracy. Real-time capabilities allow users to see capture results at the same instant as the subject is performing a specific task.

a.



b.

Figure 5.1 (a) The hand motion detection equipment, (b) The model of the arms in EVaRT

The Hawk Digital Camera has a 640 x 480 full resolution at up to 200 frames per second. The camera signal goes directly to the tracking computer via an Ethernet connection, and the signal processing is embedded in the camera. The EagleHub consists of a multi-port Ethernet switch (100 Mbps) and provides power for the cameras. A single Ethernet Cat 5 Cable is used for all signals and power between the camera and the EagleHub. EVaRT software provides users with a simple and powerful interface. A screenshot of the software can be seen in Figure 5.1 b. Under a single software environment users can set up, calibrate, capture motion in real-time, capture motion for post processing, edit and save data in several formats. The retro-reflective markers are available in different sizes from 4-25 mm. The Software Development Kit (SDK) helps to connect EVaRT with user developed applications and stream data in real-time through the network.

Calibration of the cameras has to be done in order to establish a relationship between real-world positions and the corresponding image coordinates from the camera view. When a target is visible in two or more camera views, there is sufficient information available to track the targets in the three-dimensional space. If rays from two cameras intersect in space at a specific time, they define the three-dimensional position of a target at that time. The calibration of a camera's view is dependent on the camera lens focal length, and the position and orientation of the camera with respect to an arbitrary reference frame called the object reference frame. Therefore, a change of any sort, which alters the relationship between the object coordinates and image coordinates, must be followed by a new calibration.

After the selection of a reference frame, a number of calibration markers with known locations have to be provided for control purposes. For precise and accurate calibration, a dynamic linearization technique is used. Motion Analysis offers a calibration square with four retro-reflective spheres. The relative positions of the spheres have been accurately measured, and the device is used for defining the XYZ axes. A 500mm wand (for large capture volumes) or a 150mm wand (for small capture volumes) is then used for establishing camera linearization parameters. The wand is waved around throughout the capture volume by somebody. Wand calibration ensures that a direct measurement of an object of known size has been made by all cameras throughout the entire capture volume. The calibration process locates the exact positions of the cameras and accurately measures the camera lens focal length. Recalibration takes moments as compared to the cumbersome, time consuming grid technique used by other motion capture systems. In EVaRT, there is a model editing tool to build and modify model parameters for motion capture. Firstly, the marker set is designed and names are assigned to the markers, and then links between the markers are defined. A template specific to the markers in use is created as well. A template tells the software what the minimum and maximum distances are that exist between markers of a relatively fixed relation. To track multiple objects, their marker set has to show asymmetry, to be able to identify them by the software.

5.3 Main features influencing hand motion interpretation

A collaborating team of designers is supposed to use the hand motion based system to conceptualize shapes of various products. The designer uses the words of a predefined HML to externalize and describe either shape elements (surface patches) or indicate intended shape construction and manipulation operations.

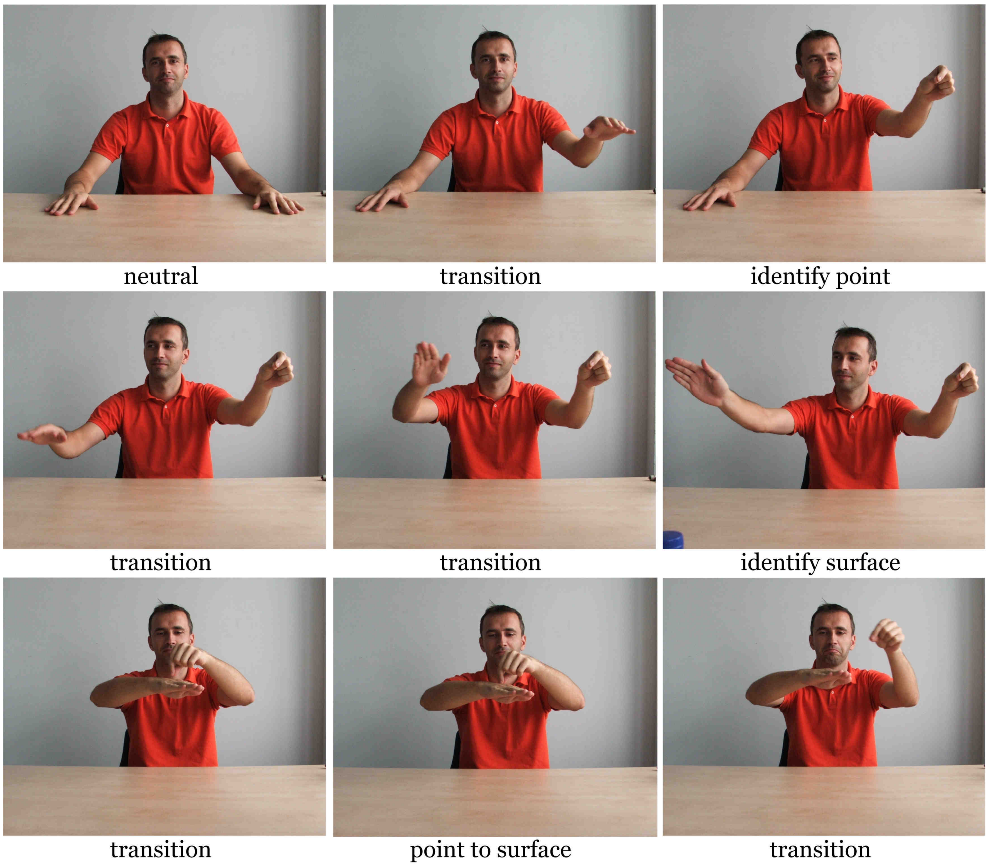


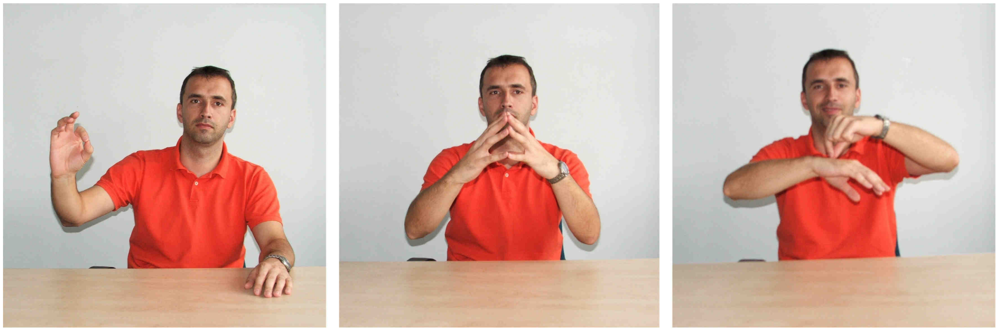
Figure 5.2 Formation of a sequence of signs

To support shape conceptualization, the HML contains geometric, identification, connectivity, positioning, scaling and assembling words. The HML (i) uses descriptive hand motions rather than symbolic hand postures, (ii) contains one-handed motions, two-handed motions, when the two hands move differently to form a manipulation command and double-handed motions, when the two hands are jointly used to generate an object. In addition, the HML is (iii) symmetric, that is, hand motions can be performed with either left or right hand. The output of the HML interpretation can be a (i) surface generated by motion trajectories of the hand, which is either a freeform or a regular shape; (ii) a manipulation command, which modifies the existing shapes, such as positioning or scaling; or

(iii) a procedural command, which facilitates the process of shape conceptualization, such as start, end, share, obtain, undo, redo.

An HML word has two states: a transition state and a steady state (Figure 5.2). In the transition state the postures of the hands are continuously changing. The steady state is the useful part of an HML word, without postural changes. Therefore, the transition state refers to a transition from one HML word to another, and the steady state has meaning indicating a modeling command.

The presence of the transition state and steady state of HML words imply that there are two actions have to be done: segmentation and recognition. The segmentation process is needed in order to find the beginning and end of an HML word, that is, to detach the meaningful part of a word represented by the steady state. The recognition process is needed in case of the steady state in order to recognize the HML word which represents a modeling command.



One-handed sign (identify point) | Two-handed sign (assemble) | Double-handed sign (point-to-surface)

Figure 5.3 Example for one-handed, double-handed and two-handed words

The HML consists of one-handed, double-handed and two-handed words (Figure 5.3). As its name implies, one-handed words are formed by a certain posture of one hand in the steady state of a sign. Double-handed words are actually the combinations of one-handed words. For instance, identification of a point and identification of a surface are one-handed words, however, when the intent of the designer is to connect this point and surface, it needs to be expressed by a double-handed word, which is in fact the combination of the mentioned two one-handed words. In the process of shape conceptualization double-handed words are typically used after a sequence of one-handed words. In case of two-handed words, the two hands are simultaneously moving and taking up the same posture on the trajectory of motion.

The above categorization of the HML words influences the recognition process in several ways. *The process of recognition has to be able to detect and process one-handed words as well as double-handed and two-handed words*. Because one-handed words can be generated by both hands, the posture recognition process should be applied to the left and the right hand individually. If both hands are moving on similar or different trajectories, the posture of the left hand and the right hand are separately recognized. However, more complex situations are to be encountered. The joint motion of the two hands may have three alternative interpretations; the words to be recognized can be one two-handed, one double-handed or two one-handed words. It is possible to take the advantage of the aforementioned features – such as the sameness of postures in the case of two-handed words, the sequence of one-handed and double-handed words - in the process of recognition. These features can be utilized to define a set of rules, which covers all the possible combinations of one-handed postures.

5.4 Algorithms for hand motion trajectory segmentation

The hand motion of the designer is continuously tracked and his two-handed posture is defined in each frame of motion. As described in Section 3.4, the basis for segmentation is the comparison of these two-handed postures. Basically, when the two-handed postures were the same in the last *n* frame of motion, it is recognized that an HML word has started.

Algorithm 5.1 Algorithm for trajectory segmentation

```
FindSegment(fno, n, m)
  ComparePostures(fno,n)
  if hasCommandBegun=NO
    then hasCommandBegun=YES
         DefineHMLWord()
  else i=CountDifferentFrames(fno)
       if i>m then hasCommandBegun=NO
```

The processing algorithm can be seen in Algorithm 5.1, which shows that the algorithm needs three inputs, the identifier of the current frame (*fno*), the number of the frames in which the two-handed postures has to be the same (*n*) and the number of frames in which it is allowed for the postures to be different (*m*). The m value prevents that the algorithm defines the end of the HML word due to momentary measurement errors. The algorithm assumes that the identified two-handed postures are stored to enable their comparison. Algorithm 5.1 shows once a command has started, the corresponding HML word is defined and the algorithm is set by a variable (*hasCommandBegun*) that from this point on it has to find the end of the command.

5.5 Algorithms for hand motion recognition

5.5.1 Process flow of hand motion recognition

First of all, I would like to clarify that I use two similar expressions throughout this section, namely posture recognition and HML word recognition. There is a significant difference between these two expressions. HML words are formed by a sequence of postures. However, recognition of the HML words requires further processing of sequence of postures. In short, the posture recognition is only the first step in HML word recognition.

The exploration of HML features in the previous section was made in order to be able to select a recognition method, which fits the HML the best. Since the postures do not change in the steady state of the HML words, I focused on available fast and sound posture recognition methods which can be adapted. See5 software, developed upon a decision tree based classifier method created by Ross Quinlan, was selected (http://www.rulequest.com/see5-info.html), since it enables the classification of postures based on their descriptive parameters. However, the recognition of two-handed postures requires further processing.

The process flow of two-handed posture recognition can be seen in Figure 5.4. *Personalization* is necessary to be done because of the differences in the hand sizes of people, which influences the recognition and plays an important role in the surface generation process as well. The personalization process has to be performed before the shape conceptualization process, by showing some certain postures to obtain the features which will be used later in the posture recognition. Markers are placed on specific parts of the hands, which are then measured by the camera system to obtain their three-dimensional coordinates. The personalization process is done for both hands separately to measure one-handed features. Features describing two-handed postures are also measured. Based on the obtained values, intervals are defined which are then assigned to the parameters describing the different postures.

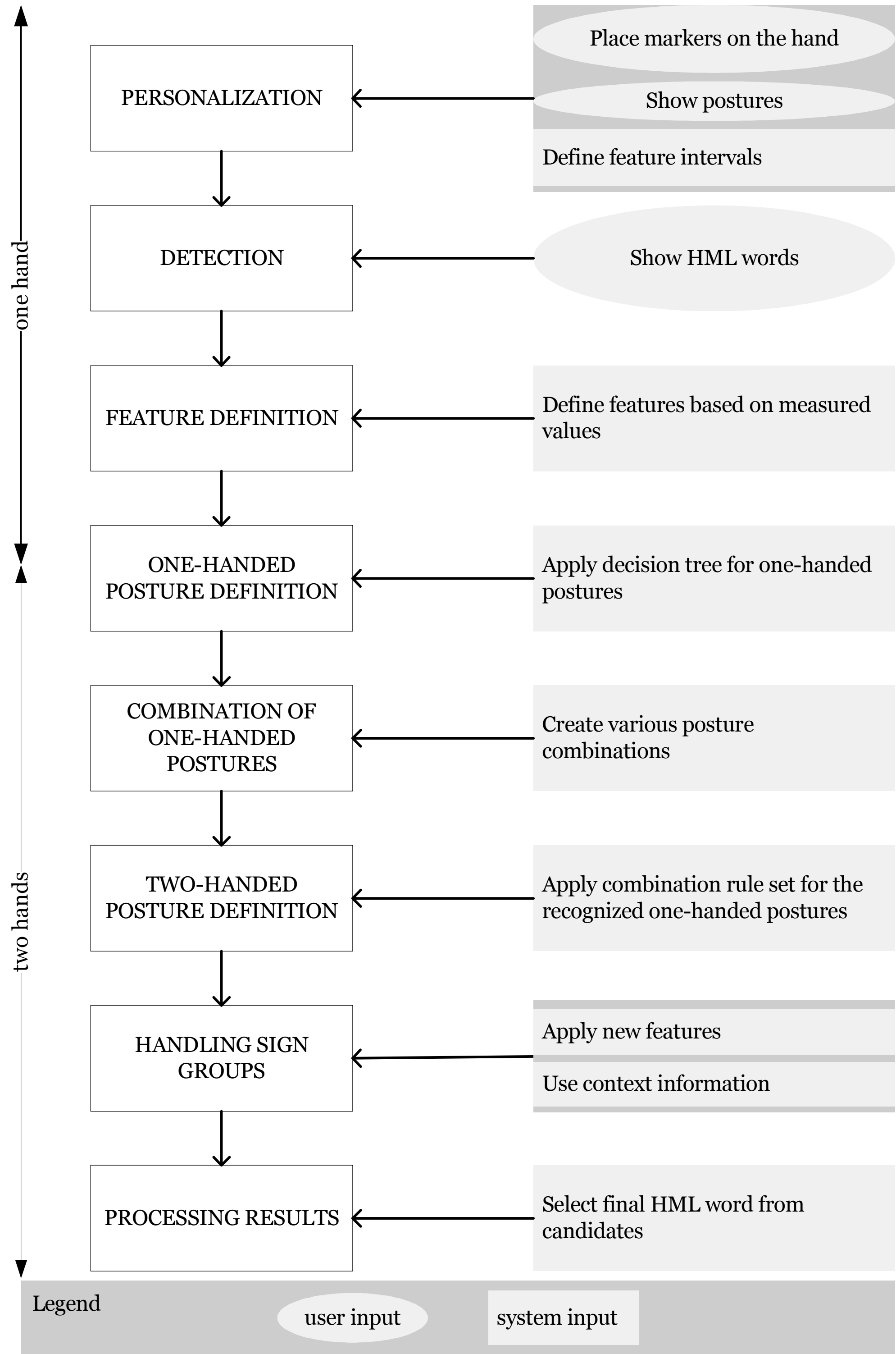


Figure 5.4 Process flow of HML word recognition

After the personalization, the designer can start with performing HML words for shape conceptualization. At one time only one designer can operate on the model and the others can ask for permission to continue with it. This way a conceptual design of a product can be generated by collaboration of a group of designers. During the *detection*, three-dimensional positions of the markers are measured repeatedly, according to the frame rate of the measuring device.

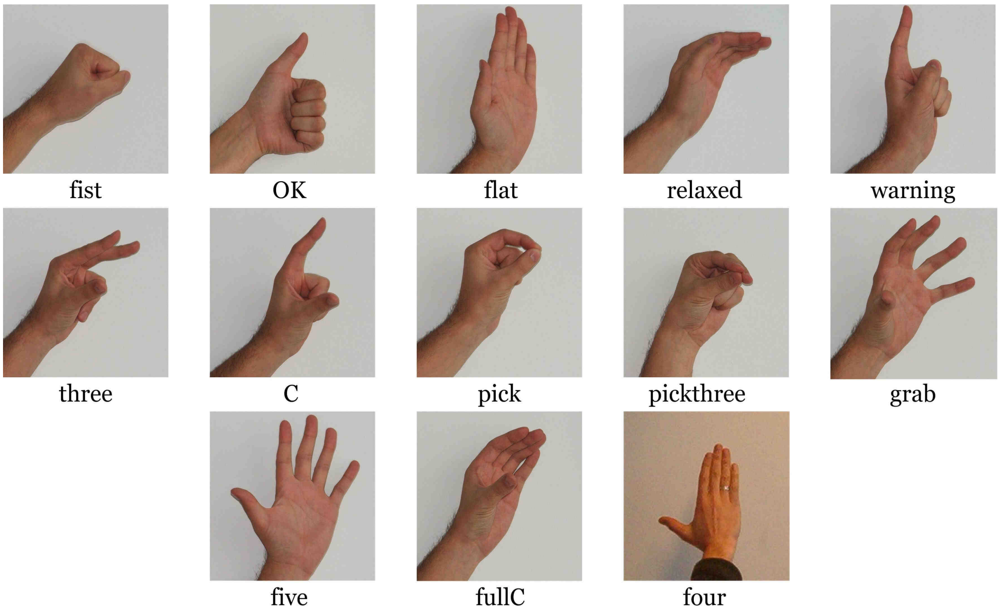


Figure 5.5 One-handed postures

The *feature definition* has to be carried out by calculating the distances of markers and by comparing them with the intervals defined in the personalization process. The defined feature values provide the bases for *posture recognition*, which is performed by the previously mentioned See5 software. Posture recognition is applied for both the left hand and the right hand individually. The result of the recognition process is a posture name for the left hand and a posture name for the right hand.

Combination of one-handed postures makes it possible to recognize two-handed postures. In most cases the one-handed posture is recognized unambiguously, but there are some situations, when the postures cannot be recognized correctly. In these cases two postures cannot be differentiated, therefore the result of the decision tree classifier is a posture group containing two postures. If it is the case for only one hand, two posture combinations are generated, and if both one-handed postures are ambiguous, four posture combinations are created. Meaningless posture combinations are immediately excluded from further investigation.

Posture combinations are the input for a set of combination rules. This set contains the meaningful one-handed posture combinations and assign the two-handed posture describing the steady state of an HML word. The result of the set of rules is either the final recognized sign or a *sign group*. Sign groups contain signs which are not possible to be recognized only based on the posture of the left hand and the posture of the right hand. In these special cases, additional information is obtained about the relations of the two hands or about the design context. Each sign group is handled individually and only the corresponding additional information is obtained, which is different for each sign group. Finally the *results are processed* by selecting a two-handed posture from the posture candidates.

5.5.2 Recognition of postures

I analyzed all possible postures and looked for those features which distinguish the postures the most. Towards a formal representation of the features a minimal set of descriptive parameters have been assigned. Intra-hand features describe one-handed, while inter-hand features two-handed postures.

Feature values:

Distance of thumb and index tip:	{ close | far }
Distance of thumb and index finger:	{ close | far }
Distance of index and middle finger:	{ close | far }
State of index finger:	{ MF/E | TF }
State of middle finger:	{ MF/E | TF }
State of ring finger:	{ MF/E | TF }

Predicted postures:

{ fist|OK|flat/relaxed|warning|three|C|pick|pickthree|grab/five|fullC/four|no-meaning }

Figure 5.6 Possible feature values and predicted postures

These features are actually the relations of specific landmarks of the arm and hand, more explicitly; features are distances of landmarks. At first, a set of features was collected intuitively by analyzing photos of the HML words. Then each of them was measured for all postures and those were kept for further investigation that showed significant differences in their values for the different postures. Finally, redundant information was eliminated and a minimal feature set was defined.

After a careful analysis of the possible descriptive features, I selected six of them, which describe and distinguish each of the possible postures shown in Figure 5.5.

These features are: (i) distance of the thumb and the index fingertip, (ii) distance of the thumb and the index finger, (iii) distance of the index and the middle finger, (iv) state of the index finger, (v) state of the middle finger and (vi) state of the ring finger. The distance features have two possible qualitative values, *close* and *far*, while the finger state features can be of *totally-flexed* or *moderately-flexed/extended*.

Algorithm 5.2 Feature assignment

Define_Left_Hand_Features(D_{LP})
    if ($(d_{L1}/d_{LP1}) < 0.17$) then f_{L1} = close else f_{L1} = far
    if ($(d_{L2}/d_{LP2}) < 0.6$) then f_{L2} = close else f_{L2} = far
    if ($(d_{L3}/d_{LP3}) < 0.2$) then f_{L3} = close else f_{L3} = far
    if ($(d_{L4}/d_{LP4}) < 0.8$) then f_{L4} = TF else f_{L4} = MF/E
    if ($(d_{L5}/d_{LP5}) < 0.8$) then f_{L5} = TF else f_{L5} = MF/E
    if ($(d_{L6}/d_{LP6}) < 0.8$) then f_{L6} = TF else f_{L6} = MF/E
Return F_L

Although in principle it would be possible to distinguish the moderately-flexed and the extended finger states in the decision tree, I decided not to do it because of the slight difference in their values. This difference may not be detected by the measuring device with sufficient precision. Therefore there are some postures, namely *flat/relaxed*, *grab/five* and *fullC/four* are grouped together in this phase of recognition. Actually, according to this method, 11 postures are built into the decision tree (Figure 5.6). The decision tree contains a posture group called *no-meaning* for theoretically possible, but in the HML dictionary meaningless postures or impossible feature value combinations. Other postures besides this set are not meaningful in the steady state of the HML words.

In the further part of this thesis the features are named as follows: (i) the distance of thumb and index tip as DTIT, (ii) the distance of thumb and index finger as DTIF, (iii) the distance of index and middle finger as DIMF, (iv) the state of index finger as SIF, (v) the state of middle finger as SMF, and the state of ring finger as SRF. Since three-dimensional positional data come from the measuring device as input to this system, they have to be transformed into the previously described feature values. As the hands show significant differences in sizes, a personalization step needs to be performed for each user.

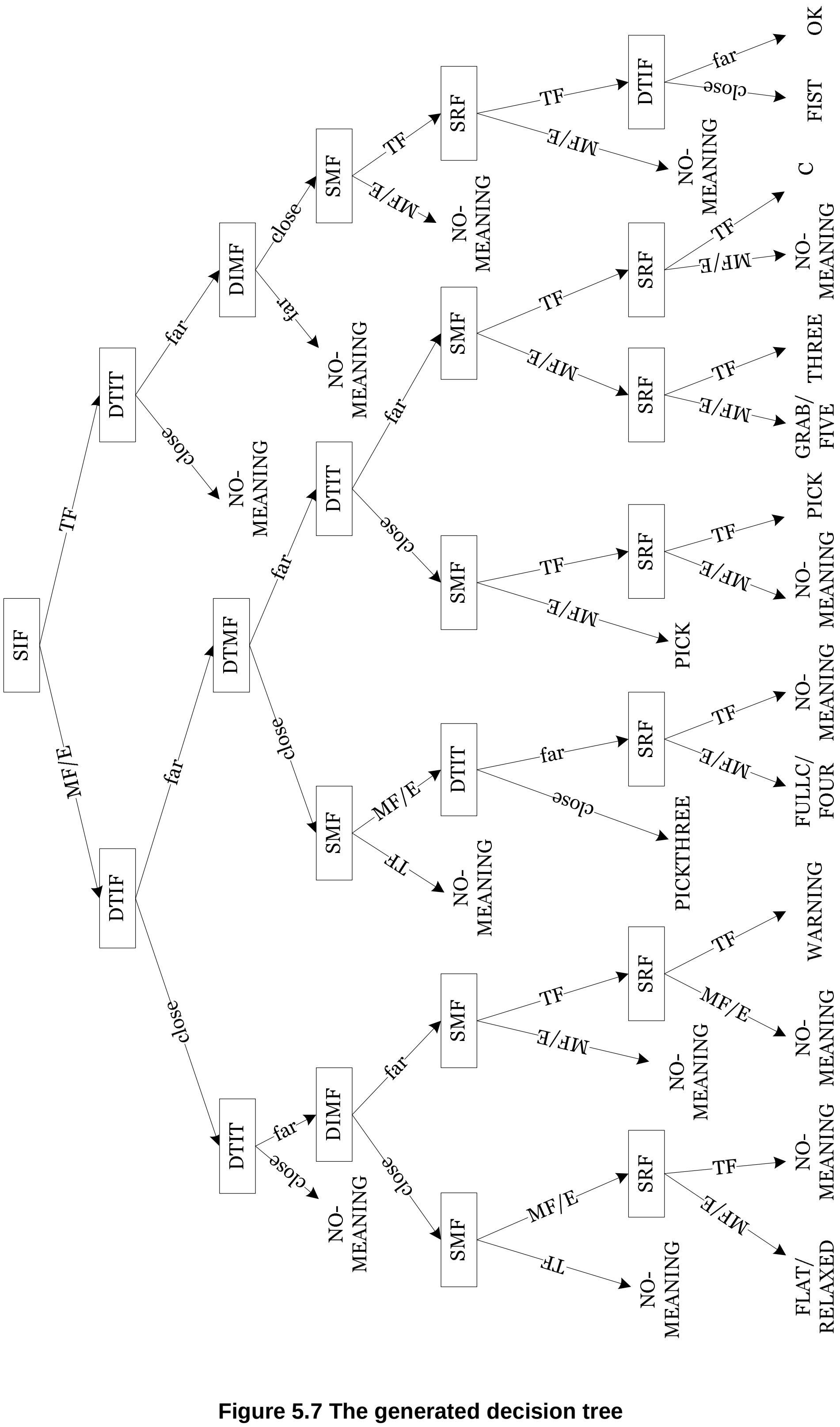


Figure 5.7 The generated decision tree

Five postures were selected to be shown to the system; these are the *pickthree*, *turn*, *flat*, *warning* and *fist* postures. From these postures, the most relevant feature values can be obtained. The *close* value of the DTIT feature is retrieved from the *pickthree* posture, while the *far* value from the *turn* posture. Other values come from the followings: DTIF *close* – *flat*, far – *pickthree*, DIMF *close* – *pickthree*, *far* – *warning*. The SIF, SMF and SRF *TF* feature values are all retrieved from the *fist*, while the *MF/E* from the *turn* posture.

close	far	close	MF/E	MF/E	TF	pickthree
close	far	close	MF/E	MF/E	MF/E	pickthree
close	far	far	MF/E	TF	TF	pick
close	far	far	MF/E	MF/E	TF	pick
close	far	far	MF/E	MF/E	MF/E	pick
far	close	close	TF	TF	TF	fist
far	close	close	MF/E	MF/E	MF/E	flat/relaxed
far	close	far	MF/E	TF	TF	warning
far	far	close	TF	TF	TF	OK
far	far	close	MF/E	MF/E	MF/E	fullC/four
far	far	far	MF/E	TF	TF	C
far	far	far	MF/E	MF/E	TF	three
far	far	far	MF/E	MF/E	MF/E	grab/five

Figure 5.8 Meaningful feature combinations

The upper limit of the feature interval is always the higher measured value, either *far* or *MF/E*. Based on several measurements of the postures, the border of the smaller and the higher value is defined according to the following percents, applied for Algorithm 5.2 – higher value interval: 15% for the DTIT, 60% for the DTIF, 20 % for the DIMF and 80% for the SIF, SMF and SRF features. At this moment these are sharp borders, so if the measured value is below it then it gets a *close* value for the DTIT, DTIF and DIMF features and a *TF* value for the SIF, SMF and SRF features. If it is above this border it gets a *far* value for the DTIT, DTIF and DIMF features and an *MF/E* value for the SIF, SMF and SRF features.

The posture recognition is performed by See5, which requires two files. One describes the attributes and the classes. In this case the attributes are the six features and the classes mean the 11 postures (Figure 5.6). For further information on formatting this file read (http://www.rulequest.com/see5-win.html). The second file provides information on the training cases from which See5 extracts patterns. All 64 feature value combinations and the corresponding classes were showed to the decision tree as training cases. The meaningful feature combinations can be seen in Figure 5.8.

After four training, the classifier reached one hundred percent prediction accuracy. Figure 5.9 shows the confusion matrix after the first, second, third and the fourth training. For instance, Figure 5.9 b shows the evaluation of the decision tree after the second training. The *Size* of the tree is its number of leaves and the column headed *Errors* shows the number and percentage of cases misclassified. The tree, with 11 leaves, misclassifies 10 of the given cases, and shows an error rate of 7.8%. The performance on the training cases is further analyzed in the *confusion matrix* that pinpoints the kinds of errors made. In this example, the decision tree misclassifies two of the fist, OK, flat/relaxed and warning cases as no-meaning and two of the no-meaning cases as pick cases. The graphical representation of the decision tree generated by See5 can be seen in Figure 5.7.

5.5.3 Recognition of Hand Motion Language words

The decision tree classifier is applied for both the left and the right hand individually. In the next step, these results are combined to be able to recognize the two-handed postures. If the posture was not possible to be recognized in one or both of the cases, then combinations are generated to be able to select the right one. For example, if *flat/relaxed* were recognized for the left hand, and *warning* for the right, then two combinations are generated for further investigation, namely, *flat-warning* and *relaxed-warning*. From this it easily turns out that only the *flat-warning* combination is meaningful. The left and right hand postures are combined with the help of a set of rules, which contains the meaningful combinations (Table 5.1). Actually, the algorithm defined for two-handed posture recognition is based on these combinations represented by *if-then* structures.

Table 5.1 shows all the posture combinations, which are meaningful, for all other combinations the system gives a *no-meaning* result. The posture group called *nothing* refers to cases when no measurement data comes from the device, which means that the hand is hidden under the table or the hand is out of the range of the measuring device.

```
Evaluation on training data (64 cases):

        Decision Tree
      ----------------
      Size      Errors

         1   13(20.3%)   <<
```

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	<-classified as
										1		(a): class fist
										1		(b): class OK
										1		(c): class flat/relaxed
										1		(d): class warning
										1		(e): class three
										1		(f): class C
										3		(g): class pick
										2		(h): class pickthree
										1		(i): class grab/five
										1		(j): class fullC/four
										51		(k): class no-meaning
												(l): class nothing

a.

```
Evaluation on training data (128 cases):

        Decision Tree
      ----------------
      Size      Errors

        11   10( 7.8%)   <<
```

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	<-classified as
										2		(a): class fist
										2		(b): class OK
										2		(c): class flat/relaxed
										2		(d): class warning
				2								(e): class three
					2							(f): class C
						6						(g): class pick
							4					(h): class pickthree
								2				(i): class grab/five
									2			(j): class fullC/four
						2				100		(k): class no-meaning
												(l): class nothing

b.

```
Evaluation on training data (192 cases):

        Decision Tree
      ----------------
      Size      Errors

        18    6( 3.1%)   <<
```

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	<-classified as
3												(a): class fist
	3											(b): class OK
										3		(c): class flat/relaxed
										3		(d): class warning
				3								(e): class three
					3							(f): class C
						9						(g): class pick
							6					(h): class pickthree
								3				(i): class grab/five
									3			(j): class fullC/four
										153		(k): class no-meaning
												(l): class nothing

c.

```
Evaluation on training data (256 cases):

        Decision Tree
      ----------------
      Size      Errors

        24    0( 0.0%)   <<
```

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	<-classified as
4												(a): class fist
	4											(b): class OK
		4										(c): class flat/relaxed
			4									(d): class warning
				4								(e): class three
					4							(f): class C
						12						(g): class pick
							8					(h): class pickthree
								4				(i): class grab/five
									4			(j): class fullC/four
										204		(k): class no-meaning
												(l): class nothing

d.

