Free Form Sketching System for Product Design Using Virtual Reality Technology

Hang Yu

A Thesis
in
the Department
of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

March 2005

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-494-04435-7
Our file Notre référence
ISBN: 0-494-04435-7

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

ABSTRACT

Free Form Sketching System for Concept Design Using Virtual Reality Technology

Hang Yu

This thesis discusses a new, Virtual Reality (VR) based sketching method that enables product designers to display their ideas in graphical forms directly in VR domain rather than using paper based sketching platform. General product design includes two main parts: conceptual design and mechanical design. Mechanical design is well studied and can effectively be done using some of the available software such AutoCAD, CATIA or ProE. Conceptual design like automotive styling on the other hand requires an environment where an industrial designer or an artist can express his/her ideas and display them in visual forms in an intuitive and unconstrained way. Since aesthetics and functionality of the product is addressed with equal importance, the conceptual design stage is considered as an art rather than engineering.

Computer Aided Design (CAD) systems are inadequate for conceptual design process particularly for sketching. Conventional paper sketching is commonly used for catching the initial inspirational imagination of designers. It is difficult for designers and decision-makers to understand stereoscopic impressions with a two-dimensional (2D) representation. Yet, such designs require a time-consuming process to replicate 2D sketches in computer, in three-dimensional (3D) form using one of the available CAD software. Hence, the demand for a system where the initial design process is performed directly in computer so that the design modifications and the format changes can be done efficiently has been increasing among designers. In this thesis, a free form conceptual design system that allows designers to perform conceptual design directly in computer is introduced. In this method, surfaces of products are created and modified directly and intuitively through control points in 3D space. The artist performs the design process freely and unconstraintly in 3D space by operating his/her hands wearing data gloves. In our system, conceptual design is simply realized in computer.

Dedicated to my parents…

ACKNOWLEDGMENTS

I could not have completed this thesis without the assistance of many people who helped me either directly or indirectly. First and foremost, I would like to express my thanks and indebtedness to my supervisor Dr. Ali Akgunduz for his constructive technical advice, financial support, constant guidance, and encouragement throughout this work. The discussion he had with me relating the thesis played the most significant role in writing this thesis. I appreciate the time he devoted to help me understand the concepts involved in this thesis.

Without the continuous support of my family and friends, it would not be possible for me to successfully complete all the complex works and program. I thank my parents for their life time support that came from the far northeast of China. I also thank my brother for his endless inspiration towards my graduate studies. I express my gratitude to all my friends who help me reading the proof of my initial report. I am grateful to all faculty members and staff of the Department of Mechanical and Industrial Engineering at Concordia University for contributing to my education, providing me an excellent work environment and making my studies and work at Concordia very enjoyable.

Finally thanks to all those people who have not been mentioned here but who have helped me during my masters.

Hang Yu

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

VR	Virtual Reality
CAD	Computer Aided Design
2D	Two Dimensional
3D	Three Dimensional
IT	Information Technology
HMD	Head Mounted Display
DOF	Degree-of-Freedom
ARCADE	Advanced Realism Computer Aided Design Environment
VE	Virtual Environment

Chapter 1

Introduction

Innovative products and shorter time-to-market are more and more essential for the competitiveness of companies. It is therefore very important for the industry to develop new methods and technologies to support the product development, production planning and other processes concerning the life-cycle of new product. VR is just one of the emerging technologies. Its popularity in design and manufacturing has been increased significantly in recent years. VR offers a new quality of presentation and interaction in a virtual 3D world in real-time. The virtual world is a computer-generated space. To be of any use in practical VR applications, the user must be able to interact with the virtual world like pressing a button to turn on light or opening the window to watch the flowing river. The virtual world can be programmed to simulate almost anything based on creative mind since there are almost no limits to artistic freedom. The domain and the scope of VR is very broad and may be seen as inclusively surrounding many other related Information Technologies (IT). VR is not only applied in the entertainment industry, but also in the fields of engineering and design, manufacturing, medicine, and training. The application of VR to industrial area is just at its beginning. As shown by several studies and sample applications from both research and industry, there is a great potential in this area.

1.1 Problem Area

The typical product development cycle includes several interrelated steps. The recent philosophy, known by the name of concurrent engineering, outlines the steps and relationships between these steps in order to develop successful product designs. In general, a product design starts with inputs from customers, competitors, future trends and available technologies. The professional figure must take into account the market and user requirements to determine the aesthetic and visual impact of the product in order to give it the added value and desirability. Once overall pictures of current needs are determined, ideas are translated in geometric forms using various sketching techniques. After the desired shapes are achieved in sketches, these ideas are then translated into CAD format using products such as CATIA, ProE, AutoCAD etc. Today, CAD software provides not only 3D representations of objects, but also enables engineers to perform various analyses. For instance, finite element analysis, assembly and disassembly simulations, manufacturability tests are possible in most CAD software. However, it is unfavorable for conceptual design which demands inspirations of designers in the early stage of the design process.

VR has been successfully adapted in product design for testing human-product and human-manufacturing interactions. Since product and manufacturing environment can be presented in the same 3D world in VR, it has become one better choice for prototyping.

Since physical mockups are time-consuming, expensive and un-flexible comparison to 3D computer models, the popularity of VR for prototyping is increasing. Current trend shows that in near future, VR will be adapted in most design and manufacturing enterprises.

Although recent developments in both CAD and VR tools in both software and hardware level helped companies to reduce cycle time of product development, the initial part of the design process, namely sketching, is still being done in conventional way using pencil and paper. Final sketches still require a long and tedious process to create their 3D replicas in computer using CAD software. In order to represent sketches into a digital format, generally some mathematical knowledge is needed. Splitting of curves, number of control points and order of curves should be taken into account by the user. Such approach represents a concrete limitation to the free expression of ideas and alternatives evaluation, typical of conceptual design. In practice, design process consists of several repetitive phases. One may require several new sketches and 3D drafting due to unexpected constraints and requirements arising in later stages. Changes in sketches are the least flexible. Mostly, it requires starting from the beginning at each time. For this reason, there has been an increasing demand for a system where all aspects of the design are addressed in computer without facing any major constraint.

In today's competitive marketplace, companies are forced to respond to the requirements of their customers and market itself faster than ever. Quick respond to the market becomes the most important factor for many companies to survive in their businesses.

This is particularly true in high-tech industry. Hence, the efficiency of the product design process sometimes determines the failure or success of a company. The highly competitive nature of today's business and advancing technologies such as VR challenges many researchers and companies to find methods to reduce the time spent in conceptual design phase. Use of VR technology is particularly effective in product design process due to its immersive nature. Yet its current use is limited to product testing, verification, process planning and machining simulations. The potential of VR to be utilized in 3D sketching is yet to be explored. Free form sketching directly in 3D world which is the only non-computerized part of today's design practices is possible using VR technology. The whole geometric design phase can be natural and unconstrained for designers or artists due to the immersion enabled in VR. Furthermore, conversion to desired CAD format where various tests are possible is much more efficient since the version in VR is already in digital format.