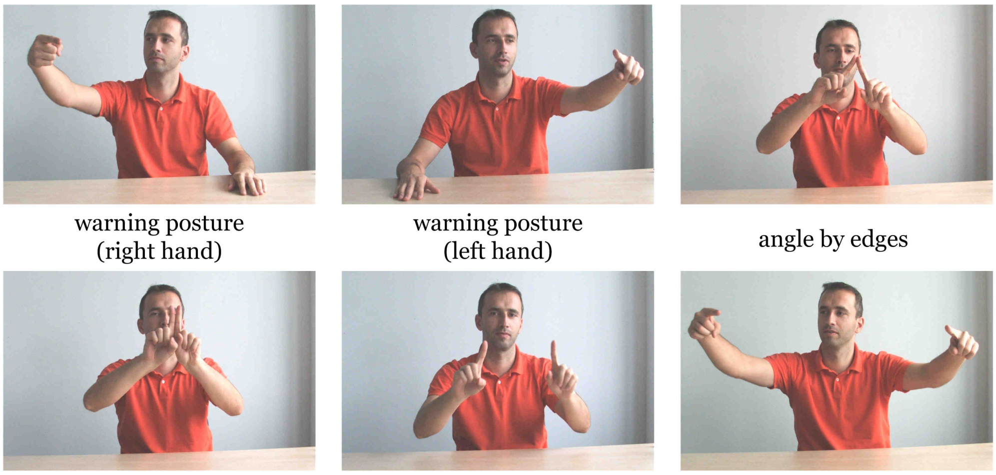
Figure 5.9 Confusion matrix after the first (a), second (b), third (c) and fourth (d) training

Table 5.1 Meaningful posture combinations

Posture1	Posture2	Command
C	nothing	identify curve
fist	nothing	{ start | end }
flat	nothing	{ identify plane | put aside | bring in | cut through | cut out }
pick	nothing	identify point
three	nothing	identify-surface
four	nothing	turn
warning	nothing	identify line
fullC	fullC	{ compose | decompose }
C	C	{ size by curves | identify two curves }
five	five	{ assemble | disassemble }
flat	flat	{ identify object | angle by surfaces | surface to surface | distance by surfaces | identify two planes }
flat	pick	(point to surface | identify plane and point }
flat	warning	{ curve to surface | identify plane and line }
grab	grab	{ construct | deconstruct }
OK	OK	{ stop | resume }
pick	warning	{ point to curve | identify point and line }
pick	pick	{ point to point | distance by points | size by points | identify two points }
pickthree	pickthree	zoom
relaxed	relaxed	{ share | obtain | undo | redo }
three	three	{ size by surfaces | identify two surfaces }
warning	warning	{ angle by edges | curve to curve | distance by curves | identify two lines }
C	flat	identify curve and plane
C	pick	identify curve and point
C	three	identify curve and surface
C	four	identify curve and turn
C	warning	identify curve and line
flat	three	identify plane and surface
flat	four	identify plane and turn
pick	four	identify point and turn
three	pick	identify point and surface
three	four	identify surface and turn
three	warning	identify surface and line
four	four	two turns
four	warning	identify line and turn

However, for the recognition of some two-handed postures more information is needed. After the creation of the set of rules, which combines the one-handed postures, I realized that some two-handed postures cannot be distinguished only relying on this information. I collected these postures into posture groups, which can be handled individually. For each group, extra information was generated, in forms of inter-hand features and in forms of design context information (e.g. which objects are selected or what the last command was). It can be seen in Table 5.1 that fifteen posture combinations create a group of possible commands. These are handled separately with the help of situation-based reasoning as described above.

In Figure 5.10 an example can be seen for a sign group, which is created by two *warning* postures. I know from Table 5.1 that the corresponding modeling command can be *angle by edges*, *curve to curve*, *distance by curves* or *identify two lines*.



warning posture (right hand) | warning posture (left hand) | angle by edges

curve to curve | distance by curves | identify two lines

Figure 5.10 Example for sign group

For this particular sign group two additional features are computed, the distance of the fingertip of the left hand and of the right hand (DIT) and the distance of the MCP joint of the left hand and of the right hand (DIM). (For the understanding of notions refer back to Figure 3.5.) The right moment for obtaining the extra features is the end of the sign, because one command can be a transformation phase of an other one, e.g. the *distance by curve* command can end in a *curve to curve* or *angle by edges* command.

Table 5.2 Meaningful combination of additional features

Left-Attached	Right-Attached	DIT	DIM	Command
Curve	Curve	Close	Close	Curve to curve
Curve	Curve	Close	Far	Angle by curves
None	None	Far	Far	Identify two lines
curve	curve	far	far	Distance by curves

As earlier, these additional feature values can be either *far* or *close*. Two other feature values are also examined, namely, whether an element of the designed model is attached to the left and to the right hand or not. These elements are attached to the hand, when they are selected, and have five possible values: *none*, *point*, *curve*, *surface* and *object*. Table 5.2 shows the meaningful combinations of these values and the corresponding commands.

5.6 Algorithms for surface generation

The theory of surface generation was presented in Section 3.7 and this Section describes the related algorithms. Algorithm 5.3 shows that the surface generation process needs the measured coordinates of the wrist, the MCP joint and the tip of the middle finger, the measured coordinates of the MCP joint of the index finger and the identifier of the current frame.

Algorithm 5.3 Algorithm for surface generation

```
GenerateSurface(wrist, middleMCP, middleTIP, indexMCP, frame)
  for(each frame of motion)
  {
    (middlePIP, middleDIP) = ApplyHandModel(wrist, middleMCP, middleTIP, indexMCP)
    extra_points = Interpolate(wrist, middleMCP, middlePIP, middleDIP, middleTIP)
    surface = AddPointsToSurface(wirst, middleMCP, middlePIP, middleDIP, middleTIP,
                  extra_points)
  }
  surfaceID = TriangulateSurface(surface)
  return surfaceID
```

For each frame of motion first the hand model (described in Section 3.6) is applied to calculate the coordinates of the PIP and DIP joints of the middle finger. Then extra points are generated based on the measured and the calculated points of the middle finger applying an interpolation method. Based on the requirements for the quality of surface, it can be a simple linear interpolation or a NURBS interpolation. I applied linear interpolation, because the speed of processing is more important in the early stages of design than the quality of the surface.

The points are collected throughout the surface generation process, and when the end of surface generation is indicated by the designer the surface is triangulated and it gets an identifier for manipulation purposes.

5.7 Algorithms for surface manipulation

After the recognition of the HML words the corresponding geometric information has to be obtained. The HML words were analyzed individually in order to identify the corresponding geometric information and its processing. It is assumed that the geometric modeler is a point set based system; therefore entities like surfaces and objects are modeled with points as well.

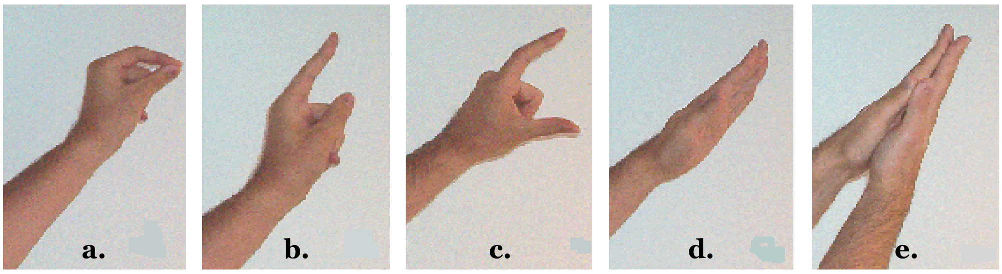


Figure 5.11 Identification words (a) Identify point (b) Identify line (c) Identify curve (d) Identify surface (e) Identify object

A basic group of the HML words is the identification words, comprising identifying point, identifying line, identifying curve, identifying surface and identifying object (Figure 5.11). As examples, algorithms are given in this section for processing the identification HML words. Algorithms for other HML words are generated in a similar way according to their meaning.

The processing of the *identify point* HML word requires finding the closest point of the product model to the P_9 computed point in Figure 5.12 a. This point is defined as the middle point of the P_3P_5 line. The effect of executing this command is that the closest point is highlighted in the model (Figure 5.13 a). As the user moves his hand, the calculation has to be done continuously, to be able to find the closest point. However, this process can be accelerated by skipping certain frames of motion in a way that the calculation is done only for every n^{th} frame. This appears in the second line of Algorithm 5.4 a, which shows that the calculation is completed for certain frames only, which is set by the *frequency* variable. This can be set according to the frame rate of the detection device.

Algorithm 5.4 Find closest point to P_1 point,

```
Find_Closest_Point_in_Selected_Frames( frame, P1)
  if ( ( frame % frequency) == 0 )
  P = Find_Closest_Point(P1)
Return P

Function Find_Closest_Point(P1)
  s = Generate_Sphere(P1, radius)
  b = Find_Intersecting_Bounding_Boxes(s)
  if ( b != NULL )
    b1 = Find_Container_Subbox(b, P1)
    Pfar = Find_Farthest_Point_in_Subbox(b1, P1);
    R = Length(P1Pfar)
    bc = Find_Close_Subboxes_with_BSP(b, P1, R)
    PC = Find_Closest_Point_in_Subboxes(bc)
Return PC
```

The algorithm specifies the steps of finding the closest point to another point in three dimensions. Because of the uncertainty of the hand motions, and because the markers I use during the detection are not points, first I build a sphere around the P_9 point with a certain *radius*. Then those bounding boxes of point sets are selected, which intersect with the generated sphere. If such a box exists, then Binary Space Partitioning (BSP) is applied in order to find the sub-box, which contains the point.

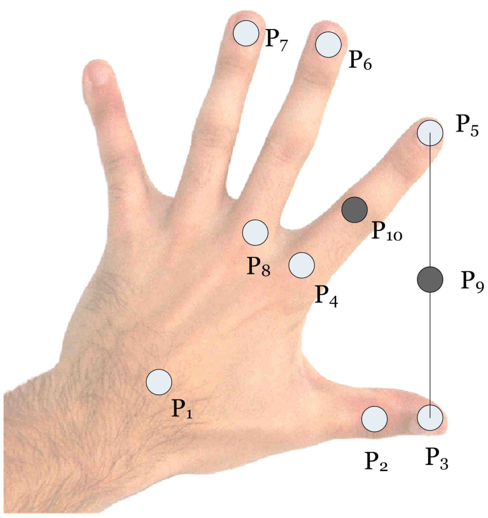
Algorithm 5.5 Find closest line to P_1 point

```
Find_Closest_Line(P1)
  s = Generate_Sphere(P1, radius)
  b = Find_Intersecting_Bounding_Boxes_of_Lines(P1)
  if ( b != NULL )
    lc = Find_Closest_Line_to_Point(l, P1)
Return lc

Function Find_Closest_Line_to_Point(l, P)
  For each line in l
    A = Get_First_Point(li)
    B = Get_Second_Point(li)
    a = Length(Cross_Product(AB, AP))
    di = a / Length(AB)
    d = (di < di-1 : di ? di-1)
Return d
```

BSP divides the bounding box, *b*, of the point set into sub-boxes containing a given number of points and then hierarchically structures the sub-boxes in a so-called BSP-tree. To find the closest point, P_C, to a given point, P_1, first the closest sub-box, b_1 is searched. Then the geometrically closest point, P_2, is identified in b_1. To investigate all possible solutions, a set of sub-boxes, b_c, is selected that is within the distance of P_2 and the farthest point from it in b_1. In the final step, the closest point, P_C, is identified which is in b_c. Using BSP reduces the computation time from $O(N)$ to $O(logN)$, where N is the number of points in the selected point set.

The frame selection method defined in Algorithm 5.4 a is applied for each algorithm presented in this section. Algorithm 5.4 b shows the process of finding the closest line to a given point. This algorithm is used for processing the *identify line* HML word. I look for the closest line to the P_{10} point shown in Figure 5.12 a. This point is denoted by P_1 in the algorithm. First, those bounding boxes of the lines in the product model are searched, which intersect with the sphere generated around the point P_1. Then for each selected line, the closest point of the line to the point P_1 is searched.



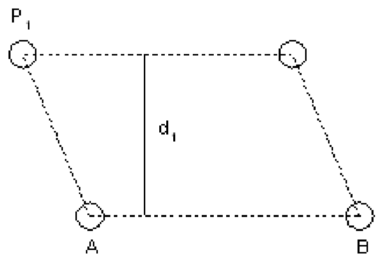


Figure 5.12 (a) Measured points on the left hand,(b) Calculation of closest point to a line

This method is based on a simple geometric solution. The *a* variable, which is calculated as the length of the cross product of the vectors $\overrightarrow{AB}$ and $\overrightarrow{AP_1}$, gives the area of the parallelogram formed by the vectors $\overrightarrow{AB}$ and $\overrightarrow{AP_1}$ (Figure 5.12 b). If the base of the parallelogram is AB, its area is AB*d_1, where d_1 is the distance what I look for. The minimum distance is then selected to be able to identify the closest line to the P_1 point.

Algorithm 5.6 Find closest curve to P_1 point

```
Find_Closest_Curve(P1)
        P = Find_Closest_Point(P1)
        c = Find_Containing_Curve(P)
Return c
```

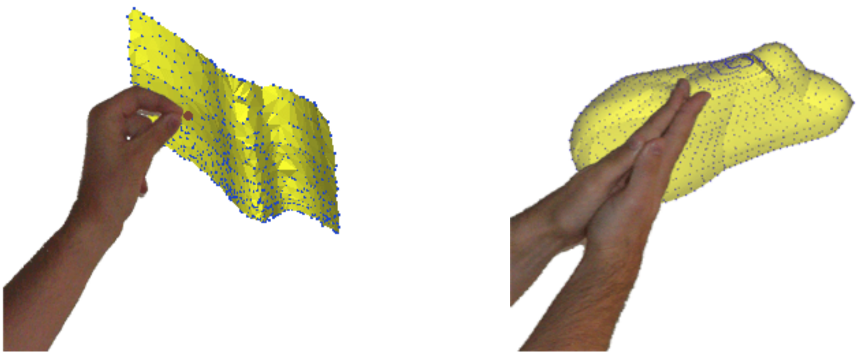
Algorithm 5.7 Find closest surface to P_1 point

```
Find_Closest_Surface(P1)
        P = Find_Closest_Point(P1)
        s = Find_Containing_Surface(P)
Return s
```

Algorithm 5.8 Find closest object to P_1 point

```
Find_Closest_Object(P1)
        P = Find_Closest_Point(P1)
        o = Find_Containing_Object(P)
Return o
```

Algorithm 5.6 a, b and c are used to find the closest point to the given entity, such as curve, surface or object. For algorithm Algorithm 5.6 a the input parameter is P_{10} and I look for the closest curve in the model to this point. For Algorithm 5.6 b the input is P_8 and the closest surface to this point is searched. In case of Algorithm 5.6 c the input parameter is the middle point of the line connecting the MCP joint of the middle finger of the left hand and the same point of the right hand, as it is defined in Appendix D, and the closest object to this point is selected (Figure 5.13 b).



a. b.

Figure 5.13 (a) Identify point, (b) Identify object

6 VALIDATION OF HAND MOTION BASED SHAPE CONCEPTUALIZATION IN APPLICATION

The previous chapter showed that a working system has been developed. It is said that the proof of the pudding is in the eating, so let's see how it works. Do people like using it? Does it provide shapes with a reasonable quality? What is the accuracy of shape manipulations that can be achieved? How the complexity of the model influences the complexity of the modeling process? These were the main questions to be answered.

Therefore, I evaluated the paradigm of hand motion-based interaction from human and technical aspects in order to gain information about its compliance with conceptual shape design. This chapter summarizes the findings of the experiments in two parts. The first part focuses on the human related aspects, and reports on a user study which was designed and conducted to study the usability of hand motions in conceptual design. The second part investigates the quality of shape definition by hand motions, and explains the conduct and results of the related experiments.

This first part of this chapter is based on the paper (Varga et al., 2007a). (Varga et al., 2007b) provided the basis for the second part, but contains much additions to it. I use the words "we" and "our" extensively throughout this chapter to acknowledge the help of my supervisors during the experiments.

6.1 Studying the usability of the hand motion interface

Our assumption was that shape modeling with HML is easier, more user-friendly, more intuitive and faster than with the conventional interface of CAD systems, especially for those product designers, who experienced difficulties with using CAD systems in the early phase of design. Therefore, a user study was designed and conducted to obtain information about the usability of the hand motion based interface. The challenge was to grasp the commonly used terms, such as user-friendliness and intuitiveness, and to convert them to analyzable and comparable quantitative and qualitative measures. In the context of usability our main research questions were: (i) What criteria can be defined for usability analysis in terms of interpreted hand motions?, (ii) What are the right methods for gathering user opinions about HML-based modeling?, (iii) Based on what characteristics can the participants of the study be sorted into comparable groups?, and (iv) How the results can be used to develop the forthcoming steps of our research. Regarding the usability of HML in conceptual shape design, compared to traditional CAD, we questioned: (i) Is HML based modeling is really faster than the keyboard-mouse based interface as we assumed?, (ii) Is the HML is more intuitive (easier to learn, to remember and to control)?, and (iii) Do the users like using the HML-based command?

6.1.1 An overview of usability definitions and measures

As a first step, we have conducted a literature review with the goal to study (i) how usability is defined and studied in connection with the widely used commercial and research software tools, (ii) what kind of qualitative and quantitative measures are used to evaluate usability in the context of the abovementioned cases, and (iii) what kind of methods exist to test usability in terms of human-computer interaction methods. Since our goal was to evaluate HML based modeling using the proof-of-concept implementation of the HML interpreter, we were curious how other researchers approached the problem of usability. Moreover, we were interested to know how much the well-known evaluation methods are relevant for our experimental software. Lastly, we wanted to learn the context in which usability has been interpreted.

Definitions of usability

There is no agreement on the definition of usability. We learnt from our literature review that some authors all agree that the importance of usability is getting more and more attention, but still there are confusions about the actual meaning and the measures related to this term. The ISO 9241-11 draft standard defines usability as the “extent to which a product can be used with effectiveness, efficiency and satisfaction in a specified context of use”. According to the ISO/IEC 9126 standard the term usability is the capability of a product to be used easily, and is related to the capability of the software product to be understood, learned, and used as well as to be attractive to the user, when used under specific conditions. Usability is further analyzed in this standard according to understandability, learnability, operability, attractiveness, and compliance. This standard also claims that the product attributes required from the viewpoint of usability depend on the characteristics of the user, task and environment. Therefore, usability is defined in this standard as a property of the overall system.

Measures of usability

(Bevan and Macleod, 1994) defined measures concerning the terms used in the definition of usability: effectiveness, efficiency, and satisfaction. The measures of effectiveness were defined by relating the goals of using the system to the accuracy and completeness with which these goals can be achieved. The measures of efficiency were determined by relating the level of effectiveness to the expenditure of resources. The resources may be mental efforts, physical effort, time, or financial cost. Finally, the measures of satisfaction describe the perceived usability and acceptability of the overall system by its users. According to (Ferre et al., 2001) usability is too abstract to be studied directly, and therefore they divided it into attributes like learnability (i.e. how easy it is to learn the functions of the system), efficiency (i.e. the number of tasks per unit of time that the user can perform using the system), user retention over time (reflecting how the users can work with the system after a period of non-usage), error rate (that addresses the number of errors the user makes while performing the task), and finally satisfaction (that shows the users' subjective impression of the system). A system's usability is not merely the sum of the values of these attributes. It is defined as reaching a certain level for each attribute.

By reviewing the related literature and legislations, Fitzpatrick and Higgins (1998) have identified a set of usability factors, which have direct impact on the end-user. These factors are suitability, installability, functionability, adaptability, ease-of-use, learnability, interoperability, reliability, safety, security, correctness, and efficiency. Several authors suggested to integrate usability testing into the software design and development processes. Answering the question of how to test usability becomes even more difficult when it comes to research software, which is usually a pilot, i.e. experimental software with limited functionality. In these cases, researchers select some important criteria, which help them prove their hypotheses (De Antonio et al., 2004).

Methods of data collection

Generally, when a system is tested, its performance is measured against predefined criteria. To test usability and to collect data for analysis, individual users are typically observed, who are completing specific tasks with the system (Avouris, 2001). The most widely accepted usability testing techniques are as follows. (a) Thinking Aloud Protocol: participants are asked to vocalize their thoughts, feelings and opinions while interacting with the software. (b) Co-discovery: a type of usability testing where a group of users perform a task together while being observed, simulating typical work processes. (c) Performance measurement: a type of test which determines hard, quantitative data, such as task completion time or CPU usage. (d) In-field studies: concern observation of the users performing their tasks in their usual environment of work. (e) Questionnaires and interview based protocols: used to ask direct questions from the users about the system. Though inquiry methods can be used to measure various usability attributes, their typical use is to measure user satisfaction. A known technique for measuring user satisfaction and for the assessment of user perceived software quality is the Software Usability Measurement Inventory (SUMI). The results of SUMI are analyzed according to five sub-scales, namely, affect, efficiency, helpfulness, control, and learnability.

In summary, there is no common agreement on the definitions, measures, and testing methods of usability. Although some definitions exist in widely accepted standards, they reflect different interpretations, and researchers use them to derive measures based on their own reasoning. On the other hand, these standardized definitions, common measures, and testing methods were designed to fulfill the needs of commercial software developers. Therefore, no directions are given to researchers working with less defined prototypes. Researchers have tried to make use of those measures and methods, which make sense in their own field of interest. Our approach is similar. We focus on two major issues regarding the usability of hand motion based modeling: (i) task completion time, and (ii) user satisfaction. While task completion time can be measured relatively simply, user satisfaction is more complicated to be measured. For measuring user satisfaction, we decided to use a questionnaire-based method. The questionnaire was constructed based on SUMI, as it is the most commonly used questionnaire for gathering users' opinion. Our questionnaire incorporates specific questions concerning hand motion based modeling. As it was suggested by several authors, we also considered usability as a system property, which covers all the hardware and software aspects, as well as the interaction method.

In the remained rest of this chapter, first we introduce our assumptions and hypotheses related to the usability of hand motion based modeling, and then we elaborate on the description of the testing environment, and on the design and the results of the user study.

6.1.2 Establishing evaluation criteria

We assumed that the HML based modeling gives a novel and intuitive platform for product designers, who think of computers as necessary equipment in their work, but actually they have had reluctance to use it. What we would like to prove is that HML based modeling performs better than traditional CAD in conceptual design tasks. It first of all means that products could be modeled faster. But we also believe that the use of this method supports creativity and increase the enthusiasm of designers. However, due to the novelty of hand motion based shape design, there was no guarantee that this assumption is correct. The purpose of our user study was to prove it, or at least, to gain more information about the abovementioned aspects.

Because we wanted to compare the hand motion based interaction paradigm with the conventional graphical interaction (CAD) paradigms on technology level, it seemed to be logical to select a commercial CAD software tool, which is widely used in the industry and at the academia as a basis for comparison. When we refer to the HML technology and CAD technology, we always mean it on a system level, which integrates the necessary hardware (input device, visualization device, etc.), the software (HML interpreter or shape modeler) and most importantly the interaction method itself (hand motions or keyboard- and mouse control). With regards to HML based modeling, our hypotheses were that (i) users can conceptualize shapes much faster with HML, (ii) HML is more intuitive for both novel and expert CAD users, (iii) HML is more attractive, which means that people are more enthusiastic to use it, (iv) HML is mainly attractive for those users, who has no experience with CAD, and, as an obvious disadvantage that (v) HML is more tiring physically, because of its motoric nature, but on the other hand it is less tiring (actually, more stimulating) mentally.

To be able to evaluate the abovementioned assumptions and to prove our hypotheses, several criteria were defined. To demonstrate the speed of our technology and method, we decided to measure the time the users spent with both systems, to compare them, and to figure out if there is any relevant difference between them. For analyzing the intuitiveness of the HML based modeling method, the criteria of understandability, learnability, and operability were introduced. To point out if people would like to use such a system, the criteria of satisfaction was defined. Finally, to learn more about the mental and physical involvement of people when using hand motions (i.e. how tired they become mentally and physically), criteria were set for cognitive load and physical comfort.

6.1.3 Design of the experiment

The user study was conducted with fifty participants, who were different in gender and age, and they either had experience with computers and CAD software, or not. The characteristics of the participants are shown in Table 6.1. The participants were asked to perform the same modeling task with the proof-of-concept hand motion modeling software and with a commercial CAD software. Video recording and questionnaires were used as data collection instruments. Each session was recorded on videotape, and the participants were asked to fill out two types of questionnaires. The pre-study questionnaire contained information about the participant, such as gender, age, educational background, level of general computer experience and level of experience with any CAD software. Based on these information, comparable groups were formed, such as (a) male vs. female participants, (b) beginner or intermediate level participants vs. participants with advanced level of general computer experience, (c) participants without any CAD experience vs. participants with CAD experience. Likewise, sub-groups of participants with CAD experience, participants with beginner or intermediate level vs. participants with advanced level of CAD experience were identified.

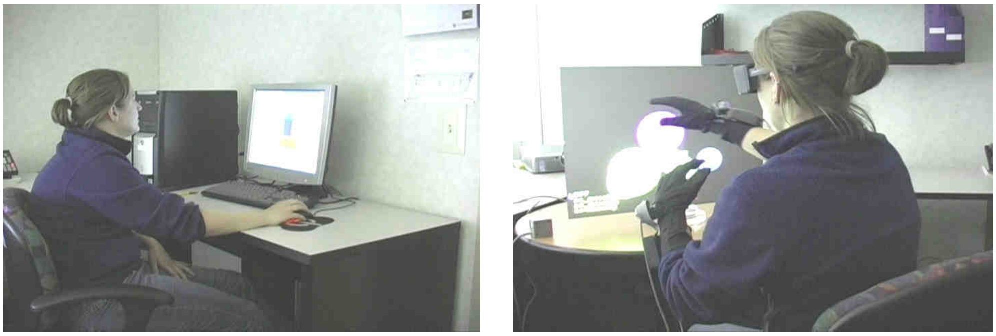
Table 6.1 Characteristics of user study participants

Gender	Age	General computer experience	Prior knowledge to CAD software
37 male	Min. 18 and max. 54	10 beginner or intermediate	36 yes
13 female	Average 27	40 advanced	14 no

The post-study questionnaire was related to the modeling technologies the participants used during the experiment, and inquired their experience and opinion. A separate post-study questionnaire was filled out for HML based modeling and for CAD modeling. The questionnaire contained twenty statements and used a forced choice Likert-scale with four answers: strongly agree, agree, disagree, and strongly disagree. The numbers in the table correspond to these four options of an answer. The higher value shows the better results. The full list of questions can be seen in Table 6.2. The first and the last question were general questions with regards to the environment and to the achieved result. The other questions were selected from the SUMI sample questionnaire with a view to the previously formed categories: understandability, learnability, satisfaction, operability, cognitive load, stimulation and physical comfort. Those questions, which referred to documentation and help, to the design and usage of graphical controls, and to the usage of keyboard and mouse were disregarded. These questions are meaningless in the case of HML based modeling. On the other hand, some questions were added to the questionnaire regarding physical comfort, because this topic was not covered by SUMI. Three questions were combined to form a category, e.g. the category learnability involves questions no. 5, 12 and 16. Exceptions are the category of cognitive load, which contains two questions, and the category of stimulation, which has only one question.

6.1.4 Testing environment

For the purposes of our tests, the 5DT dataglove was used to measure the flexion of the fingers. To measure the three-dimensional position of the hands, a Polhemus Patriot magnetic position tracker was attached to the back of each dataglove. Participants wore three-dimensional glasses to support navigation in the virtual environment (Figure 6.1 b).



a. **b.**

Figure 6.1 (a) Testing environment CAD, (b) testing environment HML

The HML interpreter was integrated into the VR Juggler environment. VR Juggler provides a virtual platform for virtual reality application development (Bierbaum et al., 2001). In the case of the CAD environment, a conventional desktop computer was available for studying interaction and two-dimensional monitor for providing visual feedback (Figure 6.1 a). For object modeling, SolidWorks, SolidEdge and Autodesk software was at the disposal. Participants could select one of the available software in the case of the CAD environment, which they were the most familiar with. By providing several options, our goal was to avoid misleading conclusions about time, because the experience with the software influences the time a participant spends with it.

Table 6.2 Evaluation of the post-study questionnaires

	Question	Mean Value CAD	Mean Value HML
1.	I liked using this environment very much.	2.69	3.02
2.	Working with this environment did not make me physically tired.	3.29	2.65
3.	I sometimes wondered if I was using the right command.	3.18	2.57
4.	Working with this environment is mentally stimulating.	2.76	3.12
5.	It took me too long to learn the controlling commands.	2.55	1.82
6.	It was easy to make the application do exactly what I wanted.	2.35	2.23
7.	I enjoyed my session with this application.	2.80	3.35
8.	There have been times in using this environment when I felt tense.	2.47	2.37
9.	I always knew what to do next.	2.39	2.92
10.	This application behaves in a way which cannot be understood.	2.55	2.76
11.	I would recommend this to friends or colleagues.	2.63	2.96
12.	It easy to forget how to do things with this application.	2.84	1.88
13.	When using the environment, I felt pain in my shoulder or neck.	1.69	1.94
14.	I would not like to use this application every day.	2.47	2.37
15.	There are too many steps required to get something work.	2.61	1.82
16.	Learning to operate this application is easy.	2.55	3.04
17.	Using this environment is frustrating.	2.49	2.45
18.	Tasks can be performed in a straightforward manner using this application.	2.55	2.88
19.	When using the environment, I felt pain in my hands or in my fingers.	1.53	1.90
20.	I think I have completed the given task well.	2.82	2.63

6.1.5 Conducting the experiment

The protocol of the user study was the following. First the participant filled out the pre-study questionnaire. Then (s)he was explained what was going to happen in the experiment. (S)he was asked then to flip a coin in order to randomly select the first software to be used. If it had turned out heads, the participant started with the HML based modeling system, if tails, with the CAD system. In each case it was explained shortly how to use the hand motion based modeling system.

If the participant did not have any experience with any of the available CAD systems, (s)he was explained how to perform the given task in SolidWorks. Each questionnaire was filled out directly after the session either with the HML based modeler or the CAD modeler. The task was to draw a hill-like surface, to build a tower out of three given objects and put the tower on the top of the hill. The task was designed in a way, that it contained all of the basic manipulation tasks which are used during modeling, such as positioning and rotating. For generating the hill, a freeform surface should have been generated. The participants were told that the accuracy of the model was not important, and it could be in any direction and orientation. They could create any kind of object which looked like a tower on the hill using their imagination. Each session was recorded with a digital video camera to be able to precisely measure the time participants spent with the task.

6.1.6 Results and analysis

There are three main aspects to be evaluated, namely (i) the time spent on the task users were asked to perform, (ii) differences in the predefined categories comparing HML based modeling and conventional CAD modeling according to the post-study questionnaire, and (ii) differences in the groups formed based on the pre-study questionnaire. Table 6.2 shows the statements of the post-study questionnaire and the mean value of the given answers.

Time

Participants spent 375 s in average with performing the task with CAD and 267 s in average with the HML based modeler. The Wilcoxon signed-rank test was used for comparative analysis. The result of the test shows that the time spent on the task using CAD was significantly higher ($p < .05$, $r = -.29$) than the time spent when HML was used.

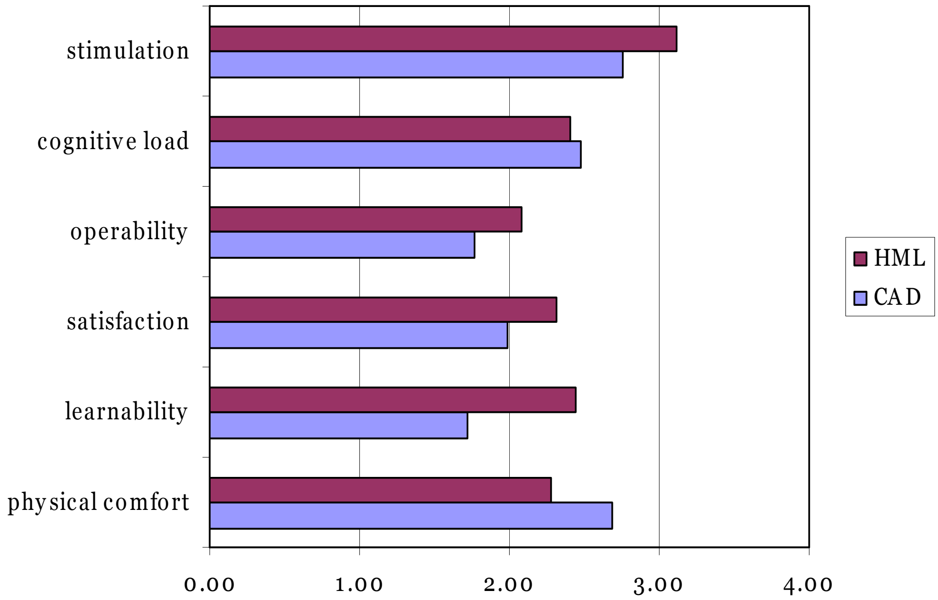


Figure 6.2 Comparison of HML and CAD based on categories

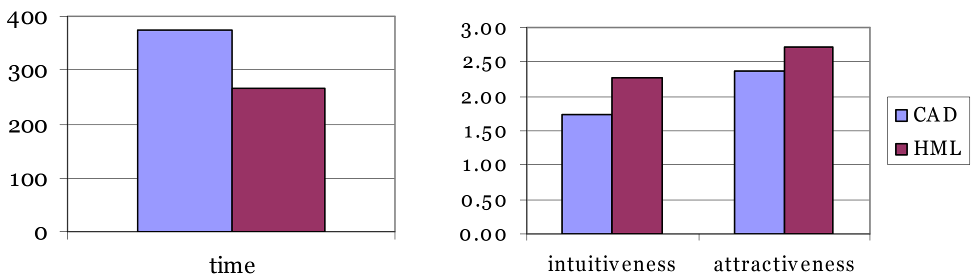
Categories

Seven categories were formed by adding positive converted statements. The categories are as follows: understandability (statement 3, 9 and 10), learnability (5, 12, 16), satisfaction (7, 11, 14), operability (6, 15, 18), cognitive load (8, 17), stimulation (4) and physical comfort (2, 13, 19). The results are shown in Figure 6.2.

Our findings are:

- HML based modeling is easier to understand ($p < .01$, $r = -.43$), which means that users (i) are sure that they are using the right command more often, (ii) are more likely to know what to do next, and (iii) understand the behavior of the HML based modeler more.
- HML based modeling is easier to learn ($p < .001$, $r = -.67$), which means that users (i) take less time to learn the controlling commands, (ii) are less likely to forget how to do things with the HML based modeler, and (iii) think it is easier to learn how to operate the HML based modeler.
- Users are more satisfied ($p < .01$, $r = -.42$) with the HML based modeler, which means that users (i) enjoy their session with it more, (ii) would recommend it to friends or colleagues more often, and (iii) would like to use it on a daily basis more often.

- The HML based modeler is easier to operate ($p < .05$, $r = -.33$), which means that (i) it is easier to make this application do exactly what the users wanted, (ii) there are fewer steps required to get something work, and (iii) tasks can be performed in a more straightforward manner.
- There is no difference found in the cognitive load on the user while operating the HML based modeler or the CAD modeler ($r = -.08$). It means that users are equally likely to feel tense or frustrated when using them.
- Participants found HML based modeling more mentally stimulating ($p < .01$, $r = -.44$), as it is specifically stated in question 4.
- CAD is less comfortable physically ($p < .001$, $r = -.58$), which means that users (i) become more tired when using it, (ii) and are more likely to feel pain in their shoulders, necks, hands and fingers.



a. b.

Figure 6.3 Comparison of CAD and HML based modeling according to (a) time, (b) intuitiveness and attractiveness

Groups

Groups of participants were formed according to gender, general computer experience, and experience with CAD software. A comparative analysis was performed with the Mann-Whitney test in order to look for differences between the groups. Results show that there were no significant differences between men and women, and between participants with beginner, or intermediate general computer experience, and with advanced general computer experience. The only significant difference is that participants with a prior knowledge of CAD software scored higher in the categories satisfaction ($p < .01$, $r = -.47$) and operability ($p < .05$, $r = -.35$) of HML, than participants without prior knowledge of CAD.

6.1.7 Discussion and conclusions

We assumed that HML-based modeling is faster, more intuitive and more attractive than CAD modeling. Speed of use was measured by the time spent on the task the users had to perform. Intuitiveness was measured by combining the categories of learnability and operability. Attractiveness was measured by the categories of satisfaction and stimulation. Together with the general satisfaction with HML-based modeling which shows to what extent the designers are willing to use the software, stimulation is also important in the conceptualization phase of design. Figure 6.3 shows that when compared to conventional CAD (i) HML-based modeling is faster (measurement in seconds), (ii) more intuitive and (iii) more attractive (according to the Likert-scale).

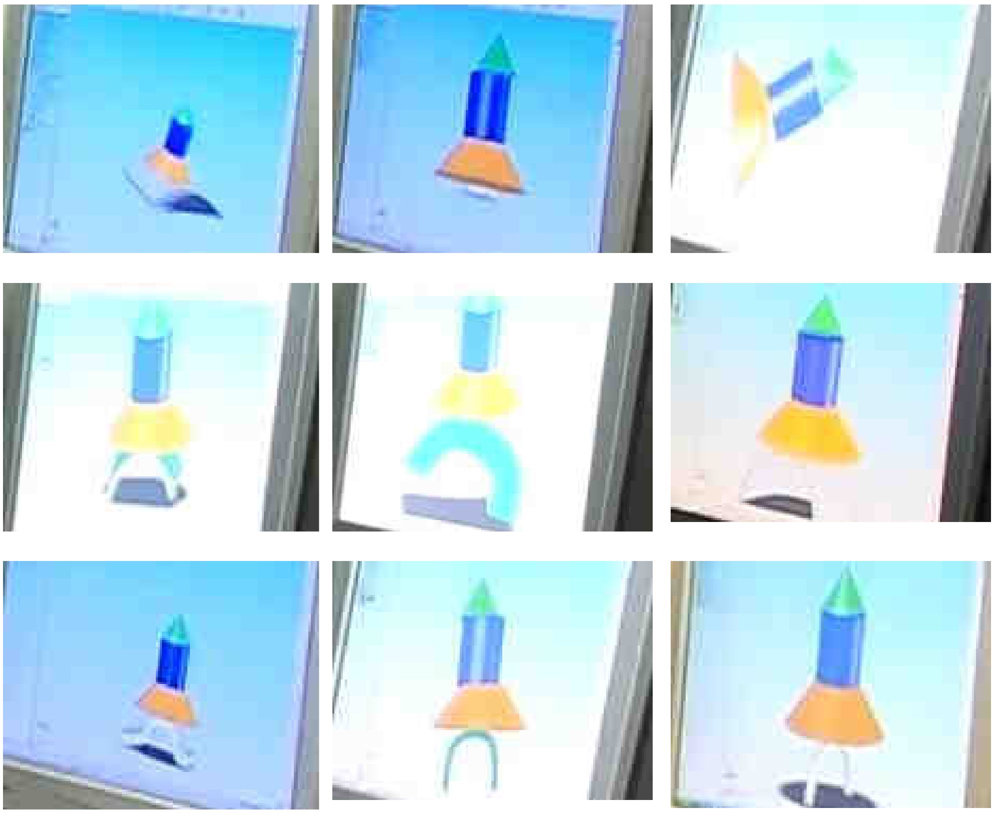


Figure 6.4 Shape variances created with CAD

Having analyzed the results of the user study, we could conclude that participants found HML-based approach more appropriate for conceptual shape design than traditional CAD. Especially the category of learnability showed significant difference in favor of HML based modeling, but the performance characteristics of operability, stimulation and satisfaction also showed considerable differences. Participants (i) were faster in creating conceptual shapes, (ii) found the hand motion input more intuitive, and (iii) were more satisfied. The last also indicated that they are more willing to use our system.

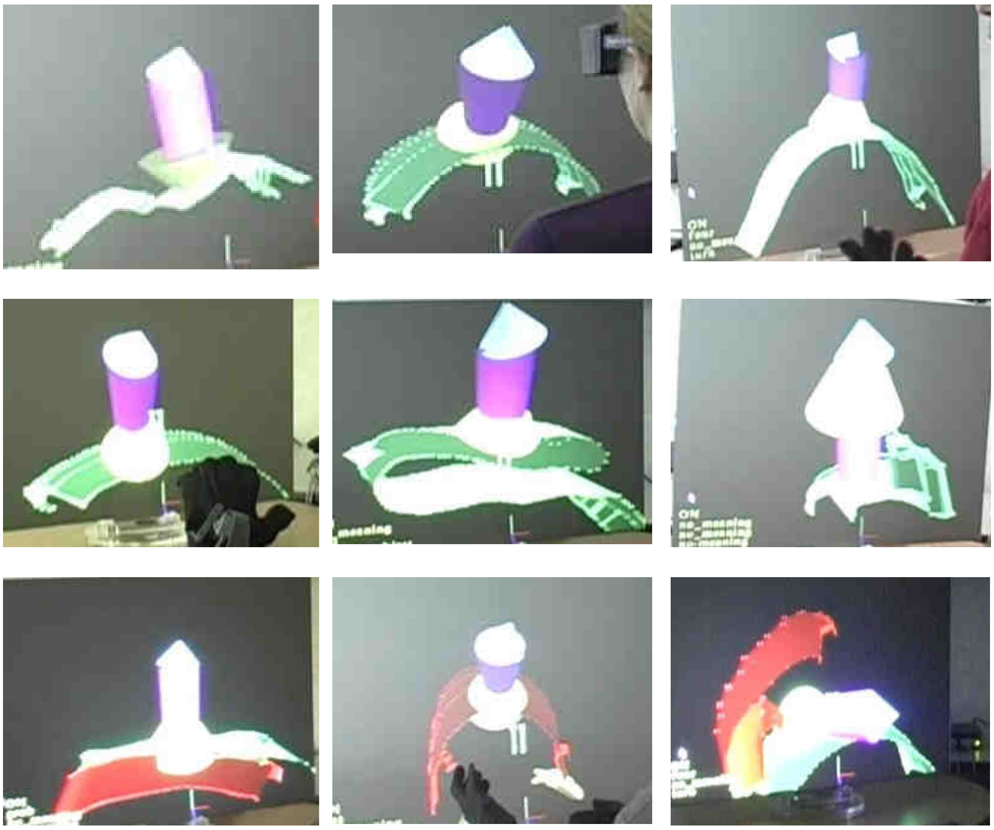


Figure 6.5 Shape variances created with HML

On the other hand, it turned out that HML-based modeling is more tiring physically than traditional CAD. However, opposite to what we expected, it turned out that the experienced CAD users found HML-based modeling to be better in terms of satisfaction and operability than the novel CAD users. An explanation for this may be the ability of the expert CAD users to adapt to this approach of shape modeling more easily due to their prior knowledge of modeling.

Looking at the generated shapes, it could also be observed that participants created a variety of shapes with the HML-based modeling method. They did not just try to copy the sample shape, which was shown to them, but created different ones based on their own imaginations (Figure 6.5). On the other hand, in the case of using CAD, they mostly concentrated on the successful completion of the task, and they cared less about the originality of their work. Therefore, the resulting shapes were very similar to each other (Figure 6.4). It is a kind of triviality that, it is very important to be creative in the conceptualization phase of product design, and to generate a variety of new shapes, from which one can be selected for further elaboration.

We also observed that different type of people reacted differently on the modeling methods. That is to say, it seemed that people, who happen to be more active, creative, and curious, liked the HML more than passive people, and they could work with software better. On the other hand, prone-to-be-nervous people had more difficulties to control the HML-based modeling software, because of their typically fast hand movements, and sudden emotional reactions when something went wrong.

6.2 Investigation of the quality of shape defined by hand motions

While Section 6.1 focused on the evaluation of human cognition aspects of hand motion-based shape design, the goal of this section is to introduce quantifiable measures for the quality of shapes, and this way to obtain information about the applicability of hand motions in conceptual shape design. In fact, we wanted to know if the achievable quality of shape definition is in harmony with the requirements of conceptual shape design. The related research questions were: (i) How well does the modeled shape reflect the morphological properties of the intended shape?, (ii) How accurately can the shape manipulation functions be performed?, and (iii) How does the complexity of the modeled products influence the number of the necessary shape manipulation words? Having analyzed the problem, we introduced some measures and qualitative indices. The indices are not only used to quantify the measures, but also to enable the comparison of the experimental results. The selected measures were: the fidelity of the generated surfaces, the accuracy of the shape manipulation functions, and the complexity of modeling.

6.2.1 Problem analysis and concepts

The challenge was to apply the measures of fidelity, accuracy and complexity in the context of hand motion-based shape definition, and to express the quality as comparable indices. These three quantified measures are necessary for answering the main research questions stated in Section 1.3. We assumed that by taking these three measures simultaneously into consideration, we can collect sufficient information to judge the actual quality of shape definition in practical cases. This can be underpinned by the fact that fidelity takes care of the morphological appropriateness of shape generation, accuracy deals with the deviations introduced by the shape manipulation functions, and complexity of modeling concerns the influences of the complexity of the shape in terms of the amount of HML words to be used correctly. As more specific hypotheses, we assumed that (i) fidelity as a primarily qualitative measure can be decomposed to quantitatively measurable shape features, (ii) accuracy of the shape manipulation functions can be studied by comparing the actual result of the manipulation with an ideal result, and (iii) complexity of modeling can be expressed in terms of the number of natural surfaces of the product related to the number of shape generation and shape manipulation functions to be used to model a certain shape. Actually, these are also the main issues for traditional CAD. While fidelity and accuracy depend on the mathematical representation of the shapes, complexity also depends on the interaction, such as the usage of menus.

Fidelity is the degree to which the representation of a shape is similar to an ideal shape. The goal of studying the fidelity of shape generation is twofold. Firstly, it is desirable to quantify the morphological compliance by investigating, for instance, entities, such as lines, planes, circles, cylinders and spheres. Naturally, these shapes cannot be formed perfectly by hand motions because of the uncertainty of hand movements and the kinematical constraints of the human hands and arms, but many algorithms are able to correct them if the maximum allowable difference between the perfect and the generated shape is known. Secondly, in the case of freeform surface generation, we wanted to learn to what extent the generated surfaces resemble the surfaces the users intended to generate.

While fidelity focuses on the morphological 'resemblance' of the shape, *accuracy* deals with the conformance to the intended (ideal) sizes of the shape. Accuracy is defined as the degree of conformity of a measurable value of a shape parameter. The purpose of analyzing the accuracy of the shape manipulation functions is to explore how well the generated shape reflects the user intended shape in terms of its actual sizes. This information is useful from the aspect of fine-tuning the algorithms related to the shape manipulation functions of the HML.

Logic implies that *complexity* of modeling depends on the complexity of the product to be modeled. Our intention was to know more about this relationship. The procedure of modeling includes alternating application of geometry generation and geometry manipulation operations. What was interesting for us was how the number of the needed geometric HML words relates to the number of the necessary manipulation HML words. We decided to express this relationship as a function of the number of bounding natural surfaces of the product and wanted to express and compare them for HML-based and CAD system-based shape modeling.

The abovementioned measures were explored with experiments involving various participants, such as designers and design researchers. From a research methodological point of view, the experiments have been designed as comparative statistical studies, where shape design by hand motions and by traditional CAD was compared. People with prior knowledge of CAD software application were selected as participants of the complexity oriented experiments. The experiments for studying fidelity and accuracy were conducted with the participation of ten people. Each participant was asked to perform the modeling task two times, and this way a data set with twenty values was collected during each measurement.

6.2.2 Overview of shape quality definitions and measurements

A forerunning literature review has been conducted in order to study how the abovementioned measures of shape fidelity are defined and measured by other researchers in similar context. While measuring accuracy and complexity in our context was clear, fidelity needed some investigation.

One typical field, where formal measures have been introduced is geometric dimensioning and tolerancing. The measures specified as tolerance values and domains (limits), and they provide a means for specifying the shape requirements and the interrelationships between part features (Feng and Hopp, 1991). Design tolerance limits express the allowable deviation from the ideal geometry. The tolerance zone method is used to specify not only the allowable size variation, but also the allowable variations of shape and of shape interrelationships (ISO 1101). Form tolerances are straightness, flatness, roundness, sphericity and cylindricity. Feature interrelationship tolerances include orientation tolerances such as perpendicularity, angularity, parallelism, and location tolerances such as position, concentricity and coaxality.

As stated in ANSI Y14.5, a tolerance zone is a virtual region formed around the true feature. Requicha proposed mathematical formulations for tolerance zones (Requicha, 1983). In his theory, a tolerance zone is a region bounded by similar perfect geometry, offset from the nominal feature surface. For physical parts, computerized dimensional measuring machines, and in computational geometry algorithms are critically needed which are functional and accurate for inspectional data analysis. The purpose of data fitting is to apply an appropriate algorithm to fit a perfect geometric form (line, plane, circle, ellipse, cylinder or sphere) to sampled data points obtained from the inspection of the generated geometric entity. For instance, circularity or roundness is a tolerance zone bounded by two concentric circles with minimum radial separation within which all measurements should lie. Other form tolerances are defined respectively, with the appropriate geometric elements as boundaries.

6.2.3 Expressing the measures by quantitative indices

Fidelity

As defined in Chapter 6.2.1, fidelity is the degree to which a particular representation of a shape resembles an ideal shape. Based on the findings of the literature study, we defined the following shape characteristics as specific (geometry dependent) measures of fidelity: straightness, flatness, cylindricity, roundness and sphericity. These are the most frequently used fidelity related measures applied in engineering modeling. Table 6.3 lists those HML words, which have consequences to the fidelity of surface generation.

Table 6.3 Fidelity with regards to geometric HML words

HML word	Fidelity aspects
Plane	straightness, flatness
Cylinder	cylindricity
Sphere	roundness, sphericity
Freeform	curvedness, formedness

To study morphological fidelity, first of all we need a reference shape that can be treated as an ideal shape and can be compared to the actual shape generated by the user during the experiments. For the purpose of a qualitative assessment, we decided to apply the method of best-fitting shape. In practice it meant that the spatial points tracked by the motion detection equipment were assessed with a view to some reference shapes. Since circles and cylinders were used as test shapes in the fidelity oriented experiment, we needed a method to construct the best-fit circle and the best-fit cylinder based on the measured points.

Finding the best-fit shape is a spatial minimization process, and we used the NLREG software (www.nlreg.com) for this purpose. In our case, the distances between the measured points and the points of the idealized shape had to be minimized. The principle of the minimization technique used by NLREG is to compute the sum of the squared residuals for a set of parameter values, which mathematically describe the idealized shape. Then, it slightly alters each parameter value and recomputes the sum of the squared residuals to see how the change of the parameter value affects the sum of the squared residuals. By dividing the difference between the original and new sum of squared residual values by the value with which the parameter was altered, NLREG is able to determine the approximate partial derivative with respect to this parameter. This partial derivative is used by NLREG to decide how to alter the value of the parameter for the next iteration. This procedure is carried out simultaneously for all parameters and is a minimization problem in the n-dimensional space where n is the number of parameters. For a detailed explanation of the minimization algorithm please refer to (Dennis et al., 1981). The construction of the best-fit circle was done by determining the origin and the radius of the circle, and the normal vector of the plane on which the circle lies, based on the measured spatial points. Similarly, the best-fit cylinder was defined by its radius, its center line defined by a point and a direction vector and its height.

The concept of Hausdorff-distance was used to measure the similarity of the generated shape and the idealized shape, and actually this has been considered to be a numerical index of *fidelity*. Hausdorff-distance is defined as the maximum distance from a set of points to the nearest point in the other set. More formally, Hausdorff-distance from set A to set B is a maximum function, defined as $h(A,B)=\max_{a\in A}\left\{\min_{b\in B}\left\{d(a,b)\right\}\right\}$, where a and b are the points of sets A and B, respectively, and $d(a,b)$ is any metric between these points. In the above formula, $d(a,b)$ is the Euclidean distance between a and b. In our case, the Hausdorff-distance is the maximum distance of the measured points and the points of the best-fit circle or cylinder. As the literature study pointed out, we can use this maximum deviation to form the tolerance zone of the shape.

In order to get the maximum relative deviation from the idealized shape, the obtained Hausdorff-distances were projected to the radius of the best-fit circle and cylinder. These values were calculated for each shape of varying sizes in each direction of shape generation, as it is defined in the next section. The deviation values indicate the utility of the HML word for the creation of the given shape.

Accuracy

This measure depends on the manipulation command. Table 6.4 lists those types of HML words, which influence the accuracy of shape manipulation. Accuracy of manipulation can be expressed in terms of the difference of the studied shape feature, such as position, distance or size, between the actual manipulated shape and an idealized shape. In order to obtain comparable values, the deviation of the actual shape from the ideal shape was calculated by relating the obtained difference values to the ideal shape feature, such as the radius of the ideal circle.

The accuracy of identification commands can also be expressed as geometric values by calculating the difference of the actual identified position and the real position of the shape. If the abovementioned distance values are below a certain limit, the accuracy of the HML word is acceptable.

Table 6.4 Accuracy with regards to identification HML words

HML word	Accuracy aspect
Identification	Difference of real and actual position
Positioning	Difference of ideal and actual position, distance or turning angle
Sizing	Difference of ideal and actual size or angle

Complexity

We assumed that *complexity* of modeling depends on the number of natural bounding surfaces of the product. The reason for this assumption is that in order to generate a model we have to generate at least these natural surfaces, but we also have to resize and position them by manipulation operations. The number of these manipulations depends on the applied technique. The aforementioned relationship can be expressed by the following expression: $N_{\Sigma} = \alpha * N$ where N_{Σ} is the total number of operations, N is the number of natural surfaces of the product, and α is the complexity index that we look for. Actually α is the number of manipulation operations, and we want to express the changes in α according to the changes in N. α is multiplied by 100 to express it as a percentage value.

6.2.4 Experiment for studying fidelity

Goal of the experiment

As it was explained earlier, fidelity of modeling has been considered to be a measure of similarity of the generated shape and the intended shape. In the practice capturing similarity involves a comparison of an idealized and an actual shape generated by the designer and recorded with the help of the motion tracking equipment. The accuracy of measurement is 0.1 mm and therefore has no influence on the evaluation of fidelity. The difficulty of generating high-fidelity shapes by hand motions originates mainly from two sources. Firstly, the kinematical properties of the human arms constraint the hands to certain motion trajectories. Due to this reason, fidelity of certain geometric entities generated in different spatial positions and orientations varies. Secondly, the hands have to work against gravity when generating shapes in the three-dimensional space and the designer has to apply varying forces in order to complete a particular motion. Gravity has influence on the forces needed to hold the hands in various postures and this should be compensated by the designer in the process of modeling by hand motions. Therefore, the goal of the experiment was to identify those spatial positions and orientations where the shapes reflect low-fidelity.

Research questions and variables

The experiments have been designed to point out

- What level of fidelity can be achieved by hand motion based modeling?
- Is this level sufficient for conceptual shape design?

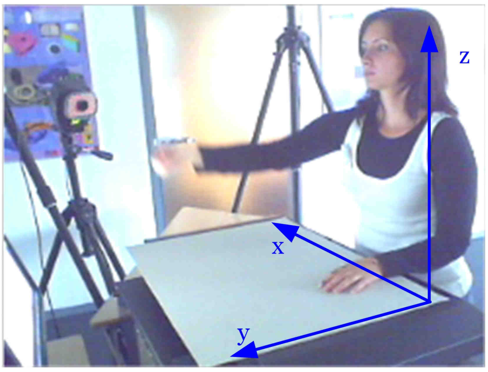
The experiment for studying fidelity has been defined in terms of various research variables. The fidelity index was in fact defined as the only output variable. The identified input variables were

- the shape to be generated,
- the spatial position and orientation of the shape in the modeling space,
- the size of the shape, and
- the fact that the shape is generated by single-handed or double-handed shape generation operations.

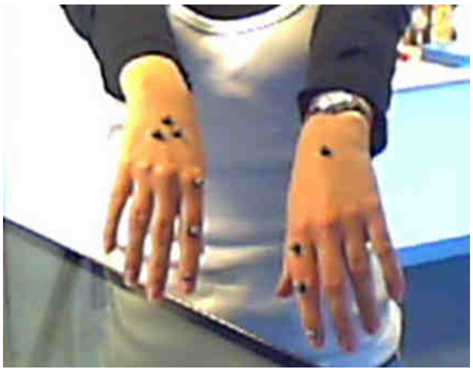
Our questions were how the input variables influenced the output variable and what the nature of the relations and correlations between the input and output research variables is.

Features influencing fidelity

For the experiments, two geometric shapes were selected to be modeled by hand motions: a two-dimensional circle and a three-dimensional cylinder, by using the relevant HML words. Participants of the experiment were asked to reconstruct circles and cylinders in different spatial positions and orientations of the modeling space, and in each situation with different sizes. The participants were supposed to generate circles by one hand and cylinders by two hands as it is defined in the HML.



a.



b.

Figure 6.6 (a) The coordinate system defined in the experiments, (b) The markers attached to the participant's hand

We were looking for the most unfavorable spatial positions and orientations of the modeling space, and one arbitrary orientation was selected by the participants according to their comfort. The results of generating the concerned shapes in the arbitrary orientation were used as control information. Based on this control information we wanted to know whether the arbitrary orientation selected by the participants have indeed resulted in a higher fidelity shape than the constrained cases. In the volume, defined by the calibration of the tracking system, the planes defined by the three axes and a diagonal orientation were selected as reference orientations. More precisely, the orientations in which the shapes had to be generated were defined by the x-y, x-z and y-z planes and a diagonal plane which connects the lower left and upper right edges of the modeling space. The user of the system faces the modeling space in a body posture in which she stretches her right arm to the right it gives the x direction, when she moves her left arm forward it gives the y direction, and z is perpendicular to the x and y axes and points in the direction of the user's head, as it is shown in Figure 6.6 a.

In the diagonal orientation, circles had to be generated on the plane as defined earlier. Cylinders had to be generated in a way that they connect the closer left and farther right corner of the modeling space. The generation of circles and cylinders in this diagonal plane is considered difficult due to the varying control of moving the hands in this plane. Furthermore, it is known to be difficult to generate the circle in the y-z orientation and the cylinder in the x-y orientation due to the kinematical constraints of the human arm. The orientations of the planes defined by the reference axes of the modeling space were selected to be tested for fidelity because of the fact that in many situations the designers want to generate shapes in these orientations as they imagined to be placed on the top of the table. In each situation, the participants were asked to draw the concerned shapes in three sizes, namely, small, medium and large, to be able to cover a large part of the modeling space. More precisely, when generating a small shape, no significant arm motions occur. In case of medium sizes, the arm and hand movement is normal and within the motion envelope of the arms without any trunk movement and without applying uncomfortable forces to reach a certain spatial position. Generating large shapes means that the user has to apply trunk movement or stretch his arms in an uncomfortable way to be able to set the size of the shape.

Evaluation method

As an evaluation of the results, the shapes were compared to the best-fit shape which could be regenerated based on the result that the user generated in the experiment. The ideal shape was defined by a best-fit method and the comparison was done by calculating the Hausdorff-distance between the ideal shape and the shape represented by the set of points detected by the tracking system (see details in Chapter 6.2.3).

Design of the experiment

The fidelity related experiment was conducted with ten participants. The participants were industrial designers and design researchers, all right-handed. They were asked to draw circles and cylinders in the three-dimensional space using hand motions. These test shapes were selected because they need continuous control of hand motions to keep the constant curvatures and diameter. In the beginning of the experiment, markers were placed on the participant's hands as it is shown in Figure 6.6 b. The different pattern of markers on the right and the left hand was formed in order to be able to make a difference between the hands, and this way to identify the concerned markers. This is actually a requirement for the tracking process.

Before the actual measurement, the hand model in the software of the tracking system was calibrated to the participant's hand to eliminate the effect of the differences of hand sizes. The three-dimensional coordinates measured by the motion tracking device were collected into a text file for further analysis. The device was set to collect the coordinates in 60 Hz. In the case of generating circles, only the coordinates of the marker on the right index fingertip were collected. In the case of generating cylinders, the coordinates of one marker on the wrist and three markers on the index finger were collected for both hands. These coordinates served as input for defining the best-fit circle or cylinder, respectively.

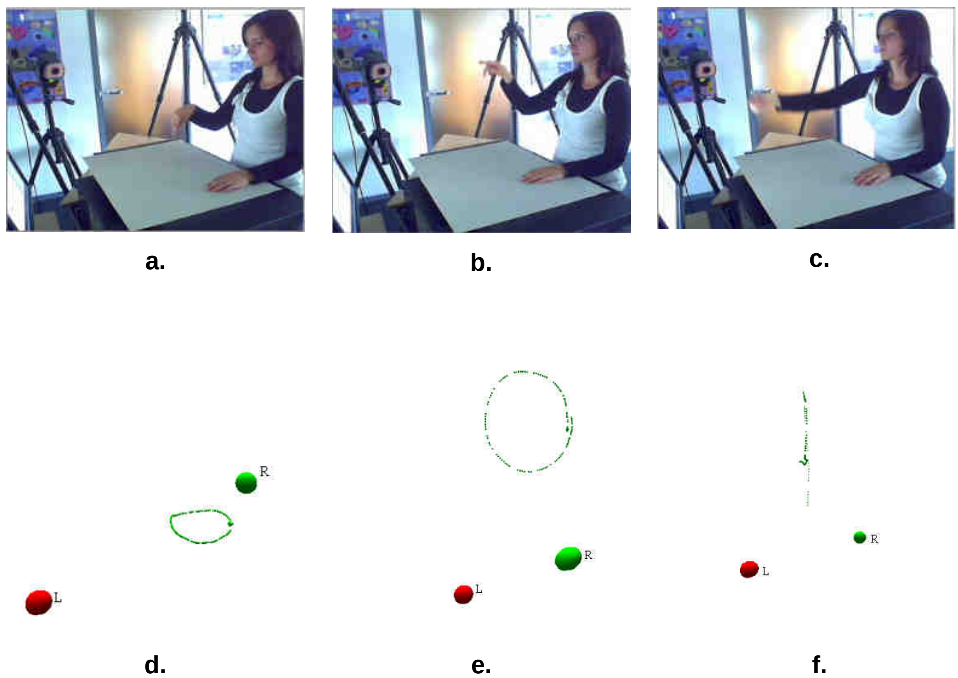
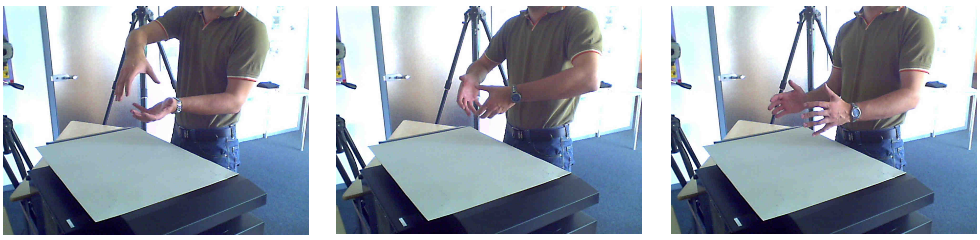


Figure 6.7(a-c)A participant generating circles in the x-y, x-z and y-z planes, (d-f) The resulting shapes as feedback for the conductor

The protocol of the experiment was the following. After attaching the markers to the participant's hand, she was asked to stand in front of a horizontal table, whose upper surface represented the physical constraint of the working space, likewise in case of a horizontally placed holographic display. The height of the table was 95 cm. The capture volume was (-300,700) in x direction, (-300,700) in y direction and (-100,900) in z direction. The participant was asked to maintain this position while generating the HML words, to use only arm movements and the least trunk movement possible. In the first part of the experiment, the participant was asked to draw circles on several imaginary planes. The first plane was selected by the participant, which s(he) thought the most comfortable to be used.

Then the x-y, x-z, y-z and the diagonal plane came in a random order. In case of the x-y plane, the participants were specifically asked to maintain the position of their right hand in the air, not letting it down to the surface of the table. The generation of a circle in the x-y, x-z and y-z plane can be seen in Figure 6.7 a-c. In each of the planes, the participant was asked to generate the circle twice in three sizes, namely, small, medium and large. They were told to generate the small circles with a minimum hand motion, the medium circle in their comfortable range of motion and when generating the large one, they had to try to use their whole working volume. This way we collected twenty measurements for each size in each plane.

In the second part of the experiment, the participants were asked to generate cylinders. The protocol for this part of the experiment was similar to the previous case, when circles were generated. The first orientation was selected by the participant, and then the participant was instructed to draw cylinders in the x, y, z and diagonal orientations in a random order, and in different sizes. See examples in Figure 6.8. As in the first part of the experiment, small, medium, and large shapes were requested. The participants were explained that these measures concern the height of the cylinder, and not the diameter. Again, two measurements were done for each size in each orientation to have twenty measurements for data analysis.



a. b. c.

Figure 6.8 Generation of cylinders in the (a) x, (b) y and (c) z directions

In the fidelity and accuracy oriented experiments, we put the emphasis on the analysis of hand motions, instead of the visual feedback for the user. In the fidelity oriented experiment, the participants did not have any visual feedback. Instead, they were asked to look at their hands and imagine the generated shape directly on the motion trajectory of their hands, as it would appear in a truly three-dimensional holographic display.The conductor of the experiment had visual feedback with the resulting shape on a traditional two-dimensional display (Figure 6.7 d-f). In the figure, the sphere denoted by *L* represents the global position of the left, and the sphere denoted by *R* the global position of the right hand.

6.2.5 Experiment for studying accuracy

Goal of the experiment

Besides fidelity, which concerns the similarity of an idealized shape and a hand motion generated shape, we were also interested in the geometric spatial accuracy of entity manipulation in the process of modeling. Manipulation operations are complementary actions of entity generation in the model building process. We must not forget that fidelity provides an index for the accuracy of entity generation implicitly. Fidelity deals with shape similarity from a morphological aspect, on the other hand, accuracy is related to the measurable sizes of the generated geometric entities.

a. b. c.

Figure 6.9 Working in the modeling workspace
(a) Scaling in the x direction, (b) in the y direction, (c) in the z direction

Below we present the experiment, which was designed to study the accuracy of the manipulation words of HML in different circumstances. The experiment was conducted by capturing the differences between an ideal shape and the actual size or position of the geometric entities generated by hand motions in specific design situations.