1.2 Objective

VR technology is a candidate platform where conceptual design, particularly free-form sketching of product geometrics, can be performed. Understanding of designers' intention while sketching the initial product imaginations in the conceptual design stage motivates the work described in this thesis. In this research work, our goal is to develop a VR system where industrial designers or artists can translate their imaginations into geometric forms freely via an unconstrained way directly in 3D space. The major problem an artist may face using CAD software to design a product is that the artist has to spend considerable amount of time and energy to deal with properties of CAD. In

CAD, objects are mostly created by connecting various vertices selected by the designer. There are number of rules to generate a line or a curve in CAD software. Yet, there are more rules to connect separate splines, or adding, subtracting new geometries to the existing ones. Such well defined and governed by several constraints platform forces an industrial designer or artist to focus on properties of geometries rather than the design itself. On contrast, the designer should spend his/her time and energy for imagination and transforming these ideas into geometric forms. As a consequence of the constraints in the existing CAD software, most designers' choice in conceptual design stage is paper and pencil based platform.

We believe that VR technology can provide a design tool in which the desired freedom is enabled and properties of geometric design are ignored. No mathematical foundation of curve and surface is required in such a platform. It allows designers to deliver their imaginations into geometric form as they are using paper and pencil. Further more, initial sketches are directly stored in computer so they can be saved for future use. Shapes can be modified for detailed analysis such as assembly and disassembly, manufacturability, finite element, and these geometries can be exported into desired CAD formats.

The VR based sketching tool described in this thesis reduces the time spent for product development process which is critical for companies' success. Shorter product development time gives companies a competitive edge in today's highly competitive marketplace.

1.3 Contribution

With our system, the conceptual design process can simply be done in computer which means that the whole development cycle of a new product including concept design and engineering design can all be finished in computer. Several months of time on full-size mock-ups is cut off and the process of product development is greatly shortened. VR technology can ensure that the product is well designed by incorporating aesthetics, engineering, manufacturing, and serviceability issues simultaneously. Real-time interactions with the potential customers are also possible during the design stage. Proposed design model has great potential to same time and money for companies. Better product and faster development process increase competitiveness of companies in today's demanding marketplace.

The remainder of this thesis is organized as follows: In Chapter 2, a summary of earlier works in VR and product design is introduced. Chapter 3 describes our free-form sketching system in details. In chapter 4, some examples created by using our system are resented. Finally, chapter 5 concludes the thesis with suggestions for future work and possible improvements.

Chapter 2

Background Information and Related Work

VR technology has been utilized frequently in product testing, performance analysis, consumer research and collaborative product development areas. The major advantage of the VR technology over the traditional simulation techniques is its potential to simulate both users (such as a customer or a designer) and products in a common interactive environment. This virtual encounter is becoming much more powerful and realistic with the availability of advanced collision detection algorithms, data-gloves and haptic devices that are essential tolls in order to perform VR functionalities such as touching, grabbing, feeling the surface of product etc. Successful development of a VR system relies on implementation of diverse technologies. The components of VR should seamlessly interact and cooperate with each other in order to achieve the realistic immersion and quality of interaction. Some major technological issues that mostly affect the design and performance quality of a VR system are briefly discussed in this chapter.

2.1 Virtual Reality

VR is a computer generated realistic world where both computer images and real objects such as human and other physical surroundings interact in real-time. Moreover, the synthesized world can respond to user inputs such as hand gestures, verbal commands, etc (Hauptmann, 1989). Some researchers have referred to VR in terms of its tools but not its purpose and functions. For example, people tend to associate VR with Head-Mounted Displays (HMD) and data gloves. However, VR simulations can be generated without HMD. Instead, some other displays such as large projection screens or desk-top graphic workstations can be used for this purpose (Sandin et al., 1991). Similarly, data-gloves can be replaced with much simpler trackballs, joysticks or even simple mouse. Conversely, data-gloves can be used in other areas such telerobotics (Card et al., 1990). Therefore describing VR in terms of some popular tools used in it is not an adequate approach.

Real-time performance and human-computer-graphic interaction are the main characteristics of the VR. Since first described as “Ultimate Display” by Ivan in 1965, VR required several years for people to take advantage of this immersive computer graphic technology. In 1994, with the invention of CAVE by Carolina Cruz-Neira, Dan Sandin, and Tom DeFanti at University of Illinois at Chicago (UIC), both scientific world and industry saw the potentials of VR in applications varying from military to product design and testing. Although many companies and research institutions were captured by the potentials of the VR, until early 2000, other than automobile and entertainment industries, VR’s potentials were not adequately utilized by other industries (P.Frederick, 1999). This was due to the technological constrains. Although technologies still limit VR to be utilized with its full strength in industrial applications, major breakthroughs have been accomplished in recent years. Today many major corporations such as Caterpillar, General Motors, Ford, CAE, Daimler Chrysler etc. installed VR systems similar to the CAVE. Not only in manufacturing, VR also found significant roll in medicine, physiology, space exploration, etc.

The quality of VR applications is judged by their abilities to replicate real life events and surrounding environments. Real life activities involves seeing, touching, talking, modifying the surroundings in real-time. Therefore a quality VR application should enable a person to be present in an environment where he/she can see and touch the surroundings, talk to others and modify the environments as the nature of surroundings objects allow. Such interactions with the computer generated worlds are only possible with the use of special devices such as special mousse (wands or data-gloves), trackers (position and rotation), special glasses (stereo-glasses, HMD), 3D sound systems, powerful computers, art graphic cards and special software. Although the sense of smell and taste are also desired in VR simulation, yet the technology is far from achieving such goals. The important components of VR are briefly described below.

2.2 Components of Virtual Reality

For human-computer interaction, it is necessary to use specially designed tools both to allow input to the computer and to provide feedback from the computer to the user. Today's VR tools are varying in functionality and purpose as they address different human sensorial channels. For example, body motion is tracked by 3-D position sensors, hand gestures are digitalized by data gloves, visual feedback is transmitted through stereo displays, virtual sound is computed by 3-D sound generators, scene perspective and orientation is changed with track-balls and joysticks, etc. Some of these I/O devices are commercially available and some are still in development stage. The following section introduces a number of VR I/O tools with their functionality and design.