The reason for performing this experiment is given by the fact that there are certain difficulties associated with hand motion based geometric entity definition and manipulation. The difficulties come from the uncertainty (i.e. natural shaking) of hand motions. In addition, the anthropometric features and kinematical constraints of the arm and hand motion also influence the accuracy, which might be critical in certain spatial positions and orientations during shape manipulation. Our experiment tried to explore what model sizes, spatial positions and orientations can be critical for the achievable accuracy.

Research questions and variables

Our research questions were

- What level of accuracy can be achieved by hand motion based modeling?

- What influence it has on conceptual shape design?

The experiment has been defined to find the relationships between the related input research variables, and the accuracy, as output variable. The identified input variables were

- the purpose of the investigated manipulation operation,
- the number of the hands used,
- the relative motion of the hands (in case of two-handed HML words),
- the spatial orientation in which the manipulation is performed, and
- the range of the manipulation motions.

In the end we have a quantitative measure (index) of the accuracy of manipulation.

Features influencing accuracy

The manipulation words of HML are of two kinds: (i) those which identify a geometric entity, and (ii) those which modify a geometric entity. In this experiment, two HML words were selected to test the accuracy of the manipulation HML words, namely, *identify point* and *size by surfaces* (Table 6.4). The identification of a point can be critical in the case of a dense point cloud. This explains why we wanted to know how accurately points can be identified. In the case of entity modification, the basis of modification is a set of translational motions. We found that all geometric entity manipulation words can be processed by similar algorithms. Therefore it is enough to consider only one in the experiments. We have chosen the *size by surfaces* operation as the objective for our experiment.

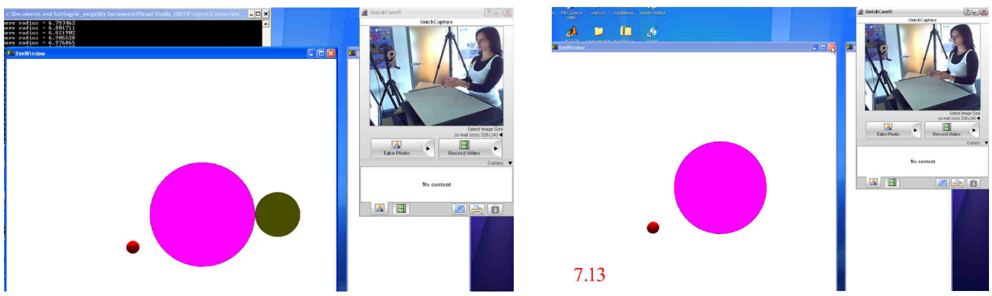
It has to be mentioned that in the case of two-handed HML words, the orientation of the manipulation also influences the comfort of execution. Similar spatial orientations were selected to be studied as in the case of the fidelity oriented experiment based on the same reasoning (Figure 6.9). As shown in Figure 6.10 and Figure 6.11, we assumed given modeling situations and asked the participants to touch a virtual geometric element with other ones, or, in other words, to bring them into virtual contact by various hand motions.

Method of evaluation

To evaluate the above experiments, the differences between the spatial position and the size of the ideal shape and the manipulated shape were calculated. More exactly, when setting the size of the sphere to get in contact with the other sphere (Figure 6.10), the distance between the contact surface and the manipulated surface was calculated. When the size of the sphere was set by the participant so as to get in contact with the other sphere, the calculated distance was compared to the radius of the ideal sphere, and the deviation was expressed in percentages. In the case of point identification, the distance between the position of the ideal point and that of the manipulated point was calculated.

Design of the experiment

The experiment was conducted with nine participants, different than the participants in the fidelity oriented experiment. The participants were asked to perform each modeling task five times, so altogether fourty-five data were collected for analysis. The participants were asked to manipulate shapes which were presented them on a two-dimensional display. It has to be noted that, by choosing a conventional display, we wanted to avoid the biases on the experiments, which would have been caused by the novelty and unusual features of the available holographic imaging device. That is, the experimental environment was the same as described in Chapter 6.2.4. The participants were given the same instructions regarding their body position and posture.

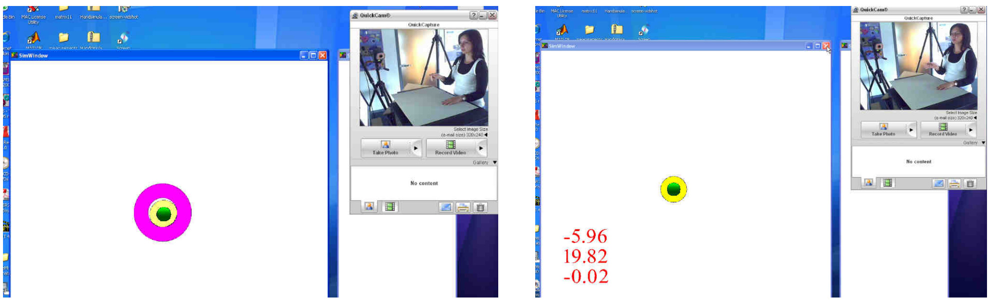


a. b.

Figure 6.10 Screenshots from the experiment regarding sizing (a) with graphical feedback, (b) with numeric feedback

In the first part of the experiment, participants were asked to manipulate the size of a circle using the *size by surfaces* HML word. The participants were told to imagine this circle as a sphere between their left and right hands. The manipulation operation was performed in the x, y, and z directions, as shown in Figure 6.9. The difference between the radius of the ideal shape and the manipulated circle was measured by subtracting the actual radius value from the radius value of the ideal shape and taking its absolute value. The measured values were collected and recorded in a text file throughout the experiment. When the user was satisfied with the result of the manipulation, she said “stop”, and the collection of the data was finished by the conductor of the experiment. In the evaluation, the average value of the last ten measurements in the text file was used to compensate the time delay between the end of the manipulation, its verbal indication and the actual ending of recording.

The above described experiment was conducted twice with different visual feedbacks. Firstly, the participants were asked to set the size of the circle to be in contact with another circle on the screen, as shown in Figure 6.10 a. Secondly, the participants were asked to set the radius of the circle to be 7, and the value of the radius was shown by numbers, as shown in Figure 6.10 b. (Actually, the value 7 was got by dividing the actual measured values in mm by 20, to be able to fit on the screen.) The reason of selecting two visual feedbacks was that we were interested whether there is a difference in accuracy between these two cases.



a. b.

Figure 6.11 Screenshots from the experiment regarding positioning (a) with graphical feedback, (b) with numeric feedback

In the second part of the experiment, the participants were asked to set the coordinates of a point to a specific value in the three-dimensional space. Likewise in the first part of the experiment, two different feedbacks were provided, for the same reasons. In the case of the first type of feedback, participants were asked to set the position of the small circle to be in contact with the inner circumference of the annulus (Figure 6.11). When the circle and the annulus are in perfect contact, it means that the origin of the circle is exactly the ideal spatial position. In case of the second type of feedback, the x, y, and z coordinates of the origin of the circle were presented on the screen by numbers, and participants were asked to set it to (-5,20,0). Again, the end of the manipulation action was indicated by the participant by a voice command. Until that point in time, the distance between the ideal and the actual origin of the circle was measured and recorded in a text file. In the evaluation, the average of the last ten values was used, for the same reasons as in the first part of the experiment.

6.2.6 Experiment for studying complexity

Goal of the experiment

It can be seen intuitively that the complexity of the shape has a strong influence on the total number of modeling actions. With regards to hand motion based shape modeling, the difficulty of generating shapes of high complexity mainly originates from two sources. First of all, the number of the bounding surfaces of the shape to be generated by a single hand motion influences the number of manipulation operations to be performed to construct the shape. Secondly, the number of local features the shape contains also has an effect on the complexity of modeling.

Local features are small sized relative to the global shape, at the same time might have large number of natural surfaces. Therefore several manipulation operations may be needed to arrive at the final shape, size, position and orientation of a local feature. This implies that the complexity of modeling has three components: (i) the total number of operations generating the natural surfaces, (ii) the total number of construction operations needed to construct the compound shape, and (iii) the total number and the morphology of the local features. Like the global shape, local features should also be generated in the comfort zone of hand motions, where shapes are clearly visible and the arm movements are not constrained by their kinematical limits. For this reason, the local features generated in “natural” scale has to be resized relative to the global shape, and finally positioned to the required spot of the global shape.

Consequently, the goal of the experiment was twofold. First we wanted to determine the relation between the morphological complexity of the shape and the complexity of the modeling process. The complexity of the modeling process was expressed by the total number of generative and manipulative operations throughout the modeling of the shape. For this reason, three different types of shapes were selected to be modeled in the experiment, and in all the three cases, the number of operations was counted until having a certain level of details. Second, we wanted to compare the complexity of modeling by hand motions and traditional CAD software. We were especially interested how the increase of dimensionality from two-dimensions to three-dimensions influences the complexity of modeling.

Research questions and variables

Our research questions were:

- What level of complexity can be handled by hand motion based modeling?
- Is it sufficient for conceptual shape design?

The experiment for studying complexity of modeling has been defined in terms of input and output research variables. Complexity of modeling was defined as the only output variable. As input variables complexity of the shape was defined from two aspects: (i) morphological structure of the shape to be modeled and (ii) level of details of the shape. The morphological structure can be (i) simple, (ii) compound or (iii) complex; and the level of details can be (i) draft global, (ii) detailed global or (iii) featured detailed. Our question was how the input variables influence the output variable.

Features influencing complexity

For the complexity oriented experiment, three shapes were selected to be modeled by hand motions: a simple (prismatic), a compound (polyhedral) and a complex (freeform). The number of generative and the number of manipulative operations both influence the complexity of modeling. In case of entity generation, several problems can occur. When generating shapes with hand motions, in some situations the natural surface is bigger than the shape that can be generated. In these situations the shapes can only be formed by several hand motions which affect each other. In case of traditional CAD software, freeform surfaces can only be formed by defining their control points or deterministic curves and applying a surfacing method on them. Consequently, the complexity of geometry generation depends on the shape representation method of the modeling software. Therefore, one expert user of traditional CAD and one expert user of the HML modeling method were asked to model the abovementioned shapes. We assumed these cases as optimum surface generation processes. To get valid results, the generation of geometric elements was not counted.

Evaluation method

We were interested in the changes of the number of manipulation operations when the complexity of the model increases. For this reason, the number of modeling operations was counted in each modeling situation in the abovementioned three levels of shape complexity. Complexity of modeling was defined as the α value as it was described in Section 6.2.3.

To evaluate the results, we needed a measure that can be compared and is independent from the geometric representation of the shapes in the modeling system. Complexity of modeling depends on the number and type of the surfaces of the product model. The problem with comparing two different methods – HML and CAD – is the fact that every modeling system has its own surface representation method, which influences the number of actions by which a certain shape can be formed. Therefore, we used the number of the natural surfaces of the model, which is independent from the modeling systems, and is unique for each shape of a given morphology. The global shape of the product was deconstructed to natural surfaces based on changes in continuity.

Design of the experiment

Three products were remodeled in this experiment, namely, a CD case, a chair and an electric shaver. The reason for selecting these objects was the different levels of their complexity. The CD case had a simple geometry containing only planar surfaces and no local features. The chair had a compound geometry and was constructed from polyhedral parts and cylinders, and contained two cylindrical and two freeform local features. The shaver had a complex geometry constructed from freeform surfaces and contained three cylindrical and one freeform local features.

6.2.7 Results and analysis

Fidelity

The results of the fidelity-oriented experiments can bee seen in Table 6.5, Table 6.6 and Table 6.7.

Table 6.5 Hausdorff-distances in case of cylinders

CYLINDER	arbitrary=z	x	y	diagonal
small	12%	16%	17%	19%
medium	14%	22%	21%	26%
large	18%	23%	20%	31%

The calculation was done for each size of shapes created in each direction. At generating cylinders, all participants preferred the z direction as the most comfortable arbitrary, user-selected direction. Figure 6.12 shows examples for the measured points of the generated small, medium and large cylinders in the participant-selected orientation.

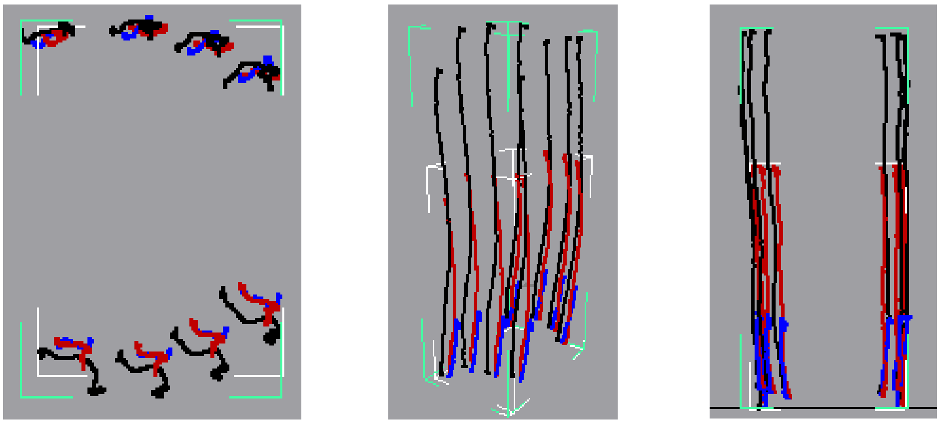


Figure 6.12 The motion trajectories in case of constructing a cylinder in the preferred orientation (in different views)

Figure 6.13 shows the trajectories describing the generated small, medium and large cylinders in the case of diagonal direction. The motion trajectories show that in this case the motion of the hand was not so stable as it is seen in Figure 6.12.

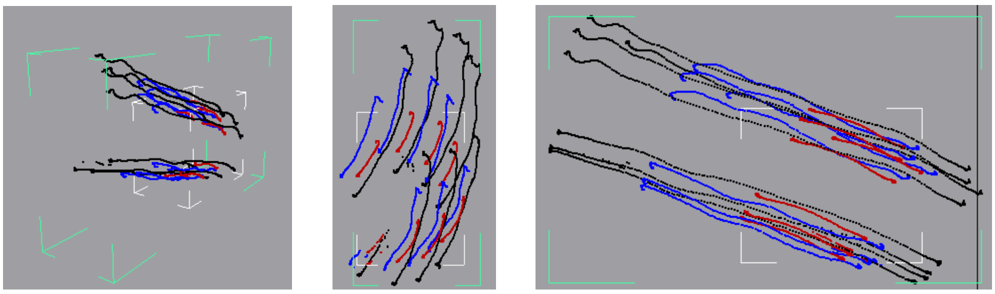


Figure 6.13 The motion trajectories in case of constructing a cylinder in the diagonal orientation (in different views)

When circles were the test shapes, the measured coordinates were used to generate the best-fit circle. For each measured point, the distance between the measured point and the best-fit circle was calculated, and then the maximum distance was defined. Also in this case, this maximum distance was defined as the Hausdorff-distance.

Again, the Hausdorff-distance was projected to the radius of the best-fit circle to be able to calculate the maximum deviation as percentages. The results can bee seen in Table 6.6. In order to evaluate also the planarity of the circle, the best-fit plane was calculated and the Hausdorff-distance of the set of measured points and the best-fit plane was calculated. The results can be seen in Table 6.7.

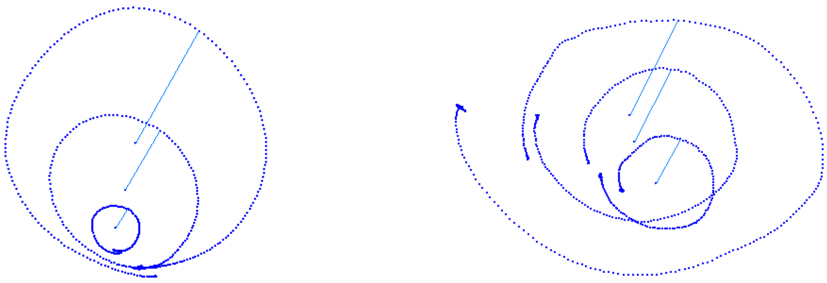
Table 6.6 Maximum deviation of the measured points of the circles and the corresponding best-fit circles

	small	medium	large
arbitrary	21%	11%	11%
xy	21%	15%	12%
xz	23%	17%	16%
yz	26%	21%	19%
diagonal	27%	21%	21%

Table 6.7 Hausdorff-distance of the points of the generated circles and the corresponding best-fit planes

	small	medium	large
arbitrary	5	9	11
xy	6	13	19
xz	12	26	30
yz	7	10	14
diagonal	38	109	157

Figure 6.14 a shows an example for high fidelity circles in small (fidelity is 8%), medium (8%), and large (4%) sizes. These circles were generated in the selected orientation, which in this case was the xy orientation and the participant pushed his finger to the table.



a. b.

Figure 6.14 (a) Small, medium and large sized circles of high fidelity, (b) Small, medium and large sized circles of low fidelity

Figure 6.14 b illustrates an example for low fidelity circles in small (fidelity is 33%), medium (26%), and large (28%) sizes. These particular circles were generated in the diagonal orientation. The estimated centers of the circles are also seen in the figures.

Accuracy

In this experiment first the participants defined the size of a circle, and then they were asked to set the position of a point. The results of setting the size and positioning in the x, y, and z directions can be seen in Table 6.8.

Table 6.8 Accuracy of setting size

		Difference (in screen measures)	Difference (in mm)	Deviation
with graphical feedback	x	0.06	1.23	0.88%
	y	0.05	1.05	0.75%
	z	0.07	1.40	1.00%
with numeric feedback	x	0.04	0.81	0.58%
	y	0.03	0.62	0.44%
	z	0.04	0.88	0.63%

Table 6.8 and Table 6.9 show the Hausdorff-distance both in screen measures, which was presented to the user, and in millimeters. As it was mentioned earlier, the actual measurements in millimeters were divided by 20 to fit on the screen.

Table 6.9 Accuracy of setting the position of a point

	Distance (in screen measures)	Distance (in mm)
with graphical feedback	0.36	7.20
with numeric feedback	0.35	7.15

Complexity

The results of the complexity-oriented experiment can be seen in Table 6.10 and the products to be modeled in Figure 6.15. The total number of operations was counted both in the case of CAD and in the case of HML-based modeling.

Table 6.10 Complexity of modeling

	Product 1	Product 2	Product 3
No. of natural surfaces (N)	18	48	40
No. of operations CAD ($N_{\Sigma,CAD}$)	15	96	315
No. of operations HML ($N_{\Sigma,HML}$)	26	94	48
$\alpha_{CAD} = N / N_{\Sigma,CAD} * 100$	120%	50%	13%
$\alpha_{HML} = N / N_{\Sigma,HML} * 100$	69%	51%	83%

The number of natural surfaces is a property of the shape and independent from its representation in the modeling system. The α efficiency value was calculated both for HML-based modeling and CAD modeling as it was defined in Section 6.2.3.

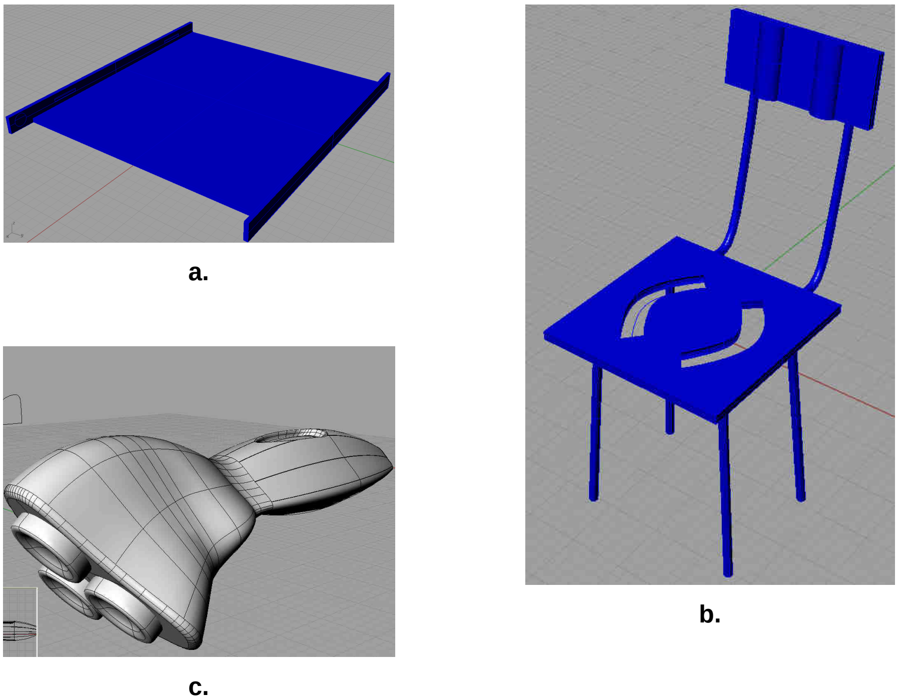


Figure 6.15 The modeled objects in the complexity-oriented experiment, (a) a part of a CD case, (b) a chair and (c) en electrical shaver

In the case of CAD modeling, all menu item or icon selections, switching between views, translations and rotations, zooming operations, each elements of part definitions and selection operations were counted. In the case of HML-based modeling the number of the HML words issued was counted.

6.2.8 Discussion and conclusions

Fidelity

Figure 6.16 shows the results of generating cylinders. It can be seen that the cylinders in the z direction show the highest fidelity. It cannot be separated from the fact that this direction was selected by all participants as the most comfortable direction. The most difficult direction was the diagonal one. As for the size of the cylinders, it can be seen that in the z and the diagonal directions the small cylinders show the highest fidelity, and large cylinders the lowest fidelity.

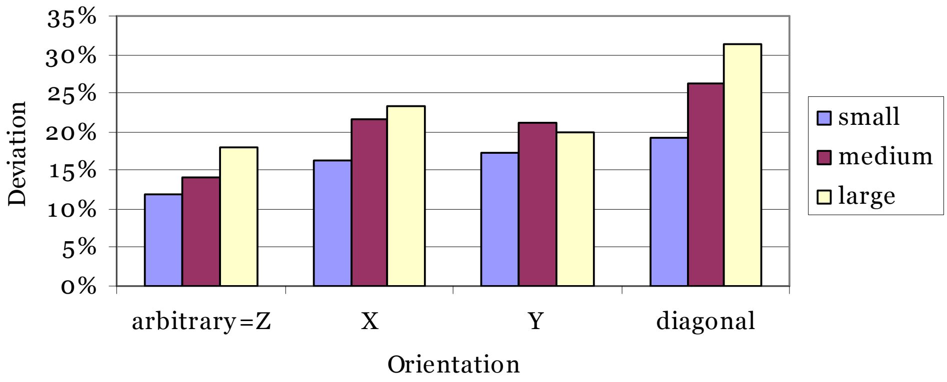


Figure 6.16 Hausdorff-distances from the best-fit cylinder in different orientations

In case of the x and y directions, again the small cylinders show the highest fidelity, and the fidelity of the medium and large cylinders are close to each other. In general, we can say that the cylinders of large fidelity values were generated in the z orientation in small sizes. Cylinders of low fidelity values were created in the diagonal orientation in large size. From the aspect of shape modeling, we can say that the best results can be achieved when first a small cylinder is created in the z orientation, and then its size is manipulated according to design intent.

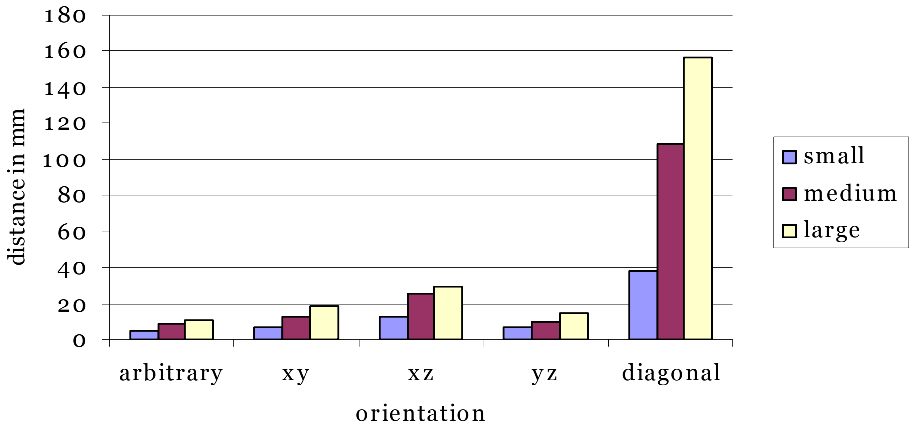


Figure 6.17 Planarity of the generated circles grouped according to orientations

Figure 6.17 illustrates the planarity of the generated circles in different orientations. It can be seen that the arbitrary, the xy, and the yz orientations all show good results. In all of these cases, for all sizes of circles the distance from the best-fit plane is less than 20mm. In the xz orientation, the values are a bit higher, but even in the worst case the difference is not more than 30mm. It could have been predicted intuitively that the circles generated in the diagonal orientation showed always the worst values in terms of planarity.

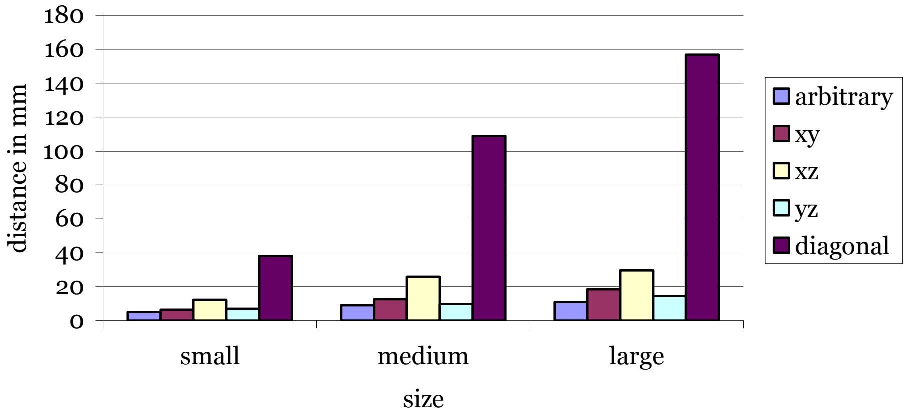


Figure 6.18 Planarity of the generated circles grouped according to sizes

Except the yz, in each orientation the small circles showed the lowest, and the large circles the highest planarity value (Figure 6.18). When the value of planarity was compared against the sizes of circles, it can be seen that in each size the picked up orientation showed the best planarity values, and the diagonal one the worst. The best planarity values were received when the size of the circle was small. It has to be mentioned here, that seven from ten participants chose to use the table to give a support to their finger, consequently, the planarity values became very high in this case. Actually, the plane of the table was the xy orientation. If the participant decided to use the table as a support, she was asked to repeat it by holding her hand in the air. In all sizes the xy and the yz orientations also show good values, the xz a bit higher, and the diagonal orientation is remarkably worse than the others.

When circularity is concerned, it can be seen in Figure 6.19, that in each orientation, the medium and large sized circles did not show large differences. At the same time, the values obtained for the small circles indicate that these ones show the worst circularity.

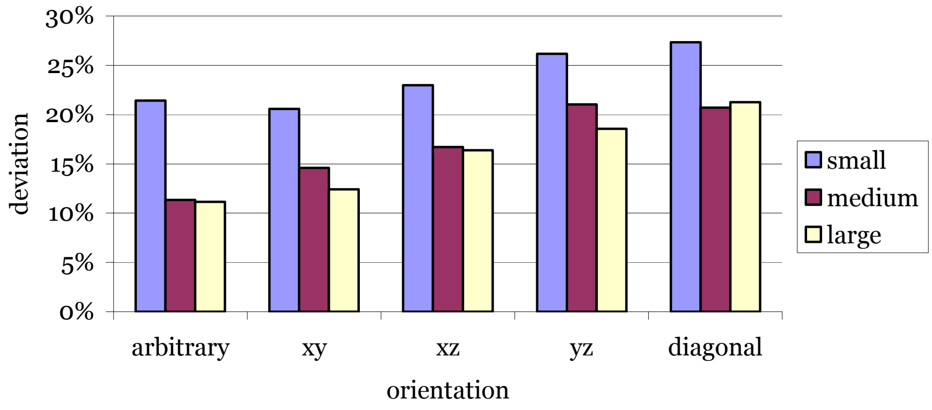


Figure 6.19 Circularity of the generated circles grouped according to orientations

However, for the circles of the three size ranges, it can be seen in that only a small difference was found between acting in the arbitrary and the xy orientations. Figure 6.20 shows that the best circularity values were found for these cases. The xz orientation shows a bit higher values in each sizes of circles, and the worst ones are the yz and the diagonal, without significant difference.

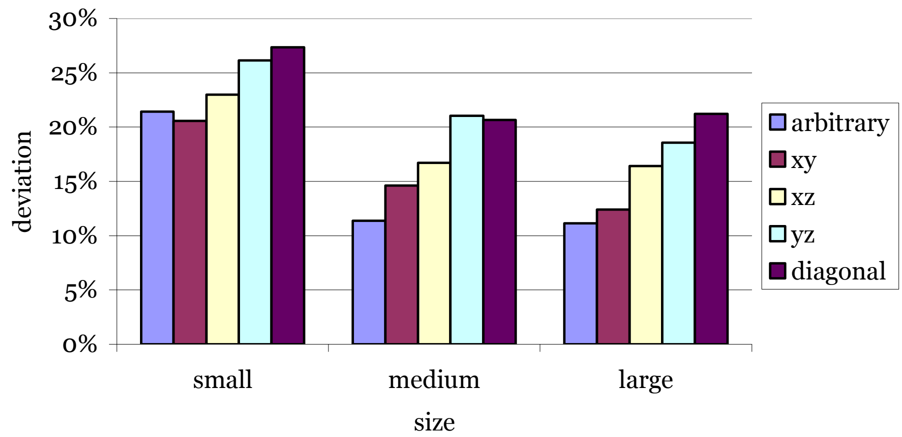


Figure 6.20 Circularity of the generated circles grouped according to sizes

Accuracy

The results of this experiment show that very high accuracy values were achieved when the size of the shape was set by hand motions. Although the values are good in each direction, the best results are related to the y orientation. Actually, the participants mentioned at the end of the experiment that they had more control over their hand motions in this orientation than in the other orientations.

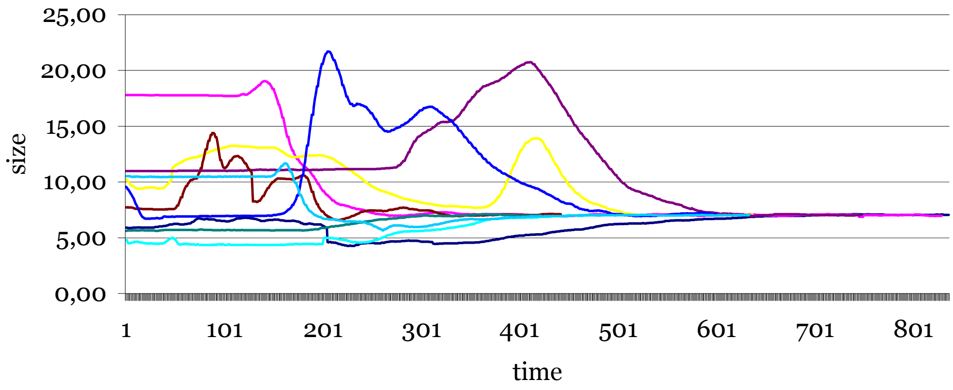


Figure 6.21 Setting size over time in design task

It is interesting to mention that there was no time limit or restriction in the experiment. It resulted in a situation that participants first tried to reach an approximate value and then made small changes to refine it. Figure 6.21 and Figure 6.22 show a particular measurement with all nine participants with the two types of visual feedback.

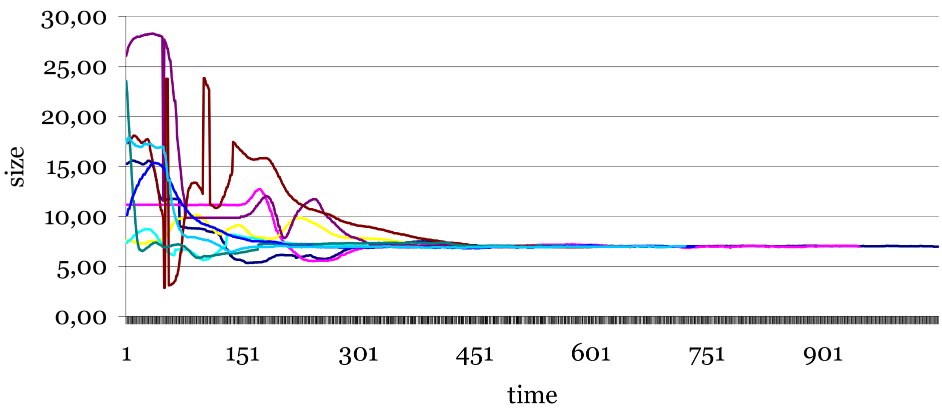


Figure 6.22 Setting size over time with numeric feedback

Figure 6.21 illustrates that in case of graphical feedback participants reached the approximate value around the 430th - 630th frame of measurement (7.2 - 10.5 s, 4% deviation), and refined it averagely for about 3 s. Figure 6.22 shows that in case of the numeric feedback the approximate value was reached around the 210th - 450th frame (3.5 - 7.5 s, 4% deviation) and it was refined for about 7 s.

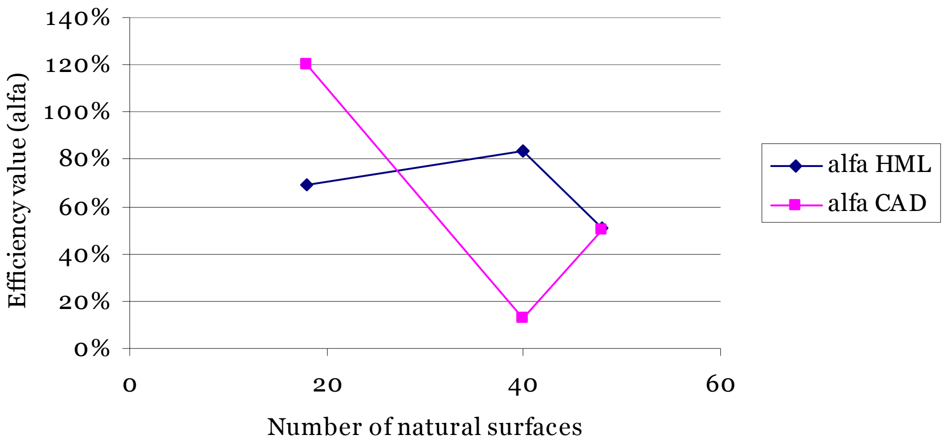


Figure 6.23 Changes in efficiency value according to the number of natural surfaces

We can conclude that this experiment pointed out the maximum achievable accuracy of manipulation when there is no time constraint. It can be seen that the participants strived for a higher accuracy when numeric feedback was provided, and they were satisfied with a bit lower accuracy when they were put into a design situation. However, this latter accuracy is suitable for shape conceptualization.

Setting the position of a point showed similar patterns than the ones presented for setting size. However, adding two additional dimensions to the task (in case of setting the size the participant controlled the radius of the circle, this time (s)he controlled the x, y and z coordinates of the point) caused a decrease in accuracy and an increase in time. The different types of visual feedback did not influence the results, and approximately the same accuracy was reached in both situations.

Complexity

Figure 6.23 shows the changes in the efficiency value of complexity according to Table 6.10 both for HML-based modeling and CAD modeling. The figure shows that in case of a small number of natural surfaces (18) CAD modeling shows a much higher efficiency value than the efficiency value of HML-based modeling.

Actually, the reason for this is the prismatic nature of Product 1 (CD case). Using CAD, boxes can be easily created by applying the solid modeling functionality of the software. However, in HML there is no special hand motion for generating boxes. They are generated by drawing the six surfaces of the box and by constructing them. Generally, we can say that a fully prismatic shape can be defined easier using CAD than using the current version of the HML vocabulary.

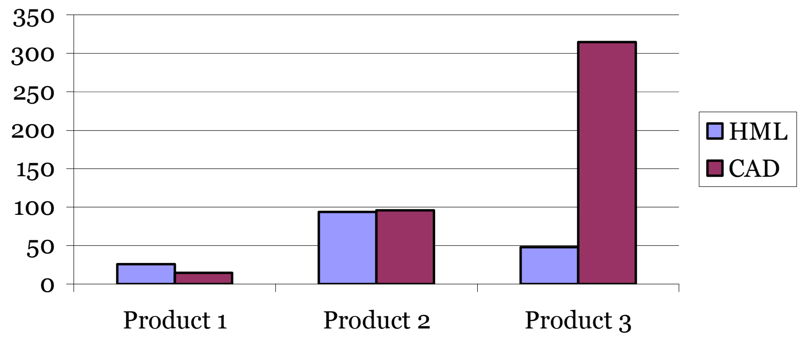


Figure 6.24 Total number of operations needed to create the three products in case of CAD- and HML-based modeling

When the number of natural surfaces is around 40, CAD modeling shows a very low and HML-based modeling a very high efficiency value. In fact, the reason for this is the shape of Product 3 (shaver), which is mostly formed by freeform surfaces. While freeform surfaces can be easily generated using hand motions, it is rather complex using CAD.

Around 48 natural surfaces, the HML-based and CAD modeling shows an almost equal efficiency value. However, the reason for this is the shape of Product 2 (chair), which mostly contains cylindrical surfaces and some prismatic forms. Although the generation of prismatic shapes requires more steps using HML-based modeling, it is compensated by the fact that cylindrical shapes are rapidly generated using the appropriate HML word and the fact that the positioning of the surfaces is much easier using HML because of its spatial nature. It can bee seen in Appendix D that CAD modeling requires a vast amount of translational and rotational operations in order to position the surface to the correct spot.

Figure 6.24 shows the total number of operations in both CAD and HML-based modeling. This number includes the geometric and the manipulation words issued during modeling. The figure shows, that in the case of the first product (CD case), the number of operations needed for modeling the shape was with HML was approximately half of the number of operations in the case of CAD modeling. Due to the prismatic nature of the shape elements of the second product (chair), the number of operations approximately equals in the cases of the two types of modeling.

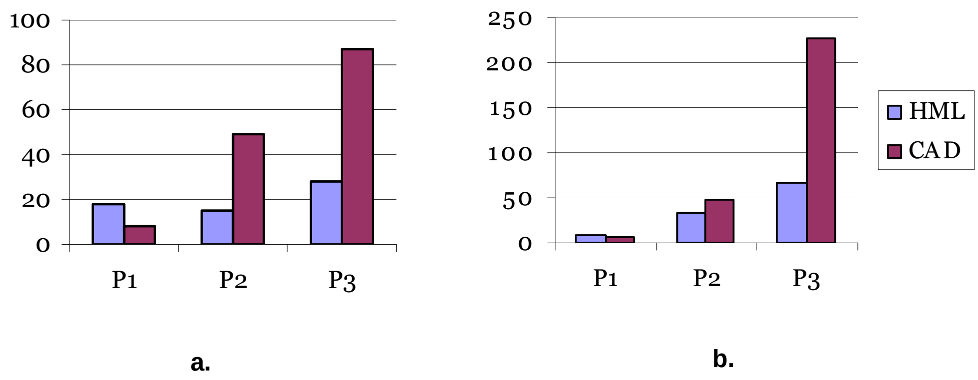


Figure 6.25 Number of (a) geometric and (b) manipulation operations needed to create the three products in case of CAD- and HML-based modeling

While prismatic shapes could be easily generated in CAD, it takes several steps to model each surface element of these shapes with HML. In the case of the third product (shaver), the number of operations in CAD was significantly higher than the number of operations with HML. The reason for this is the fact that the third product contained several freeform surfaces, which took an enormous amount of steps in CAD, and only a couple of hand motions in the case of HML-based modeling.

Figure 6.25 a shows the number of geometric operations in the cases of CAD and HML-based modeling. In the cases of all three products and in case of HML-based modeling, the number of geometric operations was less, and actually, in the case of the second and third product significantly less than in the case of CAD modeling. Due to the fact that with HML-based modeling surfaces can be directly generated in three dimensions, the modeling process contains less steps than in the case of traditional CAD modeling.

Figure 6.25 b shows the number of manipulation operations in both CAD and HML-based modeling. It can be seen that for Product 1 the number of operations does not differ significantly, and in the cases of Product 2 and Product 3 HML-based modeling required less manipulation steps than CAD modeling. Especially Product 3 shows significant difference. The explanation for this is the fact that in most cases surfaces can be generated directly on the required spot in space or need only few steps of modification.

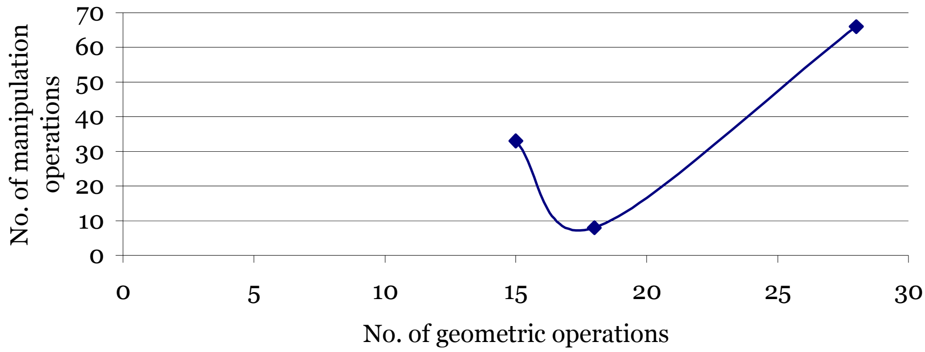


Figure 6.26 Changes in the number of manipulation operations according to the increase of geometric operations in case of HML-based modeling

Figure 6.26 shows the changes in the number of manipulation operations according to the increase of geometric operations in case of HML-based modeling, and Figure 6.27 the same changes in case of CAD modeling. While in case of CAD modeling the increase in the number of geometric operations causes an increase in the number of manipulation operation as well, the situation is different in case of HML-based modeling.

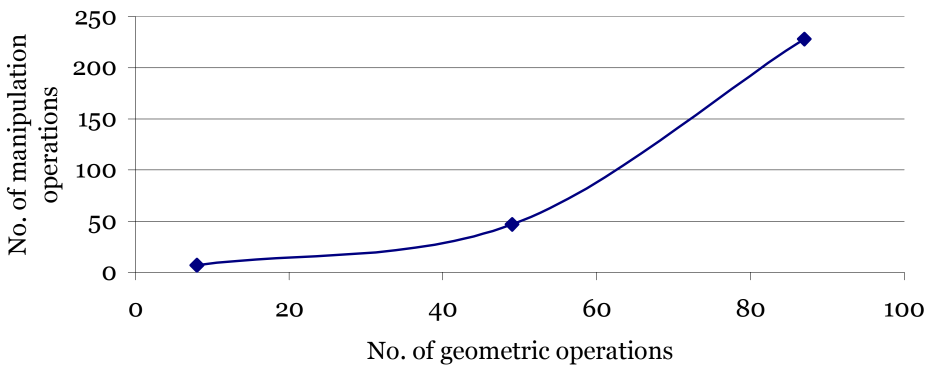


Figure 6.27 Changes in the number of manipulation operations according to the increase of geometric operations in case of CAD-based modeling

In this case, the number of manipulations mainly depends on the shape. While in case of CAD modeling a geometric operation always needs to be followed by a number of manipulation operations in order to set positions or sizes, in case of HML-based modeling the shapes can be generated directly in the spot where they are have to be and in their final size.

CONCLUSIONS AND FUTURE RESEARCH 7

In this thesis I dealt with the exciting and novel topic of hand motion based spatial shape design. I started this research with an extensive literature search, which pointed out that many directions are possible and all of them are interesting. On the other hand, the literature study clearly showed the undiscovered parts of this field of research as well. Many papers dealt with hand posture and motion recognition and their application for sign language recognition, for replacing simple mouse functionalities with hand motions or for some artistic applications. Some of them proposed methods for hand motion based shape design in two dimensions, and a few of them for spatial shape design sharing my dreams.

However, none of them discussed the information processing aspect and the actual added value of hand motions in spatial shape design. Therefore, I focused on establishing sound theories for hand motion processing and on the fast development of a proof-of-concept system, which was intended to be used for a comprehensive experimental part of research. In fact, this is the part which points out the merits and limits of hand motion based spatial shape design both from usability and utility points of view. In this chapter I discuss the main findings of this research and give some directions for future research.

7.1 Findings of the research

On the literature study and technology selection

A comprehensive literature study in the field of hand motion processing was done in order to explore the methods and tools of hand motion processing from three aspects, (i) the amount of the detected information from the hand motions, (ii) the way of transferring the information from hand motion detection device to the modeling system and (iii) the relationship of the hands and the detection device. This way I grouped the technologies into four groups: (i) direct processing of incomplete hand motions, (ii) direct processing of complete hand motions, (iii) indirect processing of incomplete hand motions and (iv) indirect processing of complete hand motions. I also tried to find contact and non-contact technologies for each of these categories. I concluded that group (i) does not provide enough information and group (iv) actually provides unnecessary redundant information. Technologies belonging to group (ii) and group (iii) showed the best potentials for further investigation. Many contact technologies were found and only one fully non-contact technology, which is based on processing of images recorded by multiple digital cameras.

As one of my goals was to answer the research questions regarding the usability and practical utility of hand motions in spatial shape design, and considering the fact the image-based recognition of hand motions is complex and time-consuming both in terms of research and in terms of computational power, I decided not to choose this option. I concluded that a passive optical tracking system with wireless markers fulfills both the ergonomics and technical requirements, since the markers are small and weightless and the tracking parameters are suitable in terms of the amount of detected data and tracking speed. After that I started an extensive market search to find the best-fitting equipment which was later used in the development of the proof-of-the concept system. However, I would like to mention that the ultimate goal is still the fully non-contact tracking of hand motions, which is an interesting topic for future research.

On usability of hand motions in conceptual shape design

I claim that a purposefully designed hand motion language is not only intuitive and enjoyable for designers but it also stimulates creativity. As creativity is a key feature of shape conceptualization, a hand motion based interface is extremely useful in the early phases of the design process. My first experiment focused on the usability of hand motions in spatial shape design. The results of the experiment clearly pointed out the advantages of hand motion-based shape conceptualization in a comparison with traditional commercial CAD software. I can confidently state that hand motion-based modeling takes less time, more intuitive and more attractive for designers than conventional CAD.

On hand motion interpretation

As far as the interpretation of hand motions is concerned, I concluded that words of a hand motion language can be interpreted based on investigating postural changes in the continuous hand motion. This way the useful parts of the continuous hand motion which carry modeling information can be found, and a wide variety of hand motions can be recognized by feeding a small number of parameters into a decision tree. Using this method, all 46 words of the Hand

Motion Language vocabulary can be identified in continuous hand motions in real-time. For modeling purposes, the Hand Motion Language comprises commands for surface generation, surface manipulation and procedural instructions. Both system control instructions and geometric entity descriptions can be generated in real time using passive optical tracking technology. A minimal set of detected landmarks enables interpretation of hand motion instructions as well as geometry generation. The kinematics of the human hand proved to be a useful source to complete missing information about hand motions as well as to improve the interpretation of hand motions.

On the practical utility of hand motions in shape conceptualization

Besides usability, which focused on human aspects of modeling, I also studied the utility of hand motions in shape conceptualization in terms of technical aspects through conducting experiments. I focused on three major issues of modeling, namely (i) fidelity of the generated geometric entities, (ii) accuracy of the surface manipulation operations and (iii) complexity of modeling. The results of the experiment show that an acceptable fidelity can be achieved even when generating regular shapes. Moreover, directions were given to designers in which spatial positions and orientations they can achieve the best quality surfaces. The results of the accuracy-oriented experiment were even surprising for me, because I did not expect such a high value in hand motion-based shape design. I can easily conclude that the achievable accuracy of shape manipulation by hand motions is more than sufficient. In the complexity-oriented experiment I compared the efficiency of HML-based modeling to traditional CAD modeling. It pointed out that especially in case of freeform shape generation HML-based modeling is significantly more efficient than CAD modeling. Due to the natural capability of hand motions that they can generate three-dimensional objects directly on the required spot in space an enormous amount of geometric and manipulation operations can be saved.

On modeling and visualization issues

Despite its proved utility, even the best designed hand motion interface can become pointless if connected to a conventional CAD system. Hand motions are mainly considered useful in conceptualization, because of its capability to support collaboration of designers and to externalize vague shape ideas. Current CAD systems do not facilitate these requirements. Therefore, I connected the hand motion-based interface to the Vague Discrete Interval Modeler to exploit the usefulness of vague information naturally produced by hand motions.

It has been also realized, that the currently widely used two-dimensional visualization methods are not applicable when controlling them by spatial hand motions. The use of the well-known stereo-glasses for three-dimensional visualization proved to be useful, however, the ultimate goal should be the usage of spatial visualization, which enables multiple designers to work on the same shape concept and to view the shape from different perspectives. With air-borne spatial visualization designers would be able to generate virtual surfaces exactly in the same space where the physical hand motions are performed. Recent research efforts focus on this topic at the Section of Computer Aided Design Engineering using a novel truly three-dimensional visualization method based on holography.

7.2 Directions for future research

It has been realized in the process of this research that there are a lot of open issues for future research. Without the need of completeness, the major ones are listed in this section.

Setup of a smart proactive design environment applying several sensing technologies is considered as a topic for the future. The intelligent design environment could be further equipped with other human sensing technologies for person and situation recognition. Such an environment is supposed to fulfill requirements from physical and information ergonomics and enable designers to use their natural communication skills when interacting with computer systems.

For hand motion detection, a method should be developed for optimum camera placement according to the positions of multiple designers. In this research I focused on the theory and implementation of hand motion processing and used an experimental setup of a motion detection environment. However, for a sound research contribution to this field a more analytical studo needs to be done regarding the placement of the cameras with a view to visibility issues. It must not be forgotten that besides the already mentioned occlusion problems, there is another issue, namely the occlusion of one designer by another when the system is used in collaboration.

The processing of the Hand Motion Language needs further improvement with a view to language and grammar processing and synchronization of modeling actions indicated by either one or multiple designers. This implies a perspective change from information technology point of view, and suggests the investigation of the usage of intelligent and deliberate software agents as programming elements. Agents are communicating with each other and decide on their actions based on the situation and the context.

Usefulness of multimodality was mentioned several times throughout this thesis. It is planned that the hand motion based design interface is combined with for instance verbal control and physical object scanning as input means for conceptual design systems, but other modalities are to be investigated as well. A little is known about the integration of different modalities and therefore a thorough study is needed regarding the optimal usage of different modalities in design.

As far as visualization is concerned, *the hand motion input will be connected to a truly three-dimensional air-borne visualization device* in the near future. The device is available in the laboratory of the CADE Section, but its capabilities are still very limited. An extensive analysis of the needs of hand motion-based design is necessary with a view to visualization. A continuous improvement of the device is expected by collaboration with the producing company.

The applicability of the hand motion based interface can be investigated in other application areas, such as the fields of medicine, marketing or different forms of art. Based on interviews and discussions with experts from several fields it turned out that the new interface might be useful in other applications as well, but a more detailed and focused investigation is needed.

REFERENCES

List of own publications

Varga, E., Horváth, I, Rusák, Z. and Broek, J. J. (2004a), „Hand Motion Processing in Applications: A Concise Survey and Analysis of Technologies", DESIGN2004, Dubrovnik, Croatia, May 18-21, pp. 811-816. (published)

Varga, E., Horváth, I., Rusák, Z., De Smit, A. and Broek, J. J. (2004b), „Survey and Investigation of Hand Motion Processing Technologies for Compliance with Shape Conceptualization", ASME Design Engineering Technical Conferences, CD-ROM CIE-57710, Salt Lake City, Utah, USA, Sept. 28 – Oct. 2. (published)

Varga, E., Horváth, I., Rusák, Z. and Antonya, C. (2005a) „Augmenting Detected Hand Motion Data by Using an Active Hand Model for Virtual Surface Design", 5th International Conference of Ph.D. Students on Engineering Sciences, Miskolc, Hungary, Aug. 14-20, pp. 195-200. (published)

Varga, E., Horváth, I., Rusák, Z. and Verlinden, J. (2005b), "On the Framework of Information Processing in a Hand Motion Based Shape Conceptualization System", ASME Design Engineering Technical Conferences, CD-ROM DETC2005-84929, Long Beach, California, USA, Sept. 24-28 (published)

Varga, E., Rusák, Z. and Horváth, I. (2006a), "Interpretation of Hand Motions for Shape Conceptualization in Virtual Design Environments", Tools and Methods of Competitive Engineering, Ljubljana, Slovenia, Apr. 18-22, pp. 309-322. (published)

Varga, E., Horváth, I., and Rusák, Z. (2006b), "An Effective Trajectory Segmentation and Processing Method for Hand Motion Based Shape Conceptualization", ASME Design Engineering Technical Conferences, CD-ROM DETC2006-99342, Philadelphia, Pennsylvania, USA, Sept. 10-13 (published)

Varga, E., Horváth, I. and Rusák, Z. (2007a), "Analysis of the Usability of Hand Motion Language in Shape Conceptualization", International Conference on Engineering Design, CD-ROM 504, Paris, France, Aug. 28-31 (published)

Varga, E., Horváth, I., Rusák, Z. (2007b), "Testing the Efficiency of Hand Motion Language Based Model Building from Geometric Information Processing Point of View", ASME Design Engineering Technical Conferences, CD-ROM DETC2007-35324, Las Vegas, Nevada, USA, Sept. 4-7 (published)

Varga, E., Rusák, Z. and Horváth, I. (2007c), "Using Hand Motions in Conceptual Shape Design: Theories, Methods and Tools", 2nd Advanced Study Institute on Product Engineering: Tools and Methods based on Virtual Reality, Chania, Crete, 30 May – 6 June, 2007 (accepted)

Other references

Abe, K., Saito, H. and Ozawa, S. (2000), "3-D Drawing System via Hand Motion Recognition from Two Cameras", IEEE International Conference on Systems, Man, and Cybernetics, 8-11 Oct., Nashville, Tennessee, USA, pp 840-845.

Aggarwal, J. K. and Cai, Q. (1997), "Human Motion Analysis: A Review", Nonrigid and Articulated Motion Workshop, 16 June, San Juan, Puerto Rico, pp. 90-102.

Aggarwal, J. K. and Cai, Q., (1999), "Human Motion Analysis: A Review", Computer Vision and Image Understanding, Vol. 73, No. 3, pp. 428 - 440.

Ahmad. S. (1995), "A Usable Real-time 3D Hand Tracker", in Proceedings of 28th Asilomar Conference on Signals, Systems and Computers, IEEE Computer Society Press, pp. 1257 - 1261.

Albrecht, I., Haber, J. and Seidel, H. P. (2003), "Construction and Animation of Anatomically Based Human Hand Models", Eurographics/SIGGRAPH Symposium on Computer Animation, 26-27 July, San Diego, California, USA, pp. 98-109.

Arangarasan, R. and Gadh, R., (2000), "Geometric Modeling and Collaborative Design in a Multi-modal Multi-sensory Virtual Environment", ASME Design Engineering Technical Conferences, Baltimore, Maryland, Sep. 10-13.

Arangarasan, R. and Phillips, G. N. Jr. (2002), “Modular Approach of Multimodal Integration in a Virtual Environment”, Fourth IEEE International Conference on Multimodal Interfaces, 14-16 October, Pittsburg, PA, USA, pp. 331-336.

Arciszewski, T., Michalski, R. S. and Wnek, J. (1995), “Constructive Induction: The Key to Design Creativity”, International Round-Table Conference on Computational Models of Creative Design, Dec. 3-7, Heron Island, Queensland, Australia, pp. 1-30.

Arendsen, J., van Doorn, A.J. and de Ridder, H. (2007). “When and how well do people see the onset of gestures?”, Gesture (in press), pp. 1-35.

Avouris, N. M. (2001), ”An Introduction to Software Usability”, Workshop on Software Usability, 8th Panhellenic Conference on Informatics, Vol 2, Nicosia, November, Athens, pp. 514-522.

Badler, N. I., Barsky, B. A. and Zeltzer, D. (1991), “Making Them Move: Mechanics, Control, and Animation of Articulated Figures”, Morgan Kaufmann Series in Computer Graphics and Geometric Modeling, Morgan Kaufmann Publishers Inc., ISBN 1-55860-106-6, pp. 1-348.

Bevan, N. and Macleod, M. (1994), “Usability Measurement in Context”, In Journal of Behavior and Information Technology, Vol. 13, pp. 132-145.

Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A. and Cruz-Neira, C. (2001), “VR Juggler: A Virtual Platform for Virtual Reality Application Development”, IEEE Virtual Reality Conference, Yokohama, Japan, March, pp.89-96.

Biermann, P., Jung, B, Latoschik, M and Wachsmuth, I. (2002), “Virtuelle Werkstatt: A Platform for Multimodal Assembly in VR”, Fourth Virtual Reality International Conference, Laval, France, pp. 53-62.

Bimber, O. (2004), “Combining Optical Holograms with Interactive Computer Graphics”, Computer, IEEE Computer Society Press, January, Vol. 37, Issue 1, pp.85-91.

Bordegoni, M. and Cugini, U. (2007), “Designer-Centered Haptic Interfaces for Shape Modeling”, 12th International Conference on Human-Computer Interaction, 22-27 July, Beijing, China, pp. 441-450.

Borenstein, E. and Ullman, S. (2002), “Class Specific Top-Down Segmentation”, 7th European Conference on Computer Vision, 27 May – 2 June, Copenhagen, Denmark, pp. 109-124.

Butnaru, T., Antonya, C. and Talaba, D. (2007), “A Wired Haptic System for Multimodal VR Interaction”, Proceedings of Advanced Study Institute on Product Engineering Tools and Methods Based on Virtual Reality, Transylvania University Press, Brasov, pp. 169-180.

Carroll, J. M. (1993), “Creating a design science of human-computer interaction”, Interacting with Computers, Vol. 5, No. 1, pp. 3-12.

Catledge, L. and Potts, C. (1996), "Collaboration During Conceptual Design," 2nd International Conference on Requirements Engineering, Colorado Springs, Colorado, USA, pp. 182—189.

Chen, Y.-Q. and Huang, T. S. (2000), “Motion Segmentation and Tracking by Embedding Global Model within a Contextual Relaxation Scheme”, IAPR Workshop on Machine Vision and Applications, November 28 - 30, Tokyo, Japan, pp. 455 - 458.

Cheng, S. W. and Dey, T. K. (1999), “Improved Construction of Delaunay Based Contour Surfaces”, ACM Symposium on Solid Modeling and Applications, 8-11 June, Ann Arbor, Michigan, USA, pp. 322-323.

Chu, C. P., Dani, T. H. and Gadh, R. (1997), “Multi-sensory User Interface for a Virtual-reality-based Computer-aided Design System”, Computer-Aided Design, Vol. 29, Nr. 10, pp. 709-725.

Clark, J. H. (1976), “Designing Surfaces in 3-D”, Communications of the ACM, Vol. 19, No. 8, pp. 454-460.

Cont, A., Coduys, T. and Henry, C. (2004), “Real-Time Gesture Mapping in Pd Environment using Neural Networks”, Conference on New Interfaces for Musical Expression, 3-5 June, Hamamatsu, Japan, pp. 39-42.

Corradini, A. and Cohen, P. R. (2002), “Multimodal Speech-Gesture Interface for Handfree Painting on a Virtual Paper Using Partial Recurrent Neural Networks”, International Joint Conference on Artificial Neural Networks, Honolulu, HI, USA, Vol. III, pp. 2293-2298.

Dani, T. H. and Gadh, R. (1997), “Creation of Concept Shape Designs via a Virtual Reality Interface”, Computer-Aided Design, Vol. 29, No. 8, pp. 555-563.

De Antonio, A., Imbert, R., Ramirez, J. and Ferre, X. (2004), “Usability Issues in the Design of an Intuitive Interface for Planning and Simulating Maintenance Interventions Using a Virtual Environment”, Virtual Reality, Vol. 7, pp. 212-221.

Deawele, G., Devernay, F. and Horaud, R. (2004), "Hand Motion from 3D Point Trajectories and a Smooth Surface Model", 8th European Conference on Computer Vision, 11-14 May, Prague, Czech Republic

Delamarre, Q. and Faugeras, O. (1999), "Articulated Models and Multi-view Tracking with Silhouettes", International Conference on Computer Vision, Corfu, Greece, September 1999, pp. 716 – 721.

Dix, A., Finlay, J., Abowd, G. and Beale, R. (2001), "Human Computer Interaction", 2nd Edition, Pearson Education Limited, Harlon

Djenidi, H., Tadj, C., Ramdane-Cherif, A. and Levy, N. (2002), "Dynamic Based Multi-Agent Architecture for Multimedia Multimodal Dialogs", IEEE Workshop on Knowledge Media Networking, 10-12 July, Kyoto, Japan, pp. 107.

Dorozhkin, D. V. and Vance, J. M. (2002), "Implementing Speech Recognition in Virtual Reality", ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Montreal, Sep. 29 – Oct. 2, CD-ROM, DETC2002/CIE-34390, pp. 1 - 5.

Dourish, P. (2001), "Where the Action Is: the Foundations of Embodied Interaction", MIT Press, ISBN 0-262-04196-0, pp. 1-256.

Eisenstein, J., Ghandeharizadeh, S., Golubchik, L., Shahabi, C., Yan, D. and Zimmermann, R. (2003), "Device Independence and Extensibility in Gesture Recognition", IEEE Virtual Reality 2003, March 22 - 26, Los Angeles, CA, USA, pp. 207.

Elkoura, G. and Singh, K. (2003), "Handrix: Animating the Human Hand", Eurographics/SIGGRAPH Symposium on Computer Animation, 26-27 July, San Diego, California, USA, pp. 110-119.

Estépar, R. S. J., Martín-Fernández, M., Alberola-López, C., Ellsemere, J. and Westin, C. F. (2003), „Freehand Ultrasound Reconstruction based on ROI Prior Modeling and Normalized Convolution", 6th International Conference on Medical Image Computing and Computer-Assisted Intervention, LNCS 2879, November 15-18, 2003. Montreal, Canada, pp. 382-390.

Ferre, X., Juristo, N., Windl, H. and Constantine, L. (2001), "Usability Basics for Software Developers", In Journal of IEEE Software, Vol. 18 (1), pp. 22-29.

Fitzpatrick, R. and Higgins, C. (1998), "Usable Software and its Attributes: A Synthesis of Software Quality, European Community Law and Human-Computer Interaction", People and Computers XIII. Human-Computer Interaction'98 Conference, Springer, London, UK, pp. 3-21.

Flippo, F., Krebs, A. and Marsic, I. (2003), "A Framework for Rapid Development of Multimodal Interfaces", International Conference on Multimodal Interfaces, 5-7 November, Vancouver, British Columbia, Canada, pp. 109-116.

Galopp, N., Tekin, S., Otaduy, M. A., Gross, M. and Lin, M. C. (2007), "Interactive Haptic Rendering of High-Resolution Deformable Objects", Virtual Reality, HCII 2007, LNCS 4563, pp. 215-223.

Gavrila, D. M. (1999), "The Visual Analysis of Human Movement: A Survey", Computer Vision and Image Understanding, Vol. 73, No. 1, pp. 82 - 98.

Gerritsen, B. H. M. (2001), "Using Weighted Alpha Complexes in Subsurface Modelling", Ph.D. Thesis, Delft University Press, Delft, pp. 1-224.

Gnanayutham, P. and George, J. (2007), "Inclusive Design for Brain Body Interfaces", Augmented Cognition, HCII 2007, LNCS 4565, pp. 103-112.

Grasso, M. A., Ebert, D. S. and Finin, T. W. (1998), "The Integrality of Speech in Multimodal Interfaces", ACM Transactions on Computer-Human Interaction, Vol. 5, No. 4, December, pp. 303-325.

Gribnau, M. W. and Hennessey, J. M. (1998), "Comparing Single- and Two-Handed 3D Input for a 3D Object Assembly Task", CHI'98 Conference on Human Factors in Computing Systems,18-23 April, Los Angeles, USA, pp. 233-234.

Grossman, T., Wigdor, D. and Balakrishnan, R. (2004), "Multi-finger gestural interaction with 3D volumetric displays", ACM Symposium on User Interface Software and Technology, 24-27 October, Santa Fe, NM, USA, pp.61-70.

Halle, M. (1997), "Autostereoscopic Displays and Computer Graphics", in Computer Graphics, ACM SIGGRAPH, 31 (2), pp. 58-62.

Hand, C. (1997), "A Survey of 3D Interaction Techniques", Computer Graphics Forum, Vol. 16, No. 5, pp. 269-281.

Heap, A. J. and Hogg, D. (1996), “3D Deformable Hand Models, Progress in Gestural Interaction”, Gesture Workshop '96, University of York, UK, March, pp. 131 – 139.

Heap, T. and Hogg, D. (1996), “Towards 3D Hand Tracking using a Deformable Model”, Second International Conference on Automatic Face and Gesture Recognition, 13-16 October, Killington, USA, pp. 140-145.

Heap, J. and Samaria, F. (1995), “Real-Time Hand Tracking and Gesture Recognition using Smart Snakes”, Interface to Real and Virtual Worlds, Montpellier, June, Olivetti Research Limited Tech Report 95.1, pp. 1-13.

Holden, E. J. and Owens, R. (2001), “Visual Sign Language Recognition”, Multi-Image Analysis, LNCS 2032, pp. 270-287.

Horváth, I. and van der Vegte, W. (2003a), „Nucleus-Based Product Conceptualization – Part 1: Principles and Formalization”, International Conference on Engineering Design, Aug. 19-21, Stockholm, Sweden, pp. 1-10.