2.2.1 Stereo Display

The immersion which is the key characteristics of VR is possible using stereo displays. Today stereo images can be generated in many different platforms (Buxton et al., 2000). Simply a PC monitor can provide the dual images to be used in VR applications. On the other hand, some other hardware such as HMD is specially designed for stereo imaging. Some of the special VR displays are summarized below:

- Head Mounted Displays: Two small monitors, one for each eye, alternatively display the same image with small rotational changes in order to generate the sense of 3D. Although it is flexible and can generate real-size images, the field of view is still its major drawback. The HMD we use at Concordia is shown in Figure 2.1.
- CAVE: It is a square room about 10 feet long that can display images on up to 5 walls. The person using the system can experience the complete immersion. Although the images were initially generated in Silicon Graphics Onyx Workstations, today powerful PCs can be used to achieve the similar performances. Dr. Ozell from École Polytechnique demonstrated the feasibility of CAVE powered by PCs. The major drawback of CAVE is its price. Comparison to 1994-1995, the cost of installing a CAVE system decreased more than half, yet

it still requires substantial amount of investment. See Figure 2.2 for the illustration of CAVE.

- Physical Mockups: This is very specialized VR environments mostly build in airplane, helicopter, automobile, ship or submarine simulators. Montreal based company called CAE is one of the major developer of such devices.
- Other large VR displays: Since the invention of CAVE, many other institutions developed their own displays. Surround-Screen Visualization Environment developed by Cruz-Neira whom also worked on the CAVE project while she was a graduate student at UIC. ImmersaDesk, PowerWall and CRUVE are some other VR solutions developed by FakeSpace whom is also the owner of CAVE system. Several extensions to HMD are also available. Boom-Mounted displays are another popular VR displays available for research and development.

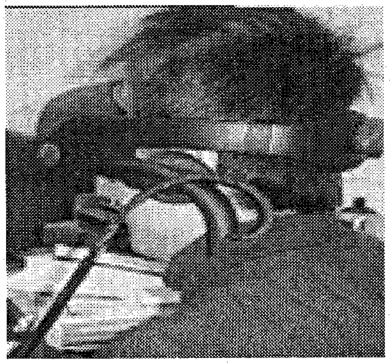

Figure 2.1: Cybermind hi-900 resolutions HMD.

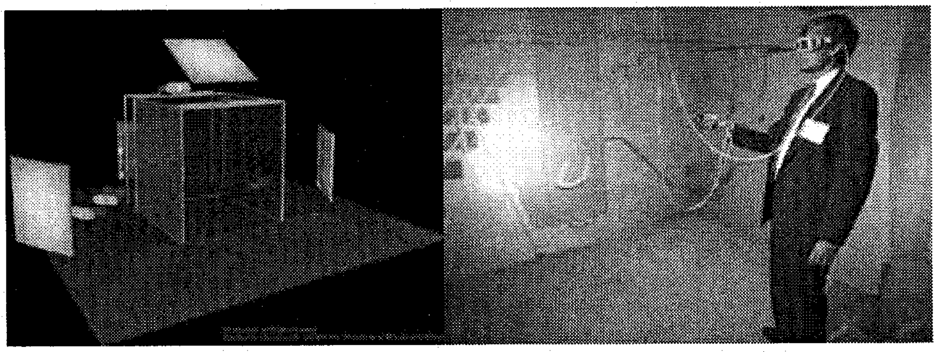

Figure 2.2: Configuration of CAVE Automated Virtual Environment (left) and a CAVE simulation (right).

2.2.2 Position Tracker

Many applications such as robotics, biomechanics, architecture, product development, education etc. require the knowledge of the real-time position and orientation of moving objects with respect to stationary ones. In order to sustain the desired realism in VR, tracking the location of dynamic object particularly human motions (head and hand) is indispensable (Cai et al., 1995). There are several tracking solutions available for VR use. Human motion and location of other physical objects in a VR simulation can be tracked by using sonar trackers, magnetic trackers, laser trackers and camera images. Figure 2.3-a shows a 6-Degree-of-Freedom (DOF) tracker from 5DT Technologies and its applications. Figure 2.3-b shows the application of trackers in CAVE. Most VR applications demand both position and rotation information of tracked objects. A mobbing object in 3D space has three translations as well as three rotations, as illustrated in Figure 2.3. The 3D measurements should not limit the freedom of moving objects, especially users' motions.

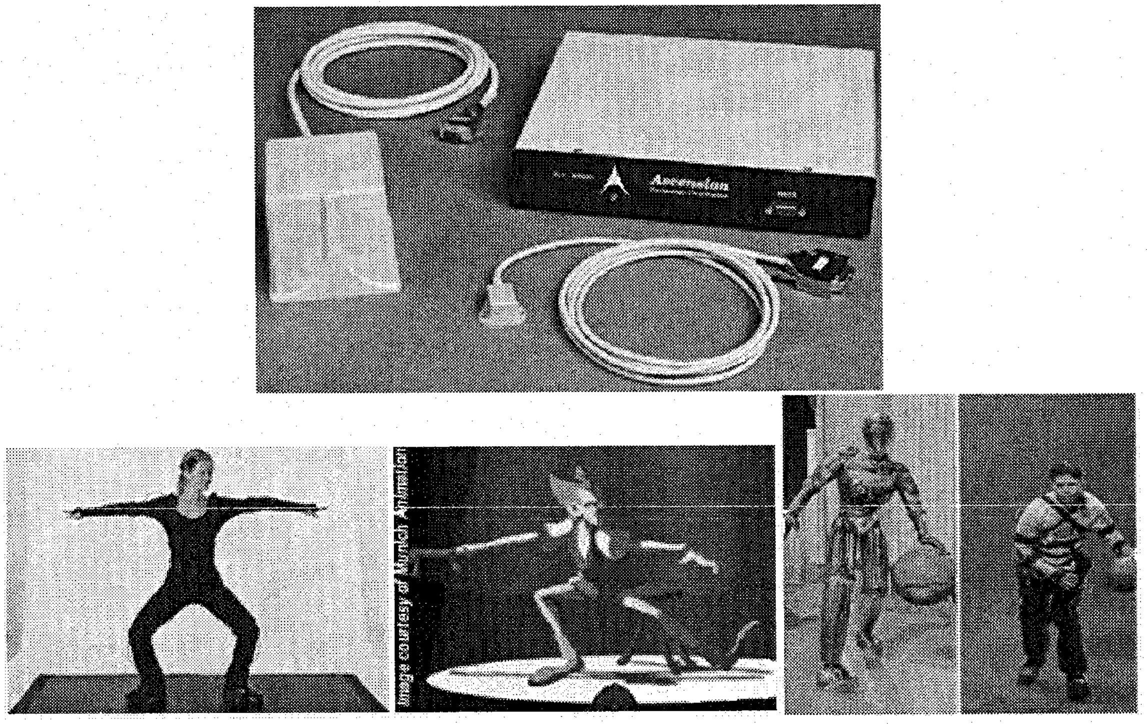

Figure 2.3-a: Ascension Flock of Birds Class B 6-DOF trackers and applications from 5DT Technologies.