Horváth, I. (2006), “On the Differences Between Research in Design Context and Design Inclusive Research”, International Design Research Symposium, , Nov. 10-11, Seoul, Korea, pp. 1-16.

Horváth, I. and Rusák, Z. (2001), „Collaborative Shape Conceptualization in Virtual Design Environments”, Communications of the ACM, Vol. 44, No. 12, pp. 59-63.

Horváth, I., Tromp, N. and Daalhuizen, J. (2003b) “Comprehending a Hand Motion Language in Shape Conceptualization”, ASME Design Engineering Technical Conferences, 2-6 September, Chicago, IL. CD-ROM CIE-48286, pp. 1-15.

Horváth, I., Vergeest, J. S. M. and Rusák, Z. (2002), “Six Ingredients of Collaborative Virtual Design Environments”, DESIGN 2002, May 14-17, Dubrovnik, Croatia, pp. 67-74.

Horváth, I. and Opiyo, E. (2007), “Qualitative Analysis of the Affordances of Three-Dimensional Imaging Systems with a view to Conceptual Shape Design”, ASME Computers in Engineering, Sep. 4-7, Las Vegas, Nevada, USA

Huang, T. S. (1990), “Modeling, Analysis, and Visualization of Non-rigid Object Motion”, 10th International Conference on Pattern Recognition, 16-21 June, Atlantic City, New Jersey, USA, pp. 361–364.

Hummels, C. and Overbeeke, K. (1999), “The Expressive Power of Gestures: Capturing Scent in a Spatial Shape”, in Lecture Notes in Artificial Intelligence, No. 1739, pp. 23-26.

Hummels, C. and Stappers, P. J. (1998), “Meaningful Gestures for Human-Computer Interaction: Beyond Hand Gestures”, Third IEEE International Conference on Automatic Face & Gesture Recognition, April 14-16, Nara, Japan, pp. 591-596.

Hummels, C. (2000), “Gestural Design Tools: Prototypes, Experiments and Scenarios”, Ph.D. Thesis, Delft University of Technology, Delft, pp. 1 – 235.

Iannizzotto, G., Villari, M. and Vita, L. (2001), “Hand Tracking for Human-Computer Interaction with Graylevel VisualGlove: Turning back to the simple way”, Workshop on Perceptual User Interfaces, November 15-16, Orlando, Florida, USA

Imagawa, K., Matsuo H., Taniguchi, R. Arita, D., Lu, S. and Igi, S. (2000), “Recognition of Local Features for Camera-Based Sign Language Recognition System”, International Conference on Pattern Recognition, Vol. 4, September 3 - 8, Barcelona, Spain, pp. 849 - 853.

Jayaram, S., Gowda, S. and Jayaram, U. (2004), "Architectures for Collaborative Virtual Prototyping", International Journal of Agile Manufacturing, Vol. 7, Issue 2.

Juhász, I. (1995), “Számítógépi Geometria és Grafika”, Miskolci Egyetemi Kiadó

Kambhamettu, C., Goldgof, D. B., Terzopoulos, D. and Huang, T. S. (1994), “Non-rigid Motion Analysis”, Handbook of Pattern Recognition and Image Processing: Computer Vision, Vol. 2, pp. 405-430.

Keefe, D. F., Feliz, D. A., Moscovich, T., Laidlaw, D. H. and LaViola, J. J. (2001), “CavePainting: A Fully Immersive 3D Artistic Medium and Interactive Experience”, Symposium on Interactive 3D Graphics, March 19-21, Research Triangle Park, USA, pp. 85-93.

Kettebekov, S. and Sharma, R. (2001), “Towards Natural Gesture/Speech Control of a Large Display”, 8th International Conference on Engineering for Human-Computer Interaction, 11-13 May, Toronto, Canada, pp. 221-234.

Kim, I. and Chien, S., (2001), “Analysis of 3D Hand Trajectory Gestures Using Stroke-Based Composite Hidden Markov Models”, Applied Intelligence 15, pp. 131-143.

Ko, B. K. and Yang, H. S. (1997), “Finger Mouse and Gesture Recognition System as a new Human Computer Interface”, Computers & Graphics, Vol. 21, No. 5, pp. 555-561.

Kölsch, M., Beall, A. and Turk, M. (2003), "Postural comfort zone for reaching gestures," Human Factors and Ergonomics Society Annual Meeting, 13-17 October, Denver, Colorado, USA, pp. 1-5.

Konishi, S., Yuille, A., Coughlan, J. and Zhu, S. C. (2003), "Statistical edge detection: Learning and Evaluating Edge Cues", Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 1, pp. 57-74.

Krause, F.-L., Göbel, M., Wesche, G. and Biahman, T. (2004), "A Three-Stage Conceptual Design Process Using Virtual Environments", Winter School of Computer Graphics, Feb. 2-6, Plzen, Czech Republic, pp. 1-4.

Kuczogi, Gy., Horváth, I. and Rusák, Z. (2002), "Verbal Interface for Vague Discrete Shape Modeler", Tools and Methods of Competitive Engineering, April 22-26, Wuhan, China, pp. 663-674.

Kuczogi, Gy., Horváth, I. Vergeest, J. S. M. and Rusák, Z. (2001), "Communication Model for the User Interface of a Shape Conceptualization System", ASME Design Engineering Technical Conferences, 9-12 September, New York, DETC2001/CIE-21300, pp. 1 - 14.

La Viola, J. J. (1999), "A Survey of Hand Posture and Gesture Recognition Techniques and Technologies", Research Report, Brown University, Department of Computer Science, CS-99-11, pp. 1 - 80.

Lamar, M. V. (2001), "Hand Gesture Recognition using T-CombNET – A Neural Network Model dedicated to Temporal Information Processing", Ph.D. thesis, Nagoya Institute of Technology, Japan

Lamar, M. V., Bhuyian, Md. S. and Iwata, A. (2000), "T-CombNET – A Neural Network Dedicated to Hand Gesture Recognition", Lecture Notes in Computer Science 1811, pp. 735-763.

Lathuiliere, F. and Herve, J. Y. (2000), "Visual Tracking of Hand Posture with Occlusion Handling", International Conference on Pattern Recognition, Sep. 3-8, Barcelona, Spain, pp. 1140 – 1145.

Latoschik, M. E. (2001), "A Gesture Processing Framework for Multimodal Interaction in Virtual Reality", 1st International Conference on Computer Graphics, Virtual Reality and Visualization, Camps Bay, Cape Town, South-Africa, pp.95-100.

Lee, Ch. and Xu, Y. S. (1996), "Online, Interactive Learning of Gestures for Human/Robot Interfaces", IEEE International Conference on Robotics and Automation, Minneapolis, MN, Vol. 4, pp. 2982 - 2987.

Lee, J. and Kunii, T. L. (1993), "Constraint-Based Hand Animation", Models and Techniques in Computer Animation, Springer-Verlag, Tokyo, pp. 110 – 127.

Lee, J. and Kunii, T. L. (1995), "Model-Based Analysis of Hand Posture", IEEE Computer Graphics and Applications, Vol. 15, No. 5, pp. 77 - 86.

Leganchuk, A., Zhai, S. and Buxton, W. (1998), "Manual and Cognitive Benefits of Two-Handed Input: An Experimental Study", ACM Transactions on Computer-Human Interaction, Vol. 5, No. 4, Dec., pp. 326-359.

Leidner, J. L. (2003), "Current Issues in Software Engineering for Natural Language Processing", Workshop on Software Engineering and Architecture of Language Technology Systems, May 27 – June 1, Edmonton, Alberta, Canada pp. 45—50.

Li, F. W. B., Lau, R. W. H. and Ng, F. F. C. (2003), "VSculpt: A Distributed Virtual Sculpting Environment for Collaborative Design", IEEE Transactions on Multimedia, Vol. 5, No. 4, pp. 570-580.

Liang, J. and Green, M. (1993), "Geometric Modeling Using Six Degrees of Freedom Input Devices", 3rd International Conference on CAD and Computer Graphics, August 23-26, Beijing, China, pp. 217-222.

Liang, R.-H. and Ouhyoung, M. (1998), "A Real-Time Continuous Gesture Recognition System for Sign Language", April 14 – 16, Nara, Japan, Third International Conference on Face & Gesture Recognition, pp. 558-563.

Lim, S. W., Lee, B. S. and Duffy, A. H. (2001), "Incremental Modeling of Ambiguous Geometric Ideas (I-MAGI): Representation and Maintenance of Vague Geometry", Artificial Intelligence in Engineering, Vol. 15, pp. 93 - 108.

Lin, I. (2003), "Reliable Extraction of Realistic 3D Facial Animation Parameters from Mirror-reflected Multi-view video clips", Ph.D. dissertation, National Taiwan University, Taiwan

Lin, J., Wu, Y. and Huang, T. (2000), "Modeling the Constraints of Human Hand Motion", Workshop on Human Motion, 7-8 December, Austin, Texas, pp. 121-126.

Lucente, M. (1997), "Interactive Three-dimensional Holographic Displays: Seeing the Future in Depth", Computer Graphics, May, Vol. 31, No. 2, pp.63-66.

Ma, L., Lau, R. W. H., Feng, J., Peng, Q. and Wong, J. (1997), “Surface Deformation Using the Sensor Glove”, ACM Virtual Reality Software and Technology, Sep. 15-17, Lausanne, Switzerland

Matsumiya, M., Takemura, H. and Yokoya, N. (2000), “An Immersive Modeling System for 3D Free-form Design Using Implicit Surfaces”, ACM Symposium on Virtual Reality Software and Technology, October 22-25, Seoul, Korea, pp. 67-74.

Moccozet, L. (1996), “Hand Modeling and Animation for Virtual Humans”, Ph.D. Thesis, University of Geneva, Geneva, Switzerland

Moeslund, T. B. and Granum, E. (2001), “A Survey of Computer Vision-based Human Motion Capture, Computer Vision and Image Understanding, Vol. 81, No. 3, pp. 231 - 268.

Murakami, T. and Nakajima, N. (2000), “DO-IT: Deformable Object as Input Tool for 3-D Geometric Operation”, Computer-Aided Design, Vol. 32, No.1, pp. 5-16.

Nam, Y., Thalmann, D. and Wohn, K. (1999) “A Hybrid Framework for Modeling Comprehensible Human Gesture”, International Conference on Computational Intelligence for Modelling, Control and Automation, Vienna, Austria, pp. 220-225.

Nam, Y. and Wohn, K. (1996), “Recognition of Space-Time Hand-Gestures using Hidden Markov Model”, ACM Symposium on Virtual Reality Software and Technology, July 1-4, Hong Kong, pp. 51 - 58.

Nishino, H., Utsumiya, K. and Korida, K. (1998), “3D Object Modeling Using Spatial and Pictographic Gestures”, Virtual Reality Software and Technology, Taipei, Nov. 2-5, Taiwan, pp. 51-58.

Nishino, H., Utsumiya, K., Kuraoka, D. and Yoshioka, K. (1997), “Interactive Two-Handed Gesture Interface in 3D Virtual Environments”, ACM Virtual Reality Software and Technology, Sep. 15-17, Lausanne, Switzerland, pp. 1-8.

Ogawara, K., Hashimoto, K., Takamatsu, J., Ikeuchi, K. (2003), “Grasp Recognition using a 3D Articulated Model and Infrared Images”, IEEE/RSJ International Conference on Intelligent Robots and Systems, October 27-31, Las Vegas, USA

Opiyo, E. Z., Horváth, I., Rusák, Z., van der Vegte, W., Varga, E. and Csongvai, R. (2007), “Developing a Proactive Reality Environment for Conceptual Design”, ASME Computers in Engineering, Sep. 4-7, Las Vegas, Nevada, USA

Opiyo, E. and Horváth, I. (2006), „Analysis of the Relevance of the Emerging Visualization Technologies and Systems to Spatial Shape Design 6th International Symposium on Tools and Methods of Competitive Engineering, Apr 18-22, Ljubljana, Slovenia, pp. 335-348.

Oviatt, S. and Cohen, P. (2000), “Multimodal Interfaces that Process What Comes Naturally”, Communications of the ACM, March, Vol. 43, No. 3, pp. 45-53.

Pavlovic, V., Sharma, R. and Huang, T. (1997), “Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.19, No.7, July, pp. 677 - 695.

Plesniak, W., Pappu, R. S. and Benton, S. A. (2003), “Haptic Holography: A Primitive Computational Plastic”, in Proceedings of the IEEE, Vol. 91, No. 9, pp. 1443-1456.

Rauterberg M. (2006), “HCI as an engineering discipline: to be or not to be!?”, African Journal of Information and Communication Technology, Vol. 2, No. 4, pp. 163-184.

Rehg, J. M. and Kanade, T. (1994), “DigitEyes: Vision-Based Hand Tracking for Human-Computer Interaction”, IEEE Workshop on Motion of Non-Rigid and Articulated Objects, Austin, Texas, November, pp. 16-22.

Rieger, T. (2003), “Avatar Gestures”, 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Feb. 3-7, Plzen-Bory, Czech Republic, Journal of WSCG, Vol. 11, No. 2, pp. 379-386.

Rijpkema, H. and Girard, M. (1991), “Computer Animation of Knowledge-Based Human Grasping”, Computer Graphics, Vol. 25, No. 4, pp. 339 - 348.

Rusák, Z. (2003), “Vague Discrete Interval Modeling for Product Conceptualization in Collaborative Virtual Design Environments”, Ph.D. Thesis, Millpress, Rotterdam, pp. 1-159.

Rusák, Z., Horváth, I. and Varga, E. (2006), “Generation and Instantiation of Vague Shapes by Hand Motions to Support Shape Conceptualization”, 12th International Conference on Machine Design and Production, 5-8 September, Kusadasi, Turkey

Russ, G., Sallans, B. and Hareter, H. (2005), “Semantic Based Information Fusion in a Multimodal Interface”, International Conference on Human-Computer Interaction, 20-23 June, Las Vegas, Nevada, USA, pp 94-100.

Schkolne, S., Pruett, M. and Schröder, P. (2001), “Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools”, Conference on Human Factors in Computer Systems, 31 March – 5 April, Seattle, Washington, USA, pp. 261-268.

Schlattman, M., Kahlesz, F., Sarlette and R., Klein, R. (2007), “Markerless 4 Gestures 6 DOF Real-Time Visual Tracking ”, Computer Graphics Forum, Vol. 26, No. 3, pp. 467-476.

Segen, J. and Kumar, S. (1998), “Video-Based Gesture Interface to Interactive Movies”, ACM and ATR Workshop on Technologies for Interactive Movies, Sep 13, Bristol, UK, pp. 39-42.

Segen, J. and Kumar, S. (1999), “Shadow Gestures: 3D Hand Pose Estimation using a Single Camera”, IEEE International Conference on Computer Vision and Pattern Recognition, June 23-25, Fort Collins, CO, USA, pp. 479-485.

Seth, A., Su, H. J. and Vance, J. M. (2006), “SHARP: System for Haptic Assembly & Realistic Prototyping”, ASME International Design Engineering Technical Conference, Sep. 10-13, Philadelphia, PA, USA

Shaw, C. and Green, M. (1994), “Two-Handed Polygonal Surface Design”, 7th ACM Symposium on User Interface Software and Technology, November 2-4, Marina del Rey, USA, pp. 205-212.

Shimada, N., Shirai, Y. and Kuno, Y. (1995), “Hand Gesture Recognition Using Computer Vision Based on Model-Matching Method”, 6th International Conference on Human Computer Interaction, Elsevier Publisher, pp. 11 - 16.

Song, Y., Vergeest, J. S. M. and Bronsvoort, W. F. (2005), “Fitting and Manipulating Freeform Shapes using Templates”, Journal of Computing and Information Science in Engineering, Vol. 5, No. 2, pp. 86-94.

Sowa, T. and Wachsmuth, I. (2002), “Interpretation of Shape-Related Iconic Gestures in Virtual Environments”, Gesture and Sign Language in Human-Computer Interaction, Springer Verlag: Berlin, pp. 2133-2146.

Stefani, O. and Rauschenbach, J. (2003), “3D Input Devices and Interaction Concepts for Optical Tracking in Immersive Environments”, 7th International Workshop on Immersive Projection Technology, 9th Eurographics Workshop on Virtual Environments, 22-23 May, Zurich, Switzerland, pp. 317-318.

Stenger, B., Mendonca, P. R. S. and Cipolla, R. (2001), “Model-Based 3D Hand Tracking of an Articulated Hand”, Conference on Computer Vision and Pattern Recognition, December, Kauai, USA, Vol. II, pages 310-315.

Sturman, D. J. (1992), “Whole-hand Input”, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, pp. 1-233.

Su, S. A. and Furuta, R. (1994), “A Logical Hand Device in Virtual Environments”, ACM Symposium on Virtual Reality Software and Technology, Aug. 23-26, Singapore, pp. 33 - 42.

Tamura, S. and Kawasaki, S. (1988), “Recognition of Sign Language Motion Images”, Pattern Recognition, Elsevier, Vol. 21, No. 4, pp. 343 - 353.

Tobely, T. E., Yoshiki, Y., Tsuda, R., Tsuruta, N. and Amamiy, M. (2000), “Dynamic Hand Gesture Recognition Based on Randomized Self-Organizing Map Algorithm”, International Conference on Algorithmic Learning Theory, Springer-Verlag, Berlin, pp. 252 - 263.

Triesch, J. and von der Malsburg, Ch. (1996), “Robust Classification of Hand Postures Against Complex Backgrounds”, 2nd International Conference on Automatic Face and Gesture Recognition, Killington, USA, October 14-16, IEEE Computer Society Press, pp. 170 - 175.

Truong, K. P., van Leeuwen, D. A. and Neerincx, M. A. (2007), “Unobtrusive Multimodal Emotion Detection in Adaptive Interfaces: Speech and Facial Expressions”, 12th International Conference on Human-Computer Interaction, July 22-27, Beijing, China, Lecture Notes in Computer Science 4565, pp. 354-363.

Utsumi, A. and Ohya, J. (1999), “Multiple-Hand-Gesture Tracking using Multiple Cameras” in Proceedings of Computer Vision and Pattern Recognition, June 23-25, Ft. Collins, CO, USA, pp. 473-478.

Van der Vegte W.F. and Rusák, Z. (2007), “Hybrid Simulation of Use Processes with Scenario Structures and Resource-Integrated Models”, ASME International Conference on Computers in Engineering, Las Vegas, Nevada, USA

Van Orden, K. F. and Broyles, J. W. (2000), “Visuospatial Task Performance as a Function of Two- and Three-Dimensional Presentation Techniques”, Displays, Elsevier, Vol. 21, No. 1, pp. 17-24.

Várady, T., Martin, R. R. and Cox, J. (1997), „Reverse engineering of geometric models – an Introduction”, Computer-Aided Design, Vol. 29, No. 4, pp. 255-268.

Vergeest, J. S. M., Spanjaard, S., Horváth, I. and Jelier, J. J. O. (2001), "Fitting Freeform Shape Patterns to Scanned 3D Objects", Transactions of the ASME, Vol. 1, September, pp. 218-224.

Verlinden, J., Horváth, I. and Edelenbos, E. (2006), "Treatise of Technologies for Interactive Augmented Prototyping", International Conference on Tools and Methods of Competitive Engineering, April 18-22, Ljubljana, Slovenia, pp. 523-536.

Wagner, S. M., Nusbaum, H. and Goldin-Meadow, S. (2004), "Probing the Mental Representation of Gesture: Is Handwriting Spatial?", Journal of Memory and Language, Elsevier, Vol. 50, pp. 395-407.

Watson, R. (1993), "A Survey of Gesture Recognition Techniques", Technical Report, TCD-CS-93-11, Trinity College, Dublin, pp. 1 - 23.

Weimer, D. and Ganapathy, S. K. (1989), "A Synthetic Visual Environment with Hand Gesturing and Voice Input", Human Factors in Computing Systems, Apr. 30-June 4, Austin, Texas, USA, pp. 235-240.

Wesche, G. and Seidel, H. P. (2001), "FreeDrawer - A Free-Form Sketching System on the Responsive Workbench", ACM Symposium on Virtual Reality Software and Technology, November 15-17, Baniff, Alberta, Canada, pp. 167-174.

Westeyn, T., Brashear, H., Atrash, A. and Starner, T. (2003), "Georgia Tech Gesture Toolkit: Supporting Experiments in Gesture Recognition", International Conference on Multimodal Interfaces, Vancouver, Canada

Wolf, J. C. and Bugmann, G. (2006), "Linking Speech and Gesture in Multimodal Instruction Systems", 15th International IEEE Symposium on Robot and Human Interactive Communication, Sep. 6-8, Hatfield, UK, pp. 141-144.

Wong, J. P. Y., Lau, R. W. H. and Ma, L. (1998), "Virtual 3D Sculpturing with a Parametric Hand Surface", Computer Graphics International, June 22-26, Hannover, Germany, pp. 178-186.

Wu, Y. and Huang, T. S. (1999), "Capturing Articulated Human Hand Motion: A Divide-and-Conquer Approach", 7th International Conference on Computer Vision, Sep. 20-27, Corfu, Greece, pp. 606 - 611.

Wu, Y. and Huang, T. S. (1999), "Vision-Based Gesture Recognition: A Review", International Gesture Workshop, March 17-19, Gif-sur-Yvette, France, pp. 103-115.

Wu, Y. and Huang, T. S. (2000), "View-independent Recognition of Hand Postures", IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, SC, Vol. II, pp. 88-94.

Wu, Y. and Huang, T. S. (2001), "Hand Modeling, Analysis, and Recognition", IEEE Signal Processing Magazine, Special Issue on Immersive Interactive Technology, Vol. 18, No. 3, May, pp. 51-60.

Xu, S., Fok, S. C. and Tor, S. B. (2000), "Rapid Creation and Direct Manipulation of Free-form Curves Using Data Glove", Conference on Engineering Design and Automation, July 30-Aug. 2, Orlando, Florida, USA, pp. 954-959.

Yu, H., Paulus, D. and Niemann, H. (2000), "Dynamic Gesture Analysis and Tracking Based on Dominant Motion Estimation and Kalman Filter", Deutsche Arbeitsgemeinschaft für Mustererkennung Symposium, Sep. 13-15, Kiel, Germany, pp. 397 - 404.

Zhang, Y., Rohling, R. and Pai, D. K. (2002), "Direct Surface Extraction from 3D Freehand Ultrasound Images", IEEE Conference on Visualization, Oct. 27-Nov. 1, Boston, MA, USA, pp. 45-52.

Zheng, J. M., Chan, K. W. and Gibson, I. (2001), "Desktop Virtual Reality Interface for Computer-Aided Conceptual Design using Geometric Techniques", Vol. 12, No. 4, 309-329.

APPENDIX A
Definition of Basic Notions

Active hand model: The hand model is activated whenever motion is sensed in the detection environment, and reproduces the motion of the physical hand.

CVDE: Collaborative Virtual Design Environment. A group of designers can work on a product together, even in a co-located or a dislocated way, while the design process is supported by the computer.

DoF: Degrees of freedom. A set of independent displacements that specify completely the displaced or deformed position of the body or system. (from www.wikipedia.org)

Hand anatomy: Information about the anatomical structure and motion constraints of the arm and the hand.

Hand ergonomics: Ergonomic data defines what is comfortable and natural for the user in terms of arm and hand motions.

Hand model: The human hand is modeled based on specific anatomic and ergonomic rules and assumptions.

Hand modeling assumptions: The hand model is built on the basis of anatomical and ergonomic rules of the human hand.

Hand motion: Sequences of two-handed hand and arm postures changing continuously over time, expressing either constructive (for geometry generation) or procedural word (for geometry manipulation) of the HML, and conveying motion trajectory and three-dimensional geometric information in real-time.

HML word recognition: In this process the performed hand motion is compared with the stored HML words and the best matching is selected as a modeling instruction.

HML word: The letters of the HML are signs produced as postures of the hand. A purposeful combination of sequences of changing postures of the hand or the hands is a word of the HML. Thus, words represent the lowest semantic level of constructive actions of shape conceptualization.

HML: Hand Motion Language. A formal modeling language which is based on the postures and the movements of the human hand. It has been developed using the analogy of a verbal-textual language, which consists of an alphabet and a grammar interweave in the language. Words and sentences are created from the letters of the alphabet according to the grammatical rules to be applied. From the sentences paragraphs and chapters are constructed according to the context of communication.

Landmark: Easily recognizable and identifiable points of the human body.

Motion detection environment: The purposefully designed and established design environment, where the user's hand motion can be detected. It contains special motion detection equipment and three-dimensional display.

Motion detection: The process in which the hands are located and tracked in the motion detection environment.

Motion scanning: The process in which the motion information is extracted from the moving hands. The motion information is – or has to be converted into- a sequence of timestamped three-dimensional positions of dedicated landmarks of the hands.

Motion trajectory generation: The process in which the information coming from motion scanning is converted into the mathematical model of the motion trajectory.

Motion trajectory modeling: The mathematical description of the motion trajectories. The basis of the modeling is a sequence of three-dimensional positions.

Motion trajectory: The path which is swept by the hand motion. We make difference between the global hand motion and the individual motion trajectories of dedicated hand landmarks.

Multimodal interaction: It provides the user with multiple modes of interfacing with a system beyond the traditional keyboard and mouse input/output. The most common such interface combines a visual modality (e.g. a display, keyboard, and mouse) with a voice modality (speech recognition for input, speech synthesis and recorded audio for output). However other modalities, such as pen-based input or haptic input/output, may be used. (from www.wikipedia.org)

Occlusion: In two-handed interaction, a situation when a landmark of the hand is hidden from the detection device, because it is covered by the other hand.

Personalization: The hand model is personalized based on the size of the actual user's hand.

Self occlusion: In two-handed interaction, a situation when a landmark of the hand is hidden from the detection device, because it is covered by other parts of the same hand.

Surface generation: When the recognized HML word is a geometric one, its motion trajectory is converted to a point cloud to generate a surface.

Surface manipulation: When the recognized HML word is a manipulation one, it is sent to the shape modeler to realize the modeling action.

User: In our application the user is a designer, an engineer or a form-giving stylist who wants to specify a certain shape appears in the conceptualization phase of the design process.

VDIM: Vague Discrete Interval Modeler, the computer-aided geometric modeler which is used for modeling and visualizing the designed shapes. The hand motion based design will be connected to the VDIM as a highly interactive input.

APPENDIX B
HML Vocabulary

Table 1 Procedural words

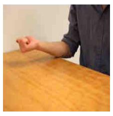
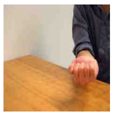
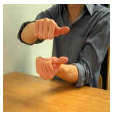
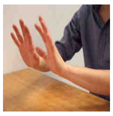
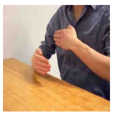
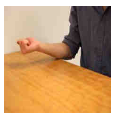
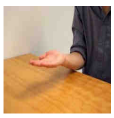
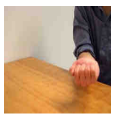
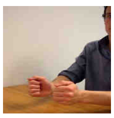
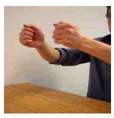
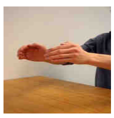
Words	Descriptions	Signs
Neutral	Designer is inactive	
Start End	Opens a sentence Closes a sentence	 
Stop Resume	Discontinues a sentence Restarts an unfinished sentence	 
Share Obtain	Stops to let another designer work on a shape Takes over a shape to further work on it	 
Undo Redo	Cancels an unwanted or faulty sentence Starts again a previously undone sentence	 

Table 2 Geometric words

Plane	Specifies a planar surface as halfspace	  
Cylinder	Specifies a cylindrical surface as halfspace	  
Cone	Specifies a conical surface as halfspace	  

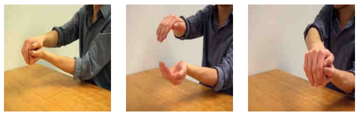
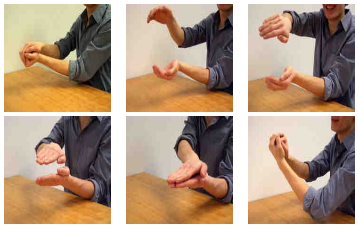
Sphere	Specifies a spherical surface as halfspace	
Ellipsoid	Specifies a ellipsoidal surface as halfspace	
Freeform	Specifies arbitrary freeform surface as halfspace	

Table 3 Identification words

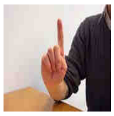
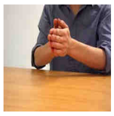
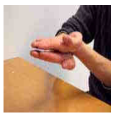
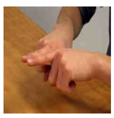
Identify point	Indicates and selects a point of a shape	
Identify curve	Indicates and selects a curve or line of a shape	
Identify surface	Indicates and selects a surface or plane of a shape	
Identify object	Indicates and selects a shape	

Table 4 Connectivity words

Surface to surface	Specifies face-to-face connection of two surfaces	
Surface to curve, or vice versa	Specifies face-to-edge or edge to face connection of two surfaces	
Surface to point, or vice versa	Specifies face-to-vertex or vertex to face connection of two surfaces	
Curve to curve	Specifies edge-to-edge connection of two surfaces	

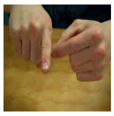
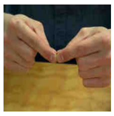
Curve to point, or vice versa	Specifies edge-to-vertex or vertex to edge connection of two surfaces	
Point to point	Specifies vertex-to-vertex connection of two surfaces	

Table 5 Positioning word

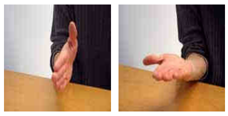
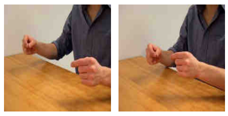
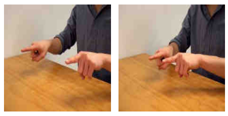
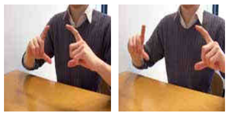
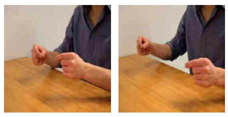
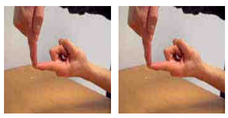
Turn	Turns a shape around by 90 degrees	
Distance by points	Increases or decreases the distance between points of two shapes	
Distance by curves	Increases or decreases the distance between curves of two shapes	
Distance by surfaces	Increases or decreases the distance between surfaces of two shapes	

Table 6 Scaling words

Size by surfaces	Increases or decreases the distance between two general surfaces or two parallel planes	
Size by curves	Increases or decreases the distance between two general curves or two parallel lines	
Size by points	Increases or decreases the distance between two points	
Angle by surfaces	Increases or decreases the angle between two surfaces	
Angle by edges	Increases or decreases the angle between two lines (tangent or common)	

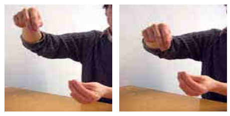
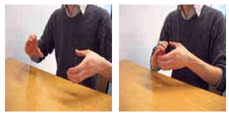
Zoom in/out	Increases or decreases the shape proportionally	

Table 7 Assembling words

Construct Deconstruct	Puts together entities to form a simple shape Separates simple shapes to shape entities	
Compose Decompose	Composes compound or hybrid shapes Decomposes shapes to elementary shapes	
Assemble Disassemble	Assembles shapes to form object with DoF Disassembles shapes of an object with DoF	
Put aside Bring in	Preserves a shape model for later reuse Retrieves a preserved shape model	
Cut through Cut out	Slices a shape model into two parts Removes part of shape model	

APPENDIX
Data Extraction for Surface Manipulation

C

Table 1 Identification words with regards to data extraction

HML word	Data to be extracted	Operation
Identify point	Coordinates of the middle point of the line connecting the tip of the index finger and the tip of the thumb	Find closest point in the model to the extracted point
Identify curve	Coordinates of the PIP joint of the index finger	Find closest line in the model to the extracted point
Identify surface	Coordinates of the MCP joint of middle finger	Find closest surface patch in the model to the extracted point
Identify object	Coordinates of the middle point of the line connecting the MCP joint of the middle finger on the left hand and the same joint on the right hand	Find closest object in the model to the extracted point

Table 2 Connectivity words with regards to data extraction

Surface to surface	The identifiers of the two surfaces to be connected and the coordinates of the coordinates of the middle point of the line connecting the MCP joint of the middle finger on the left hand and the same point on the right hand	Connect the input surfaces and place the result on the extracted point
Surface to curve, or vice versa	The identifiers of surface and the curve to be connected and the coordinates of the middle point of the line connecting the MCP joint of the middle finger on the left hand and the same point on the right hand	Connect the input surface and curve, and place the result on the extracted point

Surface to point, or vice versa	The identifier of the surface and the coordinates of the point to be connected, and the coordinates of the middle point of the line connecting the tip of the index finger and the tip of the thumb on the hand which selects the point	Connect the input surface and point and place the result on the extracted point
Curve to curve	The identifier of the curves to be connected, and the coordinates of the middle point of the line connecting PIP joint of the index finger on the left hand and the same point on the right hand	Connect the input surface and point and place the result on the extracted point
Curve to point, or vice versa	The identifier of the curve and the coordinates of the point to be connected, and the coordinates middle point of the line connecting the tip of the index finger and the tip of the thumb on the hand which selects the point	Connect the input curve and point and place the result on the extracted point
Point to point	The coordinates of the points to be connected, and the middle point of the line connecting the middle points of the lines connecting the tip of the index finger and the tip of the thumb on the left and on the right hand	Connect the points and place the result on the extracted point

Table 3 Positioning word with regards to data extraction

Turn	Identifier of the selected geometric entity, turning angle defined by the angular changes of the line connecting the wrist and the tip of thumb from frame to frame, and axis of rotation defined by the line connecting the wrist and the tip of the middle finger	Turn the selected geometric element with the extracted degree around the extracted axis
Distance by points	Identifiers of the selected entities, and the distance defined by the coordinates of the points on the lines connecting the tip of the index finger and the tip of the thumb on the left hand and the same points on the right hand	Set the distance of the selected entities to the extracted value by repositioning the entities

Distance by curves	Identifiers of the selected entities, and the distance defined by the coordinates of PIP joint of the index finger on the left hand and the same point on the right hand	Set the distance of the selected entities to the extracted value by repositioning the entities
Distance by surfaces	Identifiers of the selected entities, and the distance defined by the coordinates of MCP joint of the middle finger on the left hand and the same point on the right hand	Set the distance of the selected entities to the extracted value by repositioning the entities

Table 4 Scaling words with regards to data extraction

Size by surfaces	Identifiers of the selected entities, and the distance defined by the coordinates of the MCP joint of the middle finger on the left hand and the same point on the right hand	Set the size of the object containing the surfaces to the extracted value
Size by curves	Identifiers of the selected entities, and the distance defined by the coordinates of the PIP joint of the index finger on the left hand and the same point on the right hand	Set the size of the object containing the curves to the extracted value
Size by points	Identifiers of the selected entities, and the distance defined by the coordinates of the points on the lines connecting the tip if the index finger and the tip of the thumb on the left hand and the same points on the right hand	Set the size of the object containing the points to the extracted value
Angle by surfaces	Identifiers of the selected entities, and the turning angle defined by the lines connecting MCP joint of the middle finger and the IP joint of the thumb on the left hand and the same line on the right hand	Set the angle between the selected surfaces to the extracted degree
Angle by edges	Identifiers of the selected entities, and the turning angle defined by the lines connecting MCP joint and the tip of the index finger on the left hand and the same line on the right hand	Set the angle between the selected edges to the extracted degree
Zoom in/out	Distance of the tip of the index finger on the left hand and the same point on the right hand	Zoom the modeling space according to the extracted value

Table 5 Assembling words with regards to data extraction

Construct Deconstruct	Identifiers of the entities in the modeling space	Construct or deconstruct the entities in the modeling space
Compose Decompose	Identifiers of the entities in the modeling space	Compose or decompose the entities in the modeling space
Assemble Disassemble	Identifiers of the entities in the modeling space	Assemble or disassemble the entities in the modeling space
Put aside Bring in	Identifier of the selected entity and the coordinates of the MCP joint of the middle finger	Put the selected entity to the extracted location
Cut through Cut out	Identifier of the selected entity and the cutting surface defined by the motion trajectories of the wrist and the MCP joint and tip of the middle finger	cut through the selected entity based on the cutting surface and in case of cut out delete the half-entity which is collided by the MCP joint of the middle finger

APPENDIX Modeling Steps in HML and in CAD

D

Product 1 (CD case) – HML

1. draw first side of box (“generate plane”)
2. draw second side of box
3. draw third side of box
4. draw fourth side of box
5. draw fifth side of box
6. draw sixth side of box
7. apply “construct” HML word
8. apply “put aside” HML word
9. draw first side of box (“generate plane”)
10. draw second side of box
11. draw third side of box
12. draw fourth side of box
13. draw fifth side of box
14. draw sixth side of box
15. apply “construct HML word”
16. apply “put aside” HML word
17. draw first side of box (“generate plane”)
18. draw second side of box
19. draw third side of box
20. draw fourth side of box
21. draw fifth side of box
22. draw sixth side of box
23. apply “construct” HML word
24. apply “bring in” HML word
25. apply “bring in” HML word
26. apply “construct” HML word

Product 1 (CD case) – CAD

27. select Solid->Box->Corner To Corner, Height from menu
28. add first corner by mouse click
29. add second corner by mouse click
30. add height by mouse click
31. select Solid->Box->Corner To Corner, Height from menu
32. add first corner by mouse click
33. add second corner by mouse click
34. add height by mouse click
35. select first feature by mouse click
36. copy to clipboard by clicking on icon
37. place by clicking on icon
38. position object with the mouse and click

39. select all
40. select Solid->Union from menu
41. zoom out by clicking on icon and mouse move

Product 2 (chair) – HML

42. draw first part of torus
43. rotate view
44. draw second part of torus
45. apply “construct” HML word
46. select object
47. rotate object
48. draw third part of torus
49. apply “construct” HML word
50. select object
51. rotate object
52. draw fourth part of torus
53. apply “construct” HML word
54. rotate for viewing
55. select object
56. position object
57. draw cylinder (first leg of chair)
58. select cylinder
59. resize cylinder
60. position cylinder
61. resize cylinder
62. draw cylinder (second leg of chair)
63. select cylinder
64. resize cylinder
65. position cylinder
66. resize cylinder
67. apply “construct” HML word
68. select object
69. rotate object
70. draw cylinder (third leg of chair)
71. select cylinder
72. resize cylinder
73. position cylinder
74. resize cylinder
75. draw cylinder (fourth leg of chair)
76. select cylinder
77. resize cylinder
78. position cylinder
79. resize cylinder
80. apply “construct” HML word
81. select object
82. resize object
83. draw first part of freeform shape (seatback holder)
84. draw second part of freeform shape (seatback holder)
85. apply “construct” HML word
86. select object
87. rotate object
88. draw first part of freeform shape (seatback holder 2)

89. draw second part of freeform shape (seatback holder 2)
90. apply "construct" HML word
91. apply "put aside" HML word
92. draw first side of box (seatback)
93. draw second side of box (seatback)
94. draw third side of box (seatback)
95. draw fourth side of box (seatback)
96. draw fifth side of box (seatback)
97. draw sixth side of box (seatback)
98. apply "construct" HML word
99. apply "bring in" HML word
100. select object
101. resize object
102. select box
103. position box
104. resize box
105. position box
106. apply "construct" HML word
107. "put aside" global shape
108. draw first surface of box (seat)
109. draw second surface of box (seat)
110. draw third surface of box (seat)
111. draw fourth surface of box (seat)
112. draw fifth surface of box (seat)
113. draw sixth surface of box (seat)
114. apply "construct" HML word
115. "bring in" global shape
116. resize global shape
117. select box
118. position box
119. resize box
120. position box
121. apply "construct" HML word
122. resize global shape
123. draw cutting surface (on seatback)
124. resize cutting surface
125. apply "cut out" HML word
126. draw cutting surface (on seatback)
127. resize cutting surface
128. apply "cut out" HML word
129. draw cutting surface (on seat)
130. resize cutting surface
131. apply "cut out" HML word
132. draw cutting surface (on seat)
133. resize cutting surface
134. apply "cut out" HML word
135. rotate for viewing

Product 2 (chair) – CAD

136. select Solid->Torus from menu
137. select center of the torus
138. set radius by mouse move
139. set second radius by typing its value
140. select Solid->Cylinder from menu
141. select center of base
142. set radius
143. set height
144. position cylinder
145. select Transform->Rotate from menu
146. select cylinder
147. rotate cylinder
148. select Transform->Mirror from menu
149. select mirror plane -> second leg
150. select mirror plane -> third leg
151. select mirror plane -> fourth leg
152. select Curve->Freeform->Interpolate points from menu
153. select first point of the curve
154. select second point of the curve
155. select third point of the curve
156. select fourth point of the curve
157. select fifth point of the curve
158. select sixth point of the curve
159. select Curve->Circle->Center, Radius from menu
160. select center of circle
161. type in the radius of the circle
162. select Surface->Sweep 1 Rail from menu
163. select rail curve (circle)
164. select cross section curves (freeform line)
165. click on curve on pop-up menu
166. select Transform->Mirror from menu
167. select surface (seatback)
168. select mirror plane -> second seatback
169. select Solid->Box->Corner To Corner, Height from menu
170. select first corner
171. select second corner
172. set height
173. zoom out
174. select Solid->Box->Corner To Corner, Height from menu
175. select first corner
176. select second corner
177. add height
178. position box (seatback)
179. click on zoom icon
180. zoom in new part
181. select Transform->Rotate from menu
182. select center of rotation
183. select first reference point
184. select second reference point
185. click on zoom icon

186. zoom in new part
187. position seatback
188. select Transform->Rotate from menu
189. select center of rotation
190. select first reference point
191. select second reference point
192. browsing menu
193. save file
194. select Curve->Freeform from menu
195. select first point of curve
196. select second point of curve
197. select third point of curve
198. select fourth point of curve
199. select fifth point of curve
200. select Surface->Sweep Rail from menu
201. select rail curve
202. select cross section curve
203. select Solid->Cap Planar Holes from menu
204. select surface
205. select Transform->Mirror from menu
206. select part
207. select mirror plane
208. browsing menu
209. select Transform->Subtract Solid from menu
210. select first feature
211. select second feature
212. select third feature
213. rotate in perspective view for checking
214. select Curve->Ellipsis from menu
215. set first value
216. set second value
217. set Surface->Extrude from menu
218. select ellipsis
219. rotate view
220. set length of extrusion
221. select Solid->Cap Planar Holes from menu
222. select part
223. select Transform->Mirror from menu
224. select part
225. set snapping parameters
226. select mirror plane
227. select Solid->Union from menu
228. select first part
229. select second part
230. select third part
231. rotate in perspective view for checking

Product 3 (shaver) – HML

232. draw surface one
233. draw surface two
234. draw surface three
235. draw surface four
236. draw surface five
237. draw surface six
238. apply “construct” HML word
239. select global shape
240. position global shape
241. resize global shape
242. generate cylinder
243. resize cylinder
244. position cylinder
245. resize cylinder
246. apply “construct HML word”
247. generate cylinder
248. resize cylinder
249. position cylinder
250. resize cylinder
251. apply “construct” HML word
252. generate cylinder
253. resize cylinder
254. position cylinder
255. resize cylinder
256. apply “construct” HML word
257. put aside object
258. draw first surface of button
259. draw second surface of button
260. draw third surface of button
261. apply “construct” HML word
262. select object
263. rotate object
264. draw fourth surface of button
265. draw fifth surface of button
266. apply “construct” HML word
267. select object
268. rotate object
269. draw sixth surface of button
270. apply “construct” HML word
271. bring in global shape
272. rotate global shape
273. resize global shape
274. select button
275. position button
276. resize button
277. apply “construct” HML word
278. resize global shape
279. rotate for viewing

Product 3 (shaver) – CAD

280. click on circle icon
281. select center of circle
282. set radius of circle
283. select circle
284. click on “control points on” icon
285. select one control point
286. select another control point
287. select another control point
288. select Transform->Rotate from menu
289. select center of rotation
290. select first reference point
291. select second reference point
292. select one control point
293. select another control point
294. select another control point
295. select Transform->Rotate from menu
296. select center of rotation
297. select first reference point
298. select second reference point
299. select one control point
300. select another control point
301. select another control point
302. select Transform->Scale->Scale2D from menu
303. select origin
304. select first reference point
305. select second reference point
306. select curve
307. select Edit->Copy from menu
308. select Edit->Paste from menu
309. position second curve
310. rotate scene for viewing
311. select second curve
312. select 2D view
313. select one control point
314. select another control point
315. select another control point
316. select Transform->Scale->Scale2D from menu
317. select origin
318. select first reference point
319. select second reference point
320. deselect control point view by clicking on icon
321. select Transform->Scale->Scale2D from menu
322. select origin
323. select first reference point
324. select second reference point
325. select Edit->Copy from menu
326. select Edit->Paste from menu
327. select 4 views view
328. position third curve
329. select 2D view

330. select Transform->Scale->Scale1D from menu
331. select origin
332. select second reference point
333. select 4 views view
334. select first curve
335. select 2D view
336. select Transform->Rotate from menu
337. select origin
338. select first reference point
339. select second reference point
340. select second curve
341. position second curve
342. select 4 views view
343. select curve by points icon
344. select first point
345. select second point
346. select third point
347. select fourth point
348. select curve by points icon
349. select first point
350. select second point
351. select third point
352. rotate perspective view for checking
353. select Surface->Sweep2Rails from menu
354. select first rail curve
355. select second rail curve
356. select first cross section curve
357. select second cross section curve
358. select third cross section curve
359. adjust first curve seams
360. adjust second curve seams
361. adjust third curve seams
362. pop-up menu appears -> click on Cancel
363. select second curve
364. select Transform->Scale->Scale3D from menu
365. select origin
366. select first reference point
367. select second reference point
368. deselect second curve
369. select Surface->Sweep 2 Rails from menu
370. select first rail curve
371. select second rail curve
372. select first cross section
373. select second cross section
374. select third cross section
375. adjust first curve seams
376. adjust second curve seams
377. adjust third curve seams
378. pop-up menu appears -> click on OK
379. rotate perspective view for checking
380. select curve
381. select Edit->Copy from menu

382. select Edit->Paste from menu
383. position new curve along line
384. select surface in wireframe view
385. select Transform->Rotate from menu
386. select center of rotation
387. select first reference point
388. select second reference point
389. deselect surface
390. select Edit->Select Objects->Curves from menu
391. deselect lastly generated curve
392. select Edit->Object Properties from menu
393. pop-up menu appears -> select Layer 1
394. pop-up menu appears -> add name to it
395. deselect visualization of the first layer
396. select lastly generated curve
397. select 2D view
398. click on curve by control points icon
399. deselect curve by control points icon
400. select 4 views view
401. select lastly generated curve
402. zoom out in front view
403. select Edit->Copy from menu
404. select Edit->Paste from menu
405. select Edit->Paste from menu
406. position curve along line
407. position curve along line
408. select curve by points icon
409. select a point on the curve
410. reposition the selected point
411. select another curve
412. select Transform->Scale->Scale1D from menu
413. zoom in in front view
414. select origin
415. select first reference point
416. select second reference point
417. click on fit to screen view icon
418. zoom in in front view
419. select curve
420. select Transform->Scale->Scale2D from menu
421. select origin
422. select first reference point
423. select second reference point
424. click on fit view icon
425. select Surface->Loft from menu
426. select first curve
427. select second curve
428. select third curve
429. adjust first curve seams
430. adjust second curve seams
431. adjust third curve seams
432. rotate in perspective view for checking
433. click on shaded view icon

434. click on perspective view to enlarge
435. rotate view for checking
436. rotate object for checking
437. select 4 views view
438. reposition object in front view
439. click on curve by points icon
440. adjust drawing parameters
441. zoom in in front view
442. select first point of curve
443. select second point of curve
444. select third point of curve
445. click on curve by control points icon
446. zoom in in right view
447. adjust a point of the curve
448. rotate in perspective view for checking
449. select Surface->Patch from menu
450. select curve
451. pop-up menu appears->click on Preview button
452. click on OK button
453. rotate in perspective view for checking
454. click on curve by control points icon
455. zoom in in top view
456. rotate in perspective view for checking
457. select Surface->Blend Surface from menu
458. select perspective view
459. rotate to position
460. select curve (for first edge)
461. select curve (for second edge)
462. switch to 4 views view
463. Adjust Blend Bulge pop-up menu appears
464. click on fit view icon
465. zoom in in front view
466. adjust blend parameters
467. click on OK in pop-up menu
468. rotate in perspective view for checking
469. click on join icon
470. select first surface
471. select second surface
472. select third surface
473. select fourth surface
474. rotate in perspective view for checking
475. select surface
476. select Solid->Cap Planar Holes from menu
477. rotate in perspective view for checking
478. browsing in menu
479. select View->Set CPlane->3Points from menu
480. select CPlane origin
481. select object to orient CPlane to
482. rotate in perspective view
483. select object
484. select View->Set CPlane->World Top from menu
485. rotate in top view

486. click on curve by points icon
487. position object in front view
488. draw first point of curve
489. draw first point of curve
490. draw first point of curve
491. draw first point of curve
492. draw second point of curve
493. draw third point of curve
494. draw fourth point of curve
495. draw fifth point of curve
496. draw sixth point of curve
497. draw seventh point of curve
498. draw eighth point of curve
499. draw ninth point of curve
500. click on curve by control points icon
501. zoom in in front view
502. change the position of a point
503. change the position of a point
504. change the position of a point
505. click on curve by points icon
506. select first point of curve
507. select second point of curve
508. click on join icon
509. select first curve
510. select second curve
511. select Surface->Revolve from menu
512. select curve to revolve
513. define start of axis
514. define end of axis
515. pop-up menu appears -> click on OK
516. click on fit view icon
517. rotate in perspective view
518. select Transform->Orient->On Surface from menu
519. select object to orient
520. switch to perspective view
521. select surface to orient to
522. switch to 4 views view
523. copy cylinder
524. paste cylinder
525. position cylinder
526. paste cylinder
527. position cylinder
528. paste cylinder
529. position cylinder
530. select cylinder
531. delete cylinder
532. select the three cylinders
533. reposition cylinders
534. switch to perspective view
535. switch to 4 views view
536. zoom in in front view
537. select global shape

538. select Solid->Fillet Edge
539. select edge
540. rotate perspective view for checking
541. select first cylinder
542. select second cylinder
543. select third cylinder
544. zoom in in perspective view
545. reposition cylinders
546. rotate in perspective view for checking
547. set to top view
548. set to bottom view
549. zoom in in bottom view
550. select Curve->Freeform->Control points from menu
551. define first point of the curve
552. define second point of the curve
553. define third point of the curve
554. define fourth point of the curve
555. define fifth point of the curve
556. define sixth point of the curve
557. define seventh point of the curve
558. define eighth point of the curve
559. switch to curve view
560. click on curve by control points icon
561. change a point
562. change a point
563. change a point
564. change a point
565. change a point
566. change a point
567. zoom in in bottom view
568. change a point
569. change a point
570. deselect curve by control points icon
571. select curve
572. select Curve->Curve From Objects->Project from menu
573. select surface to project onto
574. select two curves
575. click on split icon
576. select cutting object
577. click on explode icon
578. select surface
579. click on split icon
580. select cutting object
581. select curve
582. select Transform->Scale->Scale3D
583. define origin
584. define first reference point
585. define second reference point
586. zoom in in front view
587. position curve
588. select Surface->Blend Surface from menu
589. select first curve

590. select second curve
591. Adjust Blend Bulge pop-up menu appears -> click OK
592. click fit view icon
593. rotate in perspective view for checking
594. zoom in and out in perspective view

SUMMARY

Designers have been living with the limitations of Computer Aided Design (CAD) for a rather long time. One- and two-dimensional input devices determined the way of entering shape information to design support systems. Over the years, it has been recognized that these conventional input devices pose too many constraints when applied in Computer Aided Conceptual Design (CACD) systems. The main constraints are caused by the cumbersomeness of interaction, which negatively influences creativity, the essence of shape conceptualization. As a result of four years of Ph.D. research, a new component has been studied and developed for multimodal interaction. This new component is based on human hand motions and intends to support shape externalization in advanced visualization environments. Hand motions are capable to express 3D shape information and gestural instructions concurrently and directly, and therefore relieve the designer of going into a process decomposing three-dimensional problems to two-dimensional ones. My research has proved that by extracting information from the physical motion trajectories of the human hands, information can be generated for the construction of the indicated shape elements of three-dimensional shapes as well as for shape manipulation.

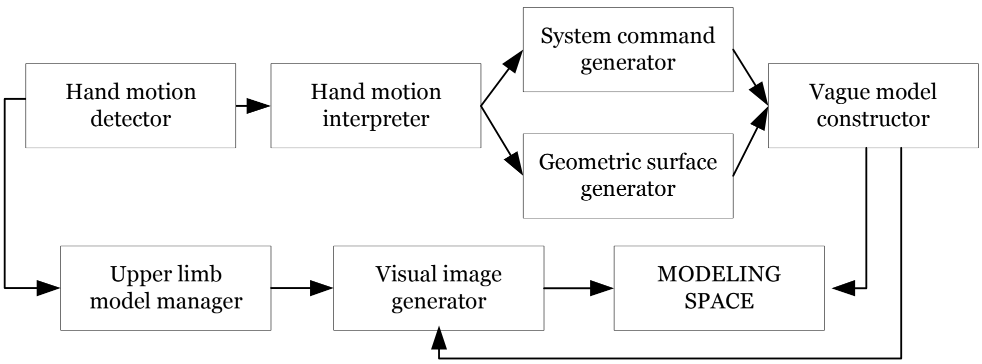


Figure 1 Hand motion-based shape generation

The joint vision of the colleagues at the Section of CADE was to offer a design environment for industrial designers, in which they use their hand motions to externalize form-giving ideas in the 3D space, and immediately store these shape ideas on the computer for later use or refinement. We hypothesized that this environment can provide designers with real-time feedback and with an intuitive way to express shape concepts. Our strategic objective was to develop a comprehensive concept and a first implementation of a hand motion-based shape design environment, as a constituent of a future proactive reality environment (PRE) for shape designers. Figure shows the components of hand motion processing, and illustrates that the major tasks to be solved were: (i) detection of spatial hand motions to gather hand motion data, (ii) interpretation of the hand motions to understand designers' actions, (iii) generation of geometric or system commands to provide the shape modeler with proper input and (iv) management of the upper limb model to visualize the designer in the virtual space. Figure suggests that in an ideal situation, the visualization happens directly in the space of the hand motions. If it is not the case, the upper limb model needs to be visualized in the virtual space to help the navigation of the designer.

In the context of hand motion processing, the main questions to be addressed were: (i) how spatial hand motions can be detected, (ii) what the minimal information that has to be detected is, (iii) how virtual surfaces can be generated with hand motions and (iv) how a predefined set of shape construction operations can be recognized. As a basis for this Ph.D. research, a framework was established to solve the problems of (i) finding a detection equipment which is able to measure 3D positions in real-time and fulfill ergonomic requirements at the same time, (ii) constructing an active upper limb model which is able to complete data coming from the detection device and makes a connection between the detection device and the modeling engine, (iii) converting physical hand motion information to virtual surface information and (iv) interpreting hand motions and converting them to modeling commands executable by the modeling engine. As far as the interpretation of the hand motions is concerned, the (i) segmentation of meaningless and meaningful parts of hand motions had to be solved, and (ii) a method for the recognition of the meaningful hand motions had to be developed.

The problem of hand motion-based shape conceptualization is complex, therefore a systematic analysis was necessary. The following sub-problems were identified as the major issues to be studied in the explorative research. I learnt (i) that the problem of hand motion-based shape design has to be approached taking into account human aspects. Therefore, a well-designed hand motion language was needed which is not only interpretable for computers, but easily understandable and learnable for humans. I also learnt that (ii) the hand motion interpretation method has to be able to handle a wide variety of hand motions in a fast manner to be able to provide the user with immediate feedback, (iii) the hand motion detection technology has to fulfill both technical requirements which enable seamless interaction, and requirements from ergonomics which provide user comfort, (iv) based on the interpreted hand motion of the designer, motion trajectories has to be converted either to geometric description of surfaces or to system control instructions, (v) virtual presence has to be solved by visualizing the user in the virtual space due to the lack of truly three-dimensional visualization methods which would enable the visualization of the physical hand motions and the virtual surfaces in the same space. Finally, I learnt that (vi) the hand motion-based interface has to be connected to a conceptual modeling system which is able to exploit the capability of hand motions to describe vague shapes.

It has been recognized that this research has to follow the rules of explorative research. Applied research methods include exploratory study, conceptualization of the hand motion-based interface and research in applicable theories, proof-of-concept implementation of the hand motion-based interface, experiments with the proof-of-concept implementation and verification of its functioning, validation of the applied theories and user studies. An extensive literature review has been completed in which the emphasis was put on the different approaches of hand motion processing. Companies producing motion tracking equipment were contacted, and after a qualitative and quantitative analysis of the options, the best-fitting technology was selected. Various theories about hand motion interpretation and about human hand modeling were investigated and analyzed and the applicable theories were adapted. Experiments were performed to verify the correctness of hand motion interpretation. The established theories were tested in an indirect way by developing the hand motion-based interface. This included the design and the proof-of-concept implementation of the dedicated algorithms for each phase of hand motion processing. A usability study was designed and conducted to test the implementation of the hand motion-based interface in a qualitative way. User opinions were gathered by a questionnaire applying Likert-scale. Case studies were developed to test the practical utility of hand motion-based shape conceptualization. The gathered quantitative data in these studies were evaluated using statistical analysis.

Towards multimodal interaction, the concept of hand motion-based interfaces has been proposed by many researchers mainly because of their potential to express geometric information directly in space. The main concern at the development of the majority of hand motion-controlled systems was the ease of recognition from a technological point of view, rather than approaching the problem from human aspects, such as the understandability and learnability of hand motions. Therefore, I adopted a set of hand motions which could be effectively used in shape conceptualization. This set, called Hand Motion Language (HML) served as a basis for shape concept generation and manipulation. The basic elements of the HML are words, and each HML word carries either geometric, structural or manipulation information for shape modeling, or control or procedural information for the modeling system. Based on this language, I defined hand motion-based shape conceptualization as a process comprising successively generated HML words. For the sake of clarity, I also defined hand motions, as sequences of two-handed hand- and arm postures changing continuously over time, expressing either shape description or modeling command, and conveying motion trajectory and 3D geometric information. It has been identified that three separate processes are needed for a successful interaction, namely, (i) hand motion detection, (ii) hand motion interpretation and (iii) geometric information generation.

In the process of hand motion detection sufficient amount of information about the motion trajectories of the hands are measured. As far as hand motion interpretation is concerned, I hypothesized that the HML words can be extracted by segmentation of the motion trajectories based on the investigation of changes in hand postures in each frame of the recorded motion. The reason for this is the fact that hand postures change in a patterned way or do not change at all while moving on a specific trajectory of motion during performing an HML word. With the help of postural changes, the beginning and the end of the HML words can be identified, and the sequence of postures can be used to interpret the relevant modeling command. The recognized command and the corresponding geometric information form the input of the geometric modeling system used for visualization.

In order to be able to interpret the instructions of the designer, first the hand motions have to be detected. For this purpose, after an extensive technology- and market search, I decided to use a passive optical tracking system. The selected camera system uses infrared light to measure the position of retro-reflective markers attached to specific landmarks of the designer’s hand. The positions of the applied marker set was defined after a careful analysis of applicable landmarks on the hand, and the minimum set was defined which is sufficient both for surface generation and the interpretation of HML words. The tracking system enables the free movement of the user, since there are no cables to connect the user to the detection device or to the computer. The small markers are extremely light and do not cause any discomfort to the designer. The six camera based system enables the recognition of complex two-handed postures and motions, and the position data measured by the tracking device serve as the input for hand motion interpretation and for surface generation.

For developing a method for hand motion interpretation, first an analysis of the HML was necessary. In this analysis I learnt that an HML word has two states: a transition state and a steady state. In the transition state the postures of the hands are continuously changing. The steady state is the useful part of an HML word, without postural changes. Therefore, the transition state refers to a transition from one HML word to another, and the steady state has meaning indicating a modeling command. It has been also learnt that the HML consists of one-handed, double-handed and two-handed words. As its name implies, one-handed words are formed by a certain posture of one hand in the steady state of a sign. Double-handed words are actually the combinations of one-handed words. In case of two-handed words, the two hands are simultaneously moving and taking up the same posture on the trajectory of motion. To interpret the steady part of the HML words, a posture recognition method was needed. For this purpose, features can be identified on the hand postures which express certain relationships between the landmarks of the hand. I exploited these characteristics of the HML during the development of the hand motion interpretation method.

I analyzed all occurring postures and looked for those features which distinguish the postures the most. Towards a formal representation of the features, a minimal set of descriptive parameters has been assigned. These are called inter-hand parameters as they refer to a certain relationship of landmarks of one hand. For classifying the postures, I decided to use a decision tree based method. The abovementioned parameter values serve as input for the decision tree, and the posture recognition process is actually a search in the tree, which has parameters in its nodes and postures in its leaves. However, this is not the end of the hand motion interpretation process. Recognized one-handed postures are combined using a set of rules to interpret two-handed HML words. Some ambiguous cases need few additional parameters, which were named inter-hand parameters referring to the fact that they represent a certain relationship of the two hands. As a result I could conclude that a large variety of hand motions can be interpreted by applying only a small number of parameters.

During this research I set up two different detection and modeling environments. One of them is the aforementioned passive optical tracking system which was calibrated to be able to track hand movements of multiple designers in an office-like environment. In this case, a regular 2D monitor was used as output device. The other setup was designed for an experiment to test the concept of the hand motion-based interface. In this environment I used a pair of datagloves with 3D magnetic trackers attached one to the back of each glove, and 3D monitor with stereo-glasses was used for visualization. This setup provided a reliable testing environment, however, only one user could work with the system at the same time. The HML interpreter was integrated into the VR Juggler environment, because VR Juggler provides a virtual platform for virtual reality application development. As far as our ultimate goal is concerned, I concluded that the passive optical tracking method offers more opportunities, and therefore it is worthwhile to invest time into future research, especially if the tracking can be connected to truly 3D visualization.

With the passive optical tracking system, a software application was provided to enable model building for tracking and real-time model visualization. The model – in our case the Upper Limb Model (ULM) – was developed upon the landmarks used for hand motion interpretation. The motion capture session starts using this software application and with the related Software Development Kit (SDK) measured data are streamed into my self-developed software. Because of the passive technology, identification of the markers attached to the dedicated landmarks happens based upon the ULM. The SDK communicates with the hand motion interpretation software, which exploits the capabilities of the See5, the adapted classification software. A decision tree was constructed with the help of See5 using the aforementioned minimal set of intra-hand parameters. The related classification code was integrated into the self-developed software for further usage of its output hand postures.

In short, the information flow in the integrated software is as follows. Raw 3D data received as input from the tracking system through its SDK are converted to descriptive parameters. These parameters form the input of the decision tree which comes back with a posture both for the left and for the right hand. Hand motion interpretation happens through two self-developed algorithms. Segmentation finds the steady parts of the hand motion by comparing the two-handed postures in consecutive frames of motion. When these two-handed postures are the same for a certain number of frames of motion optimized by experiments, an other process, namely the hand motion recognition starts. Hand motion recognition is a fast process, since the recognized two-handed postures limit the possible HML words to a very small number, and by investigating some additional inter-hand parameter, the correct HML word is immediately selected. Finally, the HML word together with the related geometric information provides the input for the geometric modeler, more specifically, to the Vague Discrete Interval Modeler (VDIM). VDIM constructs and visualizes the vague shape models generated by hand motions.

Based on an extensive literature study, the various hand motion processing technologies were sorted into four processing categories, which have been called (i) direct incomplete, (ii) direct complete, (iii) indirect incomplete and (iv) indirect complete processing. Direct and indirect refers to the way of transferring information from the physical space, in which the hands are moving, to the virtual space, where the shape is modeled. Indirect data transfer means the usage of an active hand model to extend the detected data for a better representation of the swept surfaces, or for a better mapping of a manipulative action. Human hands can be completely scanned, or some characteristic points (such as landmark points or silhouette points) can be detected. These two ways of obtaining shape information from the moving hands can be identified as complete and incomplete information extraction. These four categories were further elaborated according to the relationship of the hands and the information extracting devices. Certain devices are mounted on or touch the hand, while other devices can extract information at a distance. These relationships have been described as contact or non-contact. It has been found that direct complete and indirect incomplete hand motion processing technologies have the potential to support hand motion-based shape conceptualization. As far as the relationship of the hands and the detection device is concerned, non-contact technologies are preferred because of ergonomic requirements. However, it has been recognized after a thorough market search that current detection technologies are not fully capable of fulfilling all of the technical and ergonomic requirements, and therefore some compromise should be taken, at least in the forms of small and lightweight passive markers.

Experiments were designed and performed to test the reliability of the hand motion interpretation method. Results showed that if all data were available for the decision tree based classifier, the accuracy of HML word recognition is 100%. It has been studied that the generation of modeling commands related to HML words requires obtaining additional geometric information, and each HML word needs individual processing. However, the processing of higher dimensional entities, such as surfaces or objects, can be traced back to lower dimensional entities, such as points. This enables the reuse of the algorithms that are generated to process points, and keeps the algorithms on $O(\log N)$ order.