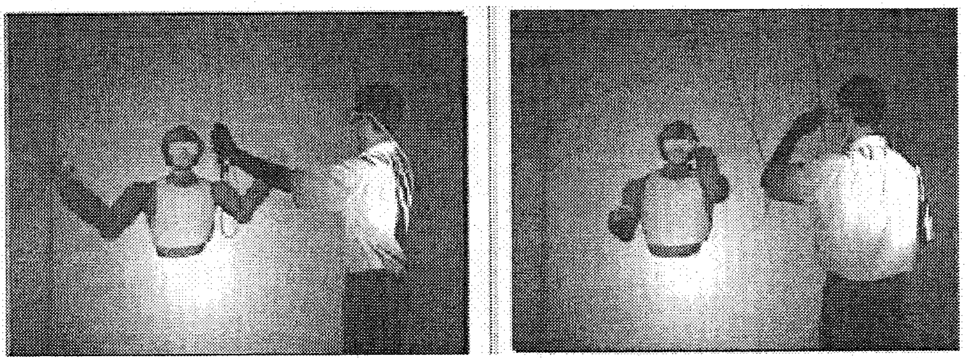

Figure 2.3-b: Tracking user motion in CAVE

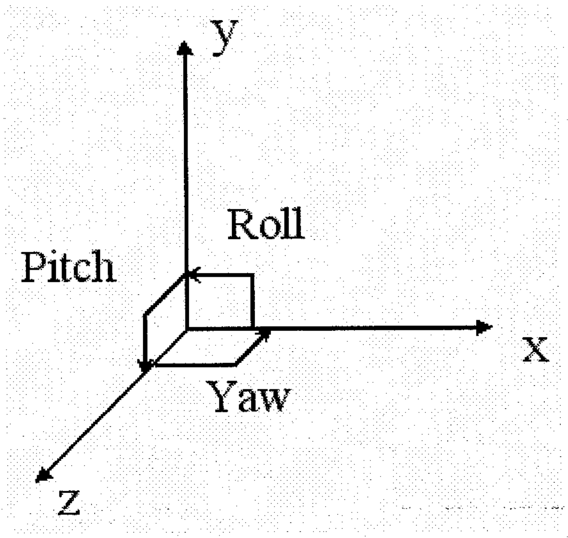

Figure 2.4: System of coordinates of a moving object.

2.2.3 Data Glove

Animation of the hand is necessary in VR interaction because most operation is directly done by user's hand (Popescu et al., 1999). Data-gloves are used to obtain relationships of joints belonging to a hand. Some of the commercial models are VPL DataGlove, Vertex CyberGlove, Mattel PowerGlove, and Exos Dextrous Hand Master.

In our system, we use a pair of 5DT Data Gloves, provided by "Fifth Dimension Technologies" (see Figure 2.5 for illustration). Each glove can output seven data. Five of them are for the flexure of each finger and the other two are for the pitch and roll of the palm. Because there is just one value for each finger, the joint angles on each finger is set to same value in our system. We input two sets of the seven data into our system for the virtual hands to animate the movement of the real hand.

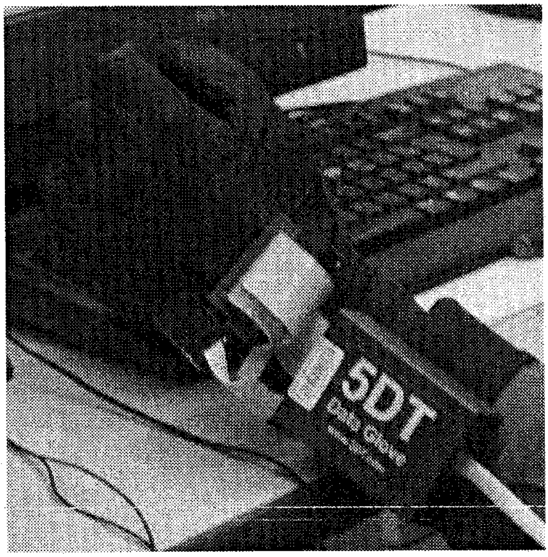

Figure 2.5: 5DT Data Glove 5.

2.2.4 Touch and Force Feedback

Touch and force feedback are important sensorial channels for immersive feeling. Enabling of tactile feedback in a VR simulation can make people to explore the surrounding environment to get identifications of objects, positions and orientations, to manipulate or move an object to perform a task (Dionisio et al., 1997). The human touch and force feedback are different in physiology and in control requirements. Information on contact-surface geometry like on a flat surface or on an edge, the smoothness of the contact surface, its temperature, or even a grasped object's slippage due to gravity can be provided by touch sensors. Conversely, force feedback gives information on the total contact force, on contact-surface compliance (soft or hard), or grasped object weight (heavy or light). In practical application, force and touch feedback are not mutually exclusive but complementary. When people manipulate objects, the weight and hardness as well as surface smoothness and geometry are felt simultaneously. For a VR feedback device, it is better to have both force and touch feedback (Burdeau et al., 2000). Phantom haptic device by SensAble Technologies is shown in Figure 2.6.

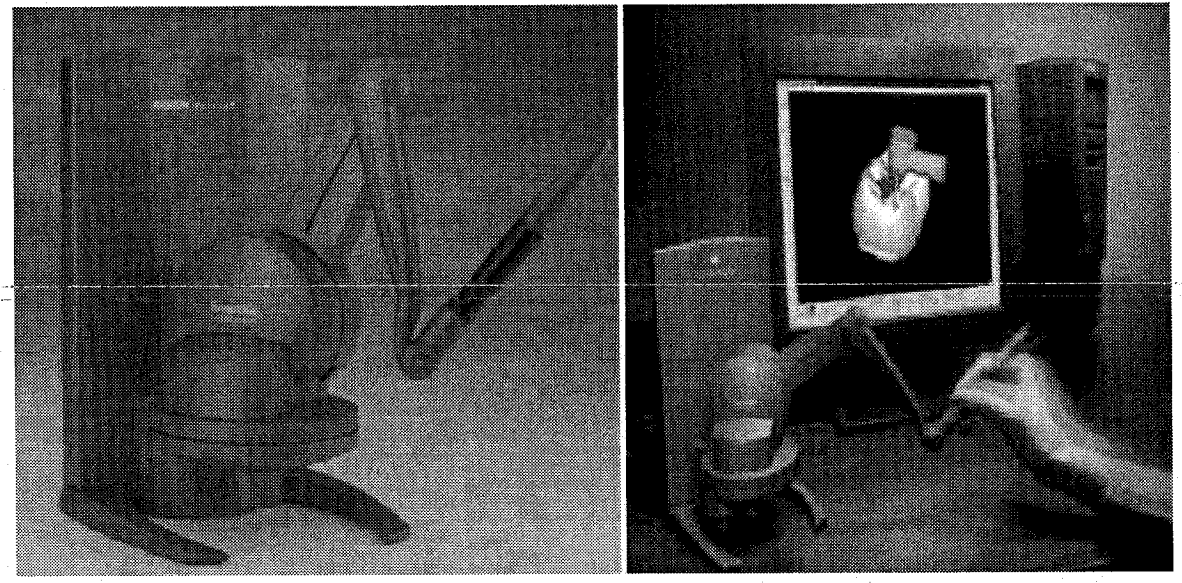

Figure 2.6: PHANTOM Force Feedback (haptic) devices and applications on dentistry.

2.3 Virtual Reality in Design

In the course of industrial product development like a car (Tovey et al., 2000), majority of decisions about the design, the user interface, functionalities, the mechanical construction, the planning of production and assembly lines, and the maintenance of the product have to be taken into account at the early stage of the development cycle. Conventionally, these decisions are made independently at various stages of the product development cycle. Such practices cause several unnecessary loops between different stages of design process. This cumbersome design approach requires sketches and physical prototypes to be modified several times in order to address all the design issues whenever they are faced. Although physical prototypes help engineers and designers to identify design problems at early stages of the design process, building physical prototypes is expensive, time-consuming, and un-flexible when modifications are required. They are also not suitable for comparing alternative solutions. Furthermore, use of prototypes may not assist engineers to evaluate manufacturability, assembly and packaging issues. Manufacturability, assembly or packaging issues may only be identified at later stages of practical tests or during actual manufacturing stages. The worse case is that the product maintenance problems may not appear until many units are sold.

The major problem that engineers face during product development is that a model of the product is not available for collecting customer, vendor, manufacturing team inputs. Manufacturing facilities are not available or not configured to assess the true capabilities of the manufacturing processes for the designed products. For instance, it is very difficult to decide only from CAD data whether a machinist would have difficulties to assemble the designed part, or to predict which difficulties a client may have when using the product.

For automotive industry, there are other problems. Henry Ford's success was based on the mass production of a single automobile model with same configurations including the color. Almost all the surviving car companies have followed this production model for quite some time. During last two decades, automobile industry has been drastically changed by the increasing competition and more demanding customers. Every car company tries to pride itself on its ability to give customers exactly what they demand.

Similarly, customers like to differentiate themselves from others by using products uniquely designed for their personal preferences. Providing such high level of customization requires complex system support, highly flexible manufacturing systems and continuously changing car body shapes requiring constant evaluations of concept designs.

Enabling flexible environment to respond to the changes and its ability to simulate processes and product designs interactively as they are in physical world, VR has became a promising complementary technology to overcome some of the difficulties faced in product development stage and increase their flexibility towards their constantly changing customer preferences. Today VR technology enables designers to:

1. demonstrate products in 3D space in real-time;
2. simulate manufacturing processes for new or existing products;
3. simulate assembly/disassembly processes;
4. train workers;
5. introduce products to potential customers for their feedback.

2.4 3D Sketching in Concept Design

For a concept design system to be truly useful in the product development process, it is necessary that the concept sketching is efficiently done and represents the details of the design well. Although majority of product development process is computerized, conceptual design has not been changed over the years. Pencil and paper based 2D sketching method is still the most common techniques for highlighting the configurations of the product in its initial design stage. Although input for the conceptual design comes from variety of different sources such as customers, engineers, company policies, market trend etc., a science conceptual design is still an art. Industrial designers give the basic shapes and functionalities of products by using their imaginations and talents on traditional pen and paper based platforms. Since in today's highly computerized manufacturing environments, having CAD images of the products is compulsory, the need for a postprocessor that automatically converts these 2D sketches into 3D CAD models is higher than ever. Today's CAD software does not provide friendly and intuitive sketching functionalities that enable industrial designers to transfer their artistic ideas into functional products.

Recent advancements in computer technology in both hardware and software levels made VR a strong candidate for simulating many real life phenomenon. Some research works and the needs, mentioned above, have created one more possible dimension to the application areas of VR which is the 3D free-form sketching (Bimber et al., 2000, Zeleznik et al., 1996). Today, a well designed VR simulator can provide a naturalistic environment for industrial designers to form the initial shapes of products in 3D space. Since the initial sketches of the products are directly placed in 3D space, outputs are in digital formats. The output of such systems has two important advantages: i) initial sketch can be visually observed from many different viewpoints like as it is a physical prototype; ii) initial sketches can be converted to the required CAD formats for further enhancements and analysis. Such processes would provide the flexibility and efficiency that companies urgently need for them to stay competitive in today's marketplace.

This thesis introduces an immersive VR based free form sketching toolbox that provides an intuitive environment for industrial designers in product design process. In this thesis, the possible ways of using data gloves to generate the sketches in 3D space are discussed. Most studies in this field use single data generating tools (similar to a pen) such as mouse, wand or specially created 3D pens (Yoshida, 2000). Although surface forming becomes challenging, use of data gloves gives two important advantages to the designer: i) better formed curves and surfaces shapes: position of each finger helps to form the curves and surfaces of object shapes; ii) a number of alternative ways to create sketches: A single finger creates a splines in 3D, side of a single finger can be used for smoothing an existing surface or creating new surfaces, finally the whole hand enables to create multi-curvature surfaces in 3D that is the natural way of working for many artists especially sculptors. Yet the main contribution of our work is to develop a two-step sketching technique. In the first step, rough sketches of the products being designed are generated using polygon meshes. In the second step, the initial sketches are used as a base to identify control points to generate smooth parametric surfaces that represent the initial sketch best. The outcome of the process is: i) intuitive and user friendly 3D free-form sketching tool; ii) accurate NURBS representation of the surfaces.

2.5 Literature Review

The need for digital free-form sketching methods was realized long before enabling technology for immersive simulation was available. Several theoretical methodologies were suggested during late 80s. Many more practical solutions in non-immersive environments were suggested at early 90s. One of the early practical attempts to sketch in

3D space was the work by Sachs et al. (1991) where the user interface was captured using two 6-DOF electromagnetic trackers in a non-immersive virtual environment. In this work, products are generated from the set of independent curves drawn in the 3D space using a 3D virtual pen. Although the suggested method was a successful attempt to present designed products in wire-frame formats, the work was not able to suggest a solution to generate surfaces of the designed objects. Baudel (1994) presented an approach that allows designers to present their ideas by hand motions in computer. Although the objective of this study is to improve the efficiency of the design process, the main idea of this work is limited on hand gesture recognition. Moreover, this method does not suggest any method to convert sketches to CAD formats which is one of the main problems of design efforts.