For evaluating the concept of hand motion-based shape generation and manipulation, a user study was designed and conducted, in which HML-based modeling was compared to conventional CAD modeling. It has been concluded that participants judged the HML method to be better than traditional CAD for conceptual shape design. Especially the category of learnability showed significant difference in favor of HML-based modeling, but the categories of operability, stimulation and satisfaction showed considerable differences as well. Participants (i) were significantly faster in creating conceptual shapes, (ii) found the hand motion input more intuitive and (iii) were more satisfied, which means that they are more willing to use this novel interaction method. It turned out that HML-based modeling makes people more tired physically than traditional CAD. It could also be observed that participants could create a variety of shapes with the HML-based modeling method using their fantasy. They did not simply try to copy the sample shape, which was shown to them, but created different ones based on their own imaginations. On the other hand, when using CAD, they mostly concentrated on the successful completion of the task, and they did not care about the originality of their work. Therefore, the resulting shapes were very similar to each other. As a personal experience I would like to mention that different type of people reacted differently on the modeling methods. More precisely, people who seemed to be more active, creative and curious, liked the HML more than passive people, and they could work with the HML software better. On the other hand, nervous type of people had difficulties to control the HML-based modeling software, because of their fast hand movements and their sudden emotional reactions when something went wrong.

Another experiment was designed to test the utility of hand motion-based shape design. The established criteria were the fidelity of the generated shapes, the accuracy of shape manipulation and the complexity of the modeling process in case of different shapes. In the fidelity oriented experiment, participants were asked to generate circles and cylinders using hand motions. They were asked to create these shapes in different orientations and different sizes. My general observation is that smaller shapes generated in the orientation preferred by the participants showed the highest fidelity. The experiment studying accuracy involved two manipulations, namely, setting the size of an object and setting the position of an object. This experiment pointed out the maximum achievable accuracy of manipulation when there is no time constraint, which is approximately 7 mm. The complexity of the shape has a strong influence on the total number of modeling actions. The objective of the last experiment was to explore this relationship. It turned out that the strongest points of hand motion-based modeling is the generation of freeform surfaces and the fact the objects can be positioned directly and immediately in the 3D space.

As final conclusions I claim that a purposefully designed hand motion language is not only intuitive and enjoyable for designers but it also stimulates creativity. As creativity is a key feature of shape conceptualization, a hand motion-based interface is extremely useful in the early phases of the design process. As far as the interpretation of hand motions is concerned, I concluded that words of a hand motion language can be interpreted based on investigating postural changes in the continuous hand motion. This way the useful parts of the continuous hand motion which carry modeling information can be found, and a wide variety of hand motions can be recognized by feeding a small number of parameters into a decision tree. For modeling purposes, our Hand Motion Language comprises surface generation- and manipulation, and procedural instructions. Both system control instructions and geometric entity descriptions can be generated in real time using passive optical tracking technology. A minimal set of detected landmarks enables both the interpretation of hand motion instructions and geometry generation as well. However, even the best designed hand motion interface can become devastating if connected to a conventional CAD system. Hand motions are mainly considered useful in conceptualization, because of its capability to support collaboration of designers and to externalize vague shape ideas. Current CAD systems do not facilitate these requirements. Therefore, I connected the hand motion-based interface to the Vague Discrete Interval Modeler to exploit the vague information naturally carried by hand motions. It has been realized, that the currently widely used two-dimensional visualization methods are not applicable when controlling them by spatial hand motions. The use of the well-known stereo-glasses for three-dimensional visualization proved to be useful, however, the ultimate goal should be the usage of spatial visualization, which enables multiple designers to work on the same shape concept and to view the shape from different perspectives. With air-borne spatial visualization designers would be able to generate virtual surfaces exactly in the same space where the physical hand motions are performed.

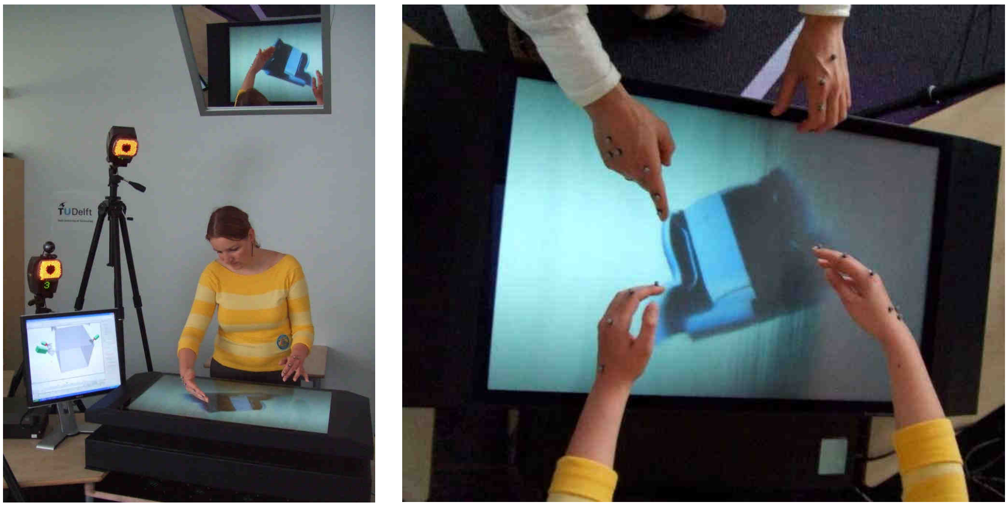


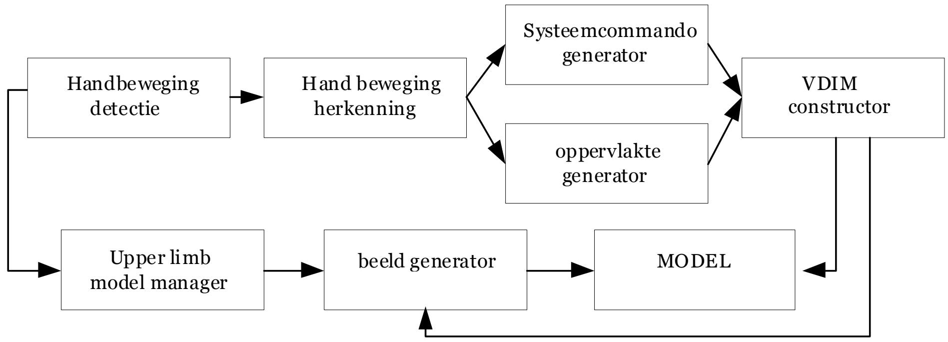
Figure 2 First implementation of the proposed design environment

Future research will focus on the setup of a smart proactive design environment applying several sensing technologies (Figure 2). For hand motion detection, a method should be developed for optimum camera placement according to the positions of multiple designers. The intelligent design environment could be further equipped with other human sensing technologies for person and situation recognition. The processing of the Hand Motion Language needs further improvement with a view to language and grammar processing, and synchronization of modeling actions indicated by either one or multiple designers. This implies a perspective change from information technology point of view, and suggests the investigation of the usage of intelligent and deliberate software agents as programming elements. Agents are communicating with each other and decide on their actions based on the situation and the context. Usefulness of multimodality was mentioned several times throughout this thesis. It is planned that the hand motion-based design interface is combined with for instance verbal control and physical object scanning as input means for conceptual design systems, but other modalities are to be investigated as well. As far as visualization is concerned, the hand motion input will be connected to a truly three-dimensional air-borne visualization device in the near future. The applicability of the hand motion-based interface will be investigated in other application areas, such as the fields of medicine and marketing as well.

SAMENVATTING

Sinds de introductie van de computer zitten ontwerpers opgescheept met de beperkingen van Computer Aided Design (CAD). Één - en tweedimensionale bedieningstechnieken dicteren de manier om vorm (geometrie) te modelleren met computersystemen. In de loop van tijd werd erkend dat deze traditionele user interface technieken teveel beperkingen opwerpen voor de conceptuele ontwerpfase (Computer Aided Conceptual Design - CACD). De belangrijkste beperking is de onhandige interactie, wat de creativiteit, de essentie van vormgeven, negatief beïnvloedt. Als resultaat van vier jaar onderzoek is een nieuwe methode ontwikkeld voor multimodale interactie. Deze nieuwe methode is gebaseerd op gebaren en heeft als doel vormgeving te ondersteunen met behulp van geavanceerde visualisatiesystemen. Bewegingen van de hand kunnen gelijktijdig vorminformatie en instructies omvatten, daarmee hoeft de ontwerper de 3D vormkenmerken plat te slaan ter wille van de computer. Mijn onderzoek heeft bewezen dat gebaren en handbewegingen kunnen worden gebruikt voor zowel de definitie van geometrie als het manipuleren hiervan.

De gezamenlijke visie van mijn collega's bij de leerstoel CADE was om een ontwerpomgeving voor industrieel ontwerpers te creëren, waarin gebarentaal of handbewegingen kan worden gebruikt om vormen uit te drukken; deze vormideeën zijn vervolgens op te slaan voor later gebruik of verfijning. Onze hypothese was dat dit systeem ontwerpers voorziet van een directe terugkoppeling en een intuïtieve manier biedt om vormconcepten uit te drukken.



Figuur 1 Handbeweging gebaseerde vormgeneratie

Onze aanpak was om een methode en een eerste implementatie te maken van een handbeweging-gebaseerde ontwerpomgeving voor het invoeren van geometrie, als onderdeel van een toekomstig systeem, genaamd Pro-active Reality Environment (PRE). Figuur 1 toont de processen om de handbewegingen te verwerken, de belangrijkste processen waren: (i) detectie van handbewegingen, (ii) interpretatie van de handbewegingen, (iii) generatie van geometrische entiteiten of systeemcommando's om de geometrie modeleringsmodule van juiste data te voorzien en (iv) het bijhouden van een digitale kopie van de arm van de ontwerper in de virtuele ruimte om deze te visualiseren. In Figuur 1 wordt uitgegaan van een ideale situatie, waarbij het beeld op dezelfde fysieke plaats geschiedt als de handbewegingen. Als dit niet het geval is, zal ook een virtueel model van de arm moeten worden gevisualiseerd.

De hoofdvragen van dit onderzoek waren (a) hoe kunnen handbewegingen of gebaren worden gedetecteerd (b) wat is de minimale set informatie om dit te doen, (c) hoe kunnen virtuele oppervlakken worden gegenereerd aan de hand van handbewegingen en (d) hoe kunnen vorm bewerkingen worden herkend? Het uitgangspunt van dit promotieproject was een raamwerk waarbij de volgende problemen zijn geformuleerd (i) het vinden van een systeem dat 3D posities snel kan meten en voldoet aan de ergonomische eisen, (ii) het modelleren van de arm waarmee de meetgegevens worden aangevuld en als zodanig een koppeling kan worden gemaakt tussen meting en modelleringmodule, (iii) omzetten van handbewegingen naar dubbelgekromde oppervlakken en (iv) het herkennen van vormbewerkingen uit gebaren. Voor wat betreft de herkenning van gebaren waren twee dingen van belang (1) zorg dragen in het onderscheiden van zinvolle van loze gebaren, (2) het ontwikkelen van een methode om dezen te herkennen.

Het probleem van het vorm modelleren door handbewegingen is complex, er is gekozen voor een systematische analyse. Tijdens de exploratieve fase van mijn traject speelden de volgende inzichten een belangrijke rol. (i) ik kwam achter het feit dat het onderzoek moest uitgaan van de menselijke maat: Een goed ontworpen gebarentaal was nodig, die niet alleen machinevertaling ondersteund, maar juist voor mensen begrijpelijk en aan te leren is. (ii) Een ander inzicht was dat de herkenning met hoge snelheid een brede variatie aan handbewegingen moest kunnen vertalen om de gebruiker directe terugkoppeling te bieden, (iii) de bewegingsdetectie moet zowel technisch robuust zijn als voldoen aan het ergonomische comfort, (iv) de uitgebeelde bewegingen moeten of als geometrische beschrijvingen of als aansturing van het systeem, (v) bij gebrek aan een echt 3D visualisatiemethode dient een digitale versie van de gebruiker te worden aangeboden om de virtuele ervaring realistisch te laten overkomen. Tenslotte leerde ik dat (vi) het herkenningssysteem moest worden gekoppeld aan een modelleringsysteem dat in staat is om de expressie van gebaren om te zetten in vaagheid.

Dit promotieproject was exploratief van aard. De volgende onderzoeksmethoden zijn toegepast: explorative studie, ontwikkeling van de handbeweging-gebaseerde bediening en onderzoek in toepasbase theorieën, proof-of-concept implementatie, experimenten op basis van de proof-of-concept en verificatie van het functioneren, validatie van de toegepaste theorieën en gebruiksonderzoek. Een uitgebreid literatuuronderzoek is uitgevoerd waarbij de nadruk lag op de verschillende manieren om gebaren en handbewegingen te verwerken. Leveranciers van zogenaamde "motion tracking" systemen (voor positie- en bewegingsbepaling) zijn benaderd, en na een kwalitatieve en kwantitatieve vergelijking van de opties is de best passende technologie uitgezocht. Verscheidene theorieën over gebaren herkenning en (anatomische) modellering van de menselijke hand zijn onderzocht en toegepast. Experimenten zijn uitgevoerd om de juistheid van de gebarenherkenning te verifiëren. De hieruit volgende theorieën werden getest op een indirecte manier door een handbeweging-gebaseerde bediening te ontwikkelen. Dit omvatte het ontwerp en de implementatie van proofs-of-concept van de algoritmen voor elke fase van de verwerking van de handbeweging. Een bruikbaarheidstudie werd uitgevoerd om de deze implementatie op een kwalitatieve manier te testen. Adviezen van de gebruiker werden verzameld door middel van vragenlijsten. Case studies zijn uitgevoerd om het praktische nut van handbeweging-gebaseerde vormconceptualisatie te testen. De resultaten van deze studies werden kwantitatief geëvalueerd door middel van statistische toetsen.

In het domein van multimodale interactie zijn handbeweging-gebaseerde systemen door vele onderzoekers voorgesteld, voornamelijk vanwege de mogelijkheid om vormen ruimtelijk uit te kunnen drukken. Bij de meerderheid van de resulterende systemen was de ontwikkeling van gericht op technologische haalbaarheid in plaats van menselijke aspecten, zoals begrijpelijkheid en leerbaarheid van de gebaren. Daarom heb ik een gebarentaal overgenomen die effectief zou kunnen worden gebruikt voor vormconceptualisatie. Deze reeks handbewegingen, genaamd Hand Motion Language (HML), is als basis voor de creatie en de manipulatie van geometrie gebruikt. De basiselementen van HML zijn woorden, en elk woord HML bevat ofwel geometrie, structuur en manipulatie voor vorm modellering, controle, of procedure-informatie voor het modelleringsysteem. Op basis van deze taal beschouwde ik handbeweging-gebaseerde vormconceptualisatie als een dialoog gebaseerd op woorden HML. Omwille van de duidelijkheid definieerde ik handbewegingen als opeenvolgingen van tweehandige configuraties van hand en arm die voortdurend in de tijd veranderen; dezen drukken of vorminformatie of een modelleringbewerking uit, and omvatten een bewegingstraject en 3D geometrische informatie. Drie separate processen zijn verantwoordelijk voor succesvolle verwerking van gebaren, namelijk (i) handbeweging detectie, (ii) de interpretatie van het gebaar en (iii) extractie van geometrische informatie.

Er wordt voldoende informatie over de bewegingstrajecten gemeten tijdens het volgen van de handbewegingen. Voor wat betreft de interpretatie van de gebaren was mij hypothese dat HML woorden geëxtraheerd kunnen worden door de bewegingstrajecten te segmenteren op basis van veranderingen in vingerzetting (de onderlinge stand van de vingers van een hand). Dit is mogelijk omdat eventuele veranderingen in de onderlinge posities van vingers volgens vaste patronen verloopt tijdens het uitbeelden van HML woorden. Zowel begin als eind van HML woorden kunnen hierdoor makkelijk worden herkend, en de opeenvolgende vingerzettingen hiertussen bepalen het betreffende modelleringcommando. Dit resultaat wordt met de bijbehorende geometrische gegevens doorgegeven aan modellering en visualisatie subsystemen. Ik heb na een uitgebreide technologie en marktscan voor een plaatsbepalingsysteem uiteindelijk gekozen voor een zogenoemd passief optisch 3D positiebepalingsysteem. Hierbij wordt er door infrarood-cameras de positie van retorreflectieve bolletjes (markers) bepaald; deze markers worden op specifieke plaatsen van de hand bevestigd, op een minimaal aantal plaatsen, nodig voor uitbeelden van oppervlakken en herkenning van HML woorden. Het 3D positiebepalingsysteem beperkt de bewegingsvrijheid van de ontwerper niet: het werkt draadloos en de kleine markers zijn lichtgewicht en comfortabel. Door het gebruik van zes camera's kunnen complexe tweehandige gebaren en armbewegingen worden gevolgd en kunnen de meetgegevens direct worden doorgegeven.

Een grondige analyse van HML was noodzakelijk om gebarenherkenning te implementeren. Hierbij bleek dat een HML woord twee uitdrukkingen bevat: transitie en statisch. Tijdens transitie verandert de vingerzetting voortdurend;

een statische expressie is bepalend voor het herkennen van HML woorden. Bij verdere bestudering van HML bleek een onderscheid tussen eenhandige, dubbelhandige en tweehandige HML woorden. Enkelhandige woorden betreffen een statische vingerzetting van één hand. Dubbelhandige woorden zijn in feite combinaties van enkelhandige woorden, waarbij beide handen tijdens het bewegen synchroon bewegen. Om de statische elementen van de HML woorden te bepalen is de herkenning van vingerzettingen noodzakelijk. Hiervoor zijn karakteristieke elementen van de vingerzetting geïdentificeerd. Dezen heb ik ingezet bij het ontwikkelen van de gebarenherkenning methode.

Ik heb alle mogelijke vingerzettingen van HML onderzocht om de meest onderscheidende element te isoleren. Dit leidde tot een zo klein mogelijke set van parameters. Deze zogenoemde inter-hand parameters specificeren de relatie tussen karakteristieke elementen van de vingerzetting. Om vingerzettingen te classificeren heb ik gebruik gemaakt van een beslisboom methode. Bovengenoemde parameters golden als invoer voor de beslisboom; gebaarherkenning is in feite een zoekactie door deze boomstructuur, die parameters in de knooppunten en vingerzettingen in de uiteinden (de “bladeren”) bevat. Het herkenningsproces omvat echter meer. Tweehandige HML woorden worden herkend door enkelhandige gebaren te combineren door middel van herkenningsregels; ter uitsluitsel van enkele ambigue gevallen zijn extra parameters toegevoegd, intra-hand parameters die de relatie tussen beide handen beschrijven. Het uiteindelijke resultaat kan een grote variatie aan gebaren herkennen terwijl een minimaal aantal beschrijvende parameters wordt gebruikt.

Ik heb twee verschillende detectie en modelleringomgevingen ingezet tijdens mijn promotieproject. Eén was gebaseerd op bovengenoemde passieve optische 3D positiebepalingsysteem, uitgerust om gebaren van meerdere ontwerpers te volgen in een kantooromgeving. Een reguliere (2D) monitor was gebruikt als visualisatie. De andere omgeving was opgezet als experiment om het concept van gebaarherkenning te toetsen. Hierbij werd gebruik gemaakt van een paar datagloves en magnetische positiebepaling en een 3D display met speciale stereobril. Dit systeem was een betrouwbare testomgeving, die echter maar door één gebruiker tegelijkertijd kon worden bediend. In dit geval was de HML herkenningssoftware geïntegreerd met het VRJuggler ontwikkelplatform, dat het gebruik van datagloves en 3D displays al ondersteunde. Echter, onze lange-termijn visie is gericht op het inzetten van passieve optische 3D positiebepaling technieken, welke meer bewegingsvrijheid geeft en flexibeler zijn in te zetten.

Deze optische techniek biedt vooral voordelen in combinatie met holografische of andere zogenoemde autostereoscopische beeldtechnieken.

Aan de hand van het passieve optische 3D positiebepaling systeem was een software module ontwikkeld om een virtueel model mee te laten bewegen en af te beelden. Dit model was in ons geval het Upper Limb Model (ULM) – welke werd gekoppeld aan de karakteristieke plaatsen die waren geïdentificeerd voor gebarenherkenning. Het 3D positiebepalingsysteem wordt door deze module opgestart en met behulp van de gerelateerde Software Development Kit (SDK) worden de gegevens doorgeven aan mijn eigen ontwikkelde software, Identificatie van de optische markers en relatie tot de karakteristieke plaatsen geschiedt op basis van het ULM. De SDK communiceert met de gebarenherkenning module, welke het beslisboom algoritme uit de See5 software bibliotheek aanroept. Deze beslisboom wordt geconstrueerd op basis van de eerder genoemde inter-hand parameters. De resulterende beslisboom werd geïntegreerd met eigen software voor de verdere verwerking van de handbewegingen.

Samenvattend was de informatiestroom van de geïntegreerde software als volgt: 3D metingen uit het optische 3D positiebepaling systeem werden door de SDK gecondenseerd tot een klein aantal parameters; deze werden door de beslisboom omgezet in gebaren voor linker- en rechterhand. Segmentatie van handbewegingen is gericht op het vinden van statische vingerzettingen; dit geschiedt door opeenvolgende vingerzettingen van beide handen te vergelijken. Wanneer deze een bepaald aantal metingen gelijk blijven, wordt automatisch het herkenning algoritme in een separaat proces aangeroepen. De herkenningsprocedure is snel, dit door de efficiënte beslisboom categorisatie en het inzetten van enkele heuristieken op basis van intra-hand parameters. Uiteindelijk worden HML word en gerelateerde geometrische informatie doorgegeven aan de modelleringmodule, in ons geval de Vague Discrete Interval Modeler (VDIM).

Op basis van literatuur onderzoek zijn de verschillende handbeweging verwerking technologieën verdeeld in vier verwerking categorieën. Deze zijn categorieën zijn de volgende namen toegekend: (i) Direct incompleet, (ii) Direct compleet, (iii) Indirect incompleet, (iv) Indirecte complete verwerking. Direct en indirect verwijzen naar de manier waarop informatie wordt vertaald van de fysieke ruimte, waar de handbewegingen plaats vinden, naar de virtuele ruimte waar de vorm wordt gemodelleerd. Indirecte data vertaling wil zeggen dat er gebruik wordt gemaakt van een actief hand model om de waargenomen data aan te vullen. De handen van mensen kunnen compleet gescand worden of sommige karakteristieke punten (zoals karakteristieke punten of schaduwen) kunnen worden gedetecteerd. Deze twee manieren om informatie ten aanzien van de vormen van bewegende handen te genereren, kunnen worden geïdentificeerd als complete en incomplete informatie extractie. Deze vier categorieën zijn verder uitgewerkt op basis van de relatie die er bestaat tussen de handen and de apparatuur dat de informatie extraheert. Bepaalde apparatuur wordt geplaatst op de hand of worden in contact gebracht met de hand, terwijl ander apparatuur informatie van op een afstand kan extraheren. Deze relaties wordt beschreven als contact en non-contact. Het is gebleken dat direct compleet en indirect incompleet handbeweging processing technologieën de potentie hebben om op handbeweging-gebaseerde conceptualisatie te ondersteunen. Wat betreft de relatie tussen de handen en de detectie apparatuur, worden non-contact technologieën geprefereerd. De redenen hiervoor liggen in de ergonomische eisen. Echter, na intensief en uitgebreid markt onderzoek is gebleken dat de huidige detectie technologieën niet capabel genoeg zijn om aan al de technische en ergonomische eisen te voldoen. Een compromis zal moeten worden bereikt in op basis van kleine en lichtgewicht passieve markers.

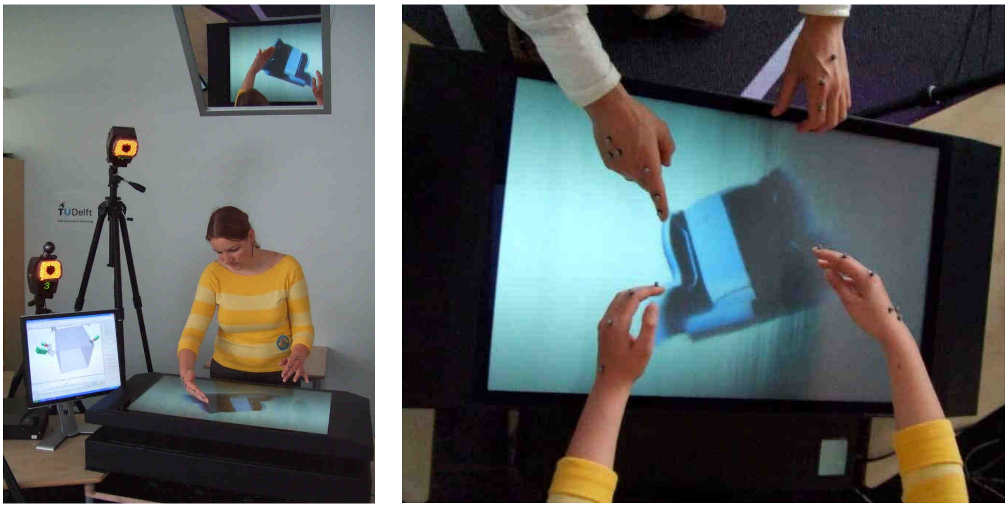
Ik heb een aantal opstellingen ontworpen en om daarmee experimenten uit te voeren met als doel om de betrouwbaarheid van de gebarenherkenning methode te testen. Het resultaat toonde aan dat als alle gegevens beschikbaar zijn voor de beslisboom algoritme, de accuratesse van HML word herkenning 100% is. Uit studies is gebleken dat de generatie van modelleringcommando's gerelateerd aan HML woorden extra geometrische informatie nodig hebben en waarbij elke HML woord een individuele verwerking vereist. Echter, de verwerking van hogere dimensionale entiteiten als de oppervlakten van objecten, kunnen terug getraceerd worden naar lagere dimensionale entiteiten als punten. Dit maakt hergebruik van de algoritmen die gebaseerd zijn voor het verwerken van punten mogelijk. Tevens worden de algoritmen op een orde van *O(logN)* gehouden.

Voor de evaluatie van het principe om met handbewegingen te vormgegeven, is er een gebruikstest ontworpen en uitgevoerd. Hierbij werden op HML gebaseerde modellering sessies vergeleken met conventioneel CAD gebruik. Er is geconcludeerd dat deelnemers de HML methode beter vinden dan de traditionele CAD methode voor het conceptueel ontwerpen van vormen. Vooral de categorie van leerbaarheid vertoonde een significant verschil in het voordeel van het op HML gebaseerde systeem. De categorieën operationaliteit, stimulatie and satisfactie vertoonden ook een aanzienlijk verschil. Deelnemers (i) waren beduidend sneller in het creëren van conceptuele vormen, (ii) ervoeren de handbeweging bediening als meer intuïtief en (iii) waren meer tevreden. Dit betekent dat zij eerder geneigd zijn deze innovatieve interactie methode te gebruiken. Het is gebleken dat de HML gebaseerde modellering mensen fysiek eerder vermoeid dan de traditionele CAD. Er werd ook geobserveerd dat deelnemers een variëteit aan vormen konden maken met de HML gebaseerde modellering waarbij hun fantasie werd gebruikt. Zij waren niet geneigd gemakshalve de voorbeeld opgave te kopiëren die aan hen werd getoond, maar creëerden andere vormen door hun fantasie te gebruiken. Aan de andere kant, wanneer gebruik werd gemaakt van CAD waren de deelnemers gericht op het zo goed mogelijk afronden van de taak die aan hen was toebedeeld en lette men niet op de originaliteit van hun werk. Het resultaat hiervan was dat de uiteindelijke vormen gelijk aan elkaar waren. Mijn persoonlijke ervaring was dat verschillende typen mensen divers reageerden op de verschillende modellering methoden. Deelnemers die actiever, creatiever en nieuwsgieriger van aard leken prefereerden de HML meer dan de deelnemers van passieve aard. Eerstgenoemden waren ook sneller in het leren van de HML software. Aan de andere kant hadden nerveuze deelnemers moeite om de HML gebaseerde modelleringsoftware te gebruiken door hun snelle handbewegingen en hun plotselinge emotionele reacties als er iets mis ging.

Een ander experiment was ontworpen om de bruikbaarheid van handbeweging gebaseerde vormgeving te testen. De criteria die hiervoor werden vastgesteld waren de betrouwbaarheid van de gegenereerde vormen, de nauwkeurigheid van de vorm manipulatie en de complexiteit van het modeleringsproces voor het geval van verschillende vormen. In het realistische experiment werden deelnemers gevraagd om cirkels en cilinders te vormen door middel van handbewegingen. Ze werden gevraagd om deze vormen in verschillende oriëntaties and maten te maken. Mijn algemene observatie is dat kleinere vormen die werden gemaakt voor een oriëntatie gekozen door de deelnemers zelf, een hogere overeenkomst vertoonden.

Het experiment gericht op nauwkeurigheid betrof twee opgaven. Deze waren het vaststellen van de grootte van een object en het vaststellen van de positie van een object. Dit experiment was gericht om de maximale nauwkeurigheid is voor manipulatie wanneer de tijd buiten beschouwing wordt gehouden. Deze nauwkeurigheid is ongeveer 7 millimeter. De complexiteit van de vorm heeft een grote invloed op het totaal aantal modelleringsancties. Het doel van het laatste experiment was om dit aspect te onderzoeken. Het is gebleken dat de grootste voordelen van handbeweging-gebaseerde modellering zijn: het maken van dubbelgekromde oppervlaken en het feit dat objecten direct gepositioneerd kunnen worden in een drie dimensionale (3D) ruimte.

Als laatste conclusie wil ik vermelden en tevens bevestigen dat een doelgericht ontworpen gebarentaal niet alleen intuïtief en plezierig is voor ontwerpers, maar ook de creativiteit stimuleert. Aangezien creativiteit een belangrijk onderdeel is voor vorm conceptualisatie, is een handbeweging-gebaseerde bediening extreem nuttig en praktisch in de conceptuele ontwerpfase. Voor wat betreft het herkennen van gebaren kan ik concluderen dat de HML woorden herkend kunnen worden door middel van het bestuderen van veranderingen in vingerzetting. Met deze aanpak kunnen bruikbare onderdelen uit handbewegingen worden herkend als gebaren middels een klein aantal beschrijvende parameters in een beslisboom. De formele Hand Motion Language bevat geometrie, structuur en manipulatie voor vorm modellering, controle, of procedure-informatie voor het modelleringsysteem. Zowel systeemcommando's als geometrische beschrijvingen kunnen worden zonder vertraging worden herkend met het passieve optische 3D positiebepaling systeem. Een minimale set van karakteristieken stuurt zowel de gebaarherkenning als het genereren van geometrie aan. Echter, zelfs de beste ontworpen gebaren bediening wordt zinloos als het wordt aangesloten op een conventioneel CAD systeem. Handbewegingen zijn vooral nuttig tijdens de conceptuele ontwerpfase, om te ondersteunen bij samenwerking tussen ontwerpers en het uiten van vaagheid in vormen. Huidige CAD systemen kunnen deze activiteiten niet ondersteunen. Deswege heb ik de handbeweging-gebaseerde bediening gekoppeld aan de Vague Discrete Interval Modeler om de vage informatie die van nature aanwezig zijn bij handbewegingen te ondersteunen. De huidige twee dimensionale beeldtechnieken niet toepasbaar zijn. Het gebruik van de welbekende stereobril voor driedimensionale visualisatie bleek goed te werken. Hierbij moet er wel rekening gehouden worden met het feit dat het ultieme doel in de lucht zwevende ruimtelijke beeldtechnieken zijn. Technieken als electroholografie maken het mogelijk om meerdere ontwerpers te laten werken aan éénzelfde concept en dit vanuit verschillende hoeken te kunnen bekijken. Met het op de markt introduceren van ruimtelijke visualisatie worden ontwerpers in de gelegenheid gesteld om virtuele oppervlakten te maken precies in dezelfde ruimte waar de fysieke handbewegingen plaats vinden.



Figuur 2 Eerste implementatie van de ontwerpomgeving.

Toekomstig onderzoek zal zich richten op het creëren van een Proactive Reality Environment waarbij verschillende positiebepalingtechnieken worden toegepast (Figuur 2). Voor handbeweging detectie zal er een methode ontwikkeld moeten worden voor de optimale positionering van camera's gebaseerd op de positie van verschillende ontwerpers in de ruimte. De intelligente ontwerpomgeving zou verder uitgebreid kunnen worden met het inzetten van andere sensoren voor persoonsidentificatie en situatieherkenning. Verbetering van de Hand Motion Language verwerking zijn gepland op gebied van taal en grammatica verwerking en synchronisatie met modelleringacties van meerdere gebruikers. Daardoor ligt het voor de hand om over te schakelen naar zogenoemde intelligent software agents. Zulke software agents communiceren met elkaar en nemen beslissingen op basis van de situatie en de context. Het gebruik van multimodale communicatie is meerdere malen naar voren gekomen in deze dissertatie. Een combinatie van gebaren met spraakbesturing en inscannen van fysieke vormen ligt voor de hand, maar andere modaliteiten dienen ook onderzocht te worden. De handbeweging bediening zal binnenkort worden gekoppeld aan een holografisch driedimensionaal display. Voorts zal het gebruik van gebaren bediening ook kunnen worden toegepast in andere domeinen, bijvoorbeeld de medische sector en marketing.