Since mid 90s, VR involvement in conceptual design has increased drastically. Due to the advancing technology and availability of more advanced VR hardware and software such as trackers, shutter glasses, HMD, 3D mousses/wands, haptic-devices and data gloves, some studies have focused on development of partially and fully immersive VR platforms to be used for free-form sketching. One of the early works uses VR as a sketching environment is the HoloSketch (Deering, 1995). HoloSketch allows users to work in 3D space using head trackers, stereo glasses and regular computer monitors. Images are generated in the 3D scene using 3D wands. Altough HoloSketch enables several options to the designers to transfer his/her thoughts to an end-product geometry, drawing images on an empty 3D space in computer generated environment neither practical nor intuitive for designers. Designers have to spend most of their time focusing on connectivity of different geometry segments. Furthermore, Deering's model does not discuss the possibility of generating CAD format of the final sketches.

Another desktop based system called "Conceptual Virtual Design System" suggested by Dani et al. (1999) to improve the indispensable efficiency of conceptual design. System integrates hand gestures recognition as input, voice input, and keyboard to create and manipulate a 3D artifact. Objects are generated by specifying shapes and dimensions of geometries. Igarashi et al. (1999) developed an interface for free-form modeling in which users first sketch geometric shapes onto a display, then the system create a matching shape for the contour. Once these shapes are generated, user can make some modifications on them.

Cutler et al. (1997) studied the interaction of two hands for the Responsive Workbench (RWB) based on how hands work together in daily life situations. Tolba et al. (2001) described a system that lets users to draw a scene with 2D strokes. The 2D images later are viewed from several different locations as if a new 3D scene is created by them. This is done by projecting the 2D strokes on the sphere with the center at the eye point and then viewing them in perspective. Zheng et al. (1998) attempted to form a curve by locally matching with other curves. Wesche et al. (2000) proposed a spline-based sketching tool for the conceptual drawing of curves and surfaces in the RWB. In this approach, users can draw images in 3D space using 3D stylus. In a more recent work, Wesche et al. (2001) released the improved module of this work in the same environment by implementing additional tools to support indirect drawing and modification of curves for product design. In our experiment with various configurations, we found that without a reference point, it is extremely difficult for designers to generate appropriate shapes of their products using splines.

There are several similar works can be found in the literature. Most of these works are similar in nature yet differentiate from each other by the technology selection. For example, Chu et al. (1997) have developed a VR based CAD system that uses a table-like back projection system, also known as Virtual Table (VT). Use of shutter glasses provides a large stereo environment. Furthermore, Schmalstieg et al. (1999) have developed a whole new user-interface design space for VT using a transparent pen and a plexiglas sheet. Subsequently Encarnacao et al. (1999) exploited this approach for sketching solid geometries through gesture input. Advanced Realism Computer Aided Design Environment (ARCADE) (Stork, 2000) is another intuitive system for modeling surfaces on VT. It offers variety of functions for surface creation. The VR environment was constructed from a pad and a pen, both equipped with 6 DOF magnetic trackers, as input devices, and a VT as stereo display device. The functionalities are accessible via the 3D main menu displayed on the pad. The major strength of this approach for sketching is that users can perform Boolean operations, access the history of an object to change it partially or copy primitives. The object manipulation functionality is performed in a gesture-based manner. ARCADE also enables its users to perform picking and manipulating operations.

The level of immersion has a significant effect on the naturalistic side of the whole sketching process. Some works aim at collecting reference points from designer's inputs, defining curves by parametric equations, generating smooth surfaces and connecting different segments of these surfaces to each other to create the complete shapes of the objects being designed. In their semi-immersive model, Bruno et al. (2003) uses a pen to generate reference points in the 3D space. The generated surfaces are NURBS representations of the collected reference points. Some of the well-known concepts in CAD such as extrusion and revolving are used in generating the surfaces. A 3D eraser pen is also introduced in this work to correct the curves quickly during the sketching.

An important approach for surface sketching is "Surface Drawing" (Schkolne, 1999). Since shapes are forged by the motions of freely moving hands, "Surface Drawing" provides creative expression and perceptual thinking of the user to the design. In this system, polygon surfaces are created by moving a hand, instrumented with a special glove, through 3D space in a semi-immersive and interaction VR environment. Users can explore the design space freely during the modeling process without the need to plane the construction of the final shape. It also allows the user to freely expand, join, and erase surfaces based on the hand motions with tangible tools. "Surface Drawing" is one of the rare works that allows users to generate surfaces directly in 3D space using hands. Although surface generation in this approach is intuitive, there is no solution is suggested for generating smooth surface results and appropriate CAD format of images.

Most of these works, except Schkolne, aim at generating parametric representation of the curves and surfaces while the designer works on the sketching. Generation of these representations during the sketching forces the designer to focus on the requirements of the parametric curves and surfaces generation such as the connectivity of different curve and surface segments. Focusing on the mathematical details rather than the product design itself limits and wastes the designer's artistic capabilities. On the other hand, parametric representations are the best choices to represent smooth and accurate surface generation. They are also more efficient in storing information and performing modification on existing designs. Hence for a VR based sketching tool to be practically acceptable, it should be able to representing final object shapes in parametric forms.

The work described in this thesis is based on the earlier works that suggest generation of surfaces using hand gestures. Shaping objects using hand is most natural way that most sculptors work on clay models. Today most clay mockups of products are generated in this way. On the other hand, sketches of conceptual designs are done using pen and paper and later their detailed representations are created in computer using CAD software. A digitalized sketching method enables formation of 3D shapes of objects directly in computer can minimize the need for inefficient pen and paper based sketching techniques. "Surface Drawing" system partially realized this idea. It only uses polygon surfaces but not parametric surfaces. Polygon surface consists of small triangle planes and is good at creating arbitrary shapes for objects. Using polygon surface for generating objects does not need to worry about mathematical problems like in NURBS. However, object shapes cannot be represented at expected degree of smoothness and precision.

Furthermore, saving the data of polygon surfaces takes more space and modifying is also a headache problem.

Chapter 3

Free-Form Sketching System

In this chapter, we outlined our solution to sketching problems in product development process. In our solution, we focus on designing a sketching tool that can capture the designers' ideas rapidly and freely in 3D space without sacrificing from the intuitive part of the design process. The reason that pencil and paper based sketching is still being used for concept design is its capability of rapidly capturing designers' inspirations. Designers can express their ideas by rapidly sketching primary lines on paper using a pen. These lines do not show the full details of the product but give the distinctive characteristics of the product. These characteristics are the ones that distinguish the product from others. Although a skilled concept designer can finish a drawing in very short time, in order for one idea to be considered for detailed design, several rounds of sketches need to be generated and reviewed until a product design process is completed. Another requirement for conceptual design is that the design process should be intuitive. It means that the available design tools should enable designers to express their inspirations freely without being constrained. Furthermore, the final results should give the realistic impression of the product being designed and people who will evaluate the design should not be forced to imagine the 3D shapes of a product from 2D sketches.

VR technology enables us to develop a system that satisfies these speed and intuitiveness requirements. We present a method that allows designers to rapidly express their inspirations in 3D space directly in an intuitive way. In our method, product design is completed in two consecutive steps. First, we use pieces of small triangle patches attached to the virtual hands to construct the rough sketches of the designed product (Figure 3.1 illustrates the rough sketching creation process). There are two advantages of this process. Firstly, constructing surfaces using patches of triangles does not require designers to know complex mathematical foundations like in the development of parametric curves and surfaces. Hence, designers are not bothered with the properties of geometries. Instead, they can focus on the design itself. This enables them to complete the initial sketching phase in a very short time. Secondly, designers are not forced to focus on the curvature continuity of different segments connections that are the most time-consuming parts of CAD designs. Since the objective of the designer in this stage is to capture the overall shape and features of the product, details should be ignored. Hand gestures are captured using data-gloves, position trackers, and mouse and keyboard enters.

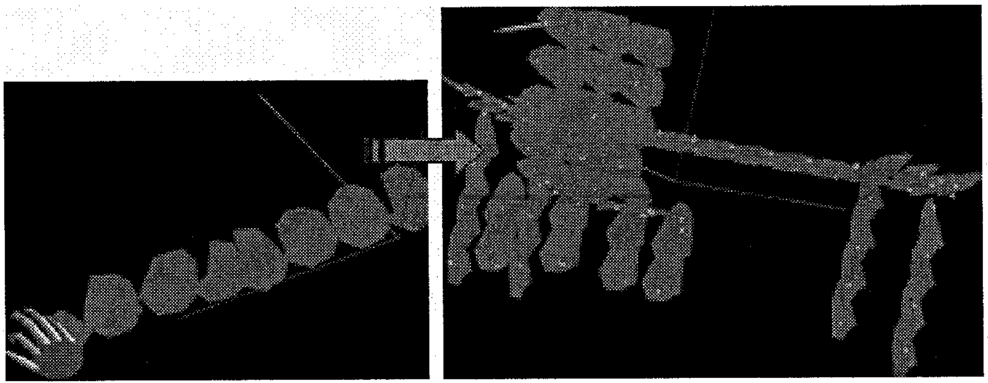

Figure 3.1: Surface generation by using Data Gloves.

This process is intuitive because designers use their hands to create the desired shape of the product directly in 3D space. Obviously, the rough sketches of the product which are composed of thousands of small triangles are not delicate enough to show the complete details of the product. As such in the paper sketching, it only shows the distinct characteristics of the product. When the designer satisfies the primary figures, she/he can generate the final, refined shape of the design using parametric surfaces in the second step (Figure 3.2 illustrates the generation of parametric surfaces from the initial sketch). In this phase of our approach, designer selects sufficient number of points using a virtual pin from the surface of the sketch created in the first step. Once sufficient number of points is selected from the sketched surfaces, smooth NURBS surfaces that interpolate these selected points are built under a series of mathematical computations. In addition to these smooth surface representations, NURBS is also appropriate choice for surface modifications. In this step, designers experiment several different options until the NURBS surfaces exactly representing the initial sketch which is the true inspiration of the designers. Finally, the overall characteristics of the designed product are presented to various groups for their inputs in 3D simulations. Since in this approach, the complete design process is conducted in computer, the proposed design process is extremely flexible in responding to changes or modifications suggested by the groups whom involve in the design process. Furthermore, alternatives of the product can easily be generated with minimal effort. This thesis presents a VR based concept design system that is fast, intuitive, 3D and practical for artists or industrial designers to work in concept design. In the next section, details of the free-form sketching tool are summarized.

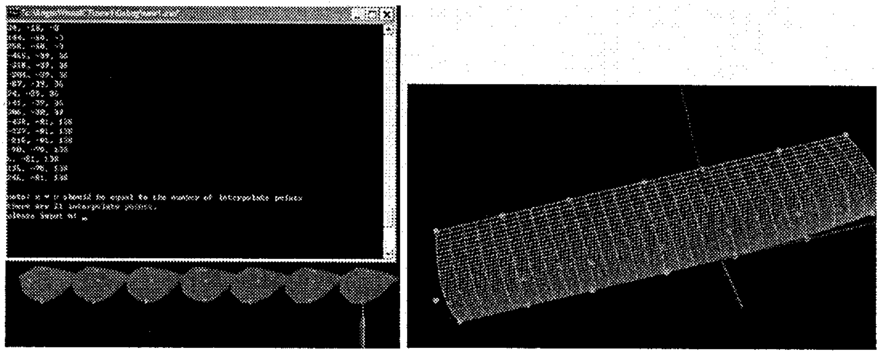

Figure 3.2: Selection of control points and parametric surface generation.

3.1 Polygon Mesh vs. Parametric Surface

There are two ways of generating 3D surfaces in computer graphics: i) polygon meshes and ii) parametric surfaces. Every method has its advantages and disadvantages. The described free-form sketching technique in this thesis uses both polygon meshes and parametric surfaces to generate representations of 3D images by maximally utilizing their advantages and minimizing their disadvantages. Combining polygon meshes and parametric surfaces systematically in the VR based toolbox enables us to realize the whole product development process from the conceptual design to the marketable product in computer.

A polygon mesh is the collection of vertices in 3D space that are connected to each other in the way that several independent but non-intersecting planes are constructed. Although they are more appropriate to generate objects with rectangular geometries, polygon meshes are also used to represent curved surfaces. Triangles are the most frequently used forms to generate meshes yet other geometries such as rectangles are also possible to generate polygon meshes. The advantages of using polygon mesh are:

- easy to generate;
- no mathematical knowledge is required;
- can be converted to any desired CAD format.

Although there are several advantages of using polygon meshes in design, there are some drawbacks of working with them as well. Some of the difficulties we encounter are:

- shapes can only be approximated, Exact shapes cannot be captured when curved surfaces exists;
- fitness to the desired shapes can only be possible with generation of significantly higher number of polygon meshes;
- increased number of polygons consume significant amount of computer memory and CPU time;
- rendering and animation of images becomes computationally expensive;
- engineering analysis such as finite-element analysis is difficult if not impossible;
- modification of the exiting geometries is difficult.

Hence, we can conclude that although polygon meshes are desirable for creating 3D images, they are not accurate enough to represent and store them.

Contrarily parametric equations can define the exact location of a point on the object surface. By changing control parameters (two for the surface generation), one can define the exact coordinates of any point belonging to the surface of an object being designed. Thus, object shapes can be represented by parametric equations by one hundred percent accuracy. By using parametric equations, almost all the necessary curves and surfaces belonging to an object can be mathematically defined using a small number of control points. Other advantages of representing objects using parametric equations are:

- less storage space is required;
- modification of surfaces is easy (Illustrated later in Figure 3.4);
- conversion to desired CAD format is easy;
- computationally less expensive for animation;
- several engineering analysis is possible (ex. finite-element analysis);
- collision detection is possible and easier comparison to polygon meshes;
- level-of-details can be controlled easily: rendering quality can be changed depends on the desired level.

Although we can list several advantages of parametric surfaces for representing 3D object designs, it is not perfect. The disadvantages of parametric surfaces are:

- designers should have a good understanding of the mathematical fundamentals of the parametric equations: the arrangement of control points in u and v directions, the connection of two surface segments, the boundary conditions etc. should be dealt appropriately during the product design process;
- they are not user friendly for creating 3D shapes.

Thus, we can conclude that although parametric surfaces are good at representing 3D images, they are not the perfect choice to generate brand new computer images.

3.1.1 Polygon Mesh

Most systems using polygon meshes to generate surfaces require coordinates of several points and the connectivity information of them. Depending on the desired quality of the representation, other properties such as surface normal and textures can be defined as well. On the other hand, when complex objects are designed, polygons are not the best option. In some cases, millions of polygons may be necessary to represent an object. In order to increase the realism of the design, normal vectors have to be computed for each vertex. Moreover, modifying the shape of an existing geometry in a simulation is almost impossible. At the early stages of the design, sketches are subject to several modifications based on the inputs received from customers, vendors, engineers, manufacturing group etc. This requires almost exclusively a brand new sketch of the design each time. If such sketches are only available in polygon mesh format in computer, modification of the shape is extremely difficult. Adding a new vertex or stretching the existing one belonging to a triangle or quadrilateral would only reshape polygons where share this vertex (see Figure 3.3 for illustration). Because one point does not have any relation to any other one, modifying one vertex does not change other points. Note that a surface may consist of millions of polygons and this makes the reshaping very troublesome. Finally, the inherent and vital shortcoming for polygon mesh is that they only approximate the true surfaces. Actually they are not surface. The word, smooth, is not at all suitable for polygons. Essentially, triangles are mathematically planes. Just because they are so small, they look like surfaces to naked eyes.

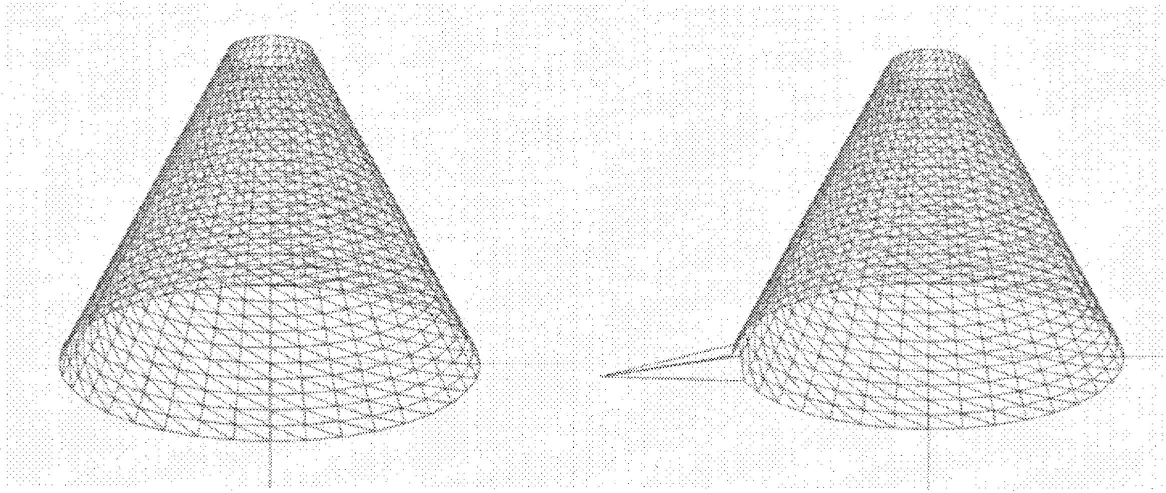

Figure 3.3: Changing the location of one vertex on a polygon mesh object only changes object shape locally.

3.1.2 Parametric Surface

Use of parametric surfaces in computer graphic is relatively new. By using parametric surface, any surface can be described mathematically by a small number of control points. Comparison to the polygon meshes, parametric surfaces requires considerably less storage spaces. Moreover, surfaces can be rendered at any precision. To improve the quality of the visualization, normal vectors can be calculated for newly created surfaces. Objects that are in close proximity to the observer and needed to be studied in details can be displayed in higher precision whereas those objects away from the observers or less importance to the objective of the simulation can be viewed in less detail. The control of detail level in VR simulations is quite important and parametric surface representation method naturally fits to the flexible detail representation concept in VR.

Parametric surface definition also enables designers to modify object shapes with minimum effort. Working with a small number of control points comparison to that of the vertices in polygon meshes makes the reshaping more feasible. Changing a single control point causes appropriate changes on the surface (see Figure 3.4 for illustration). Since each surface patch is generated by a small number of control points, designers can easily locate the appropriate control point where its modification results with the desired surface shape. Although parametric surfaces gives more precise surface definition, and several flexibilities for designing, modifying and displaying virtual object, working with them requires good knowledge of its mathematical foundations.

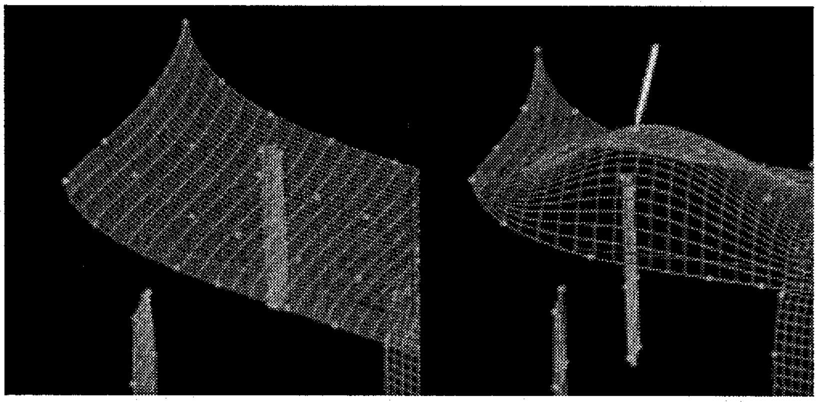

Figure 3.4: Global effect of local modifications in parametric modeling.

3.1.3 Polygon Mesh and Parametric Surface Enable Free-Form Sketching in VR

Based on the above discussion, now we introduce our sketching technique that takes advantages of strength of both polygon meshes and parametric surfaces and desert their disadvantages. In our method, approximate shapes of the objects are first created using polygon meshes. Next, polygon meshes are approximated by appropriate parametric surfaces. Finally, these parametric surfaces are modified until the most appropriate shapes of the designed objects are created.

The described sketching system is developed using C++ programming language and OpenGL graphic library. The components of the developed model are given in Figure 3.5.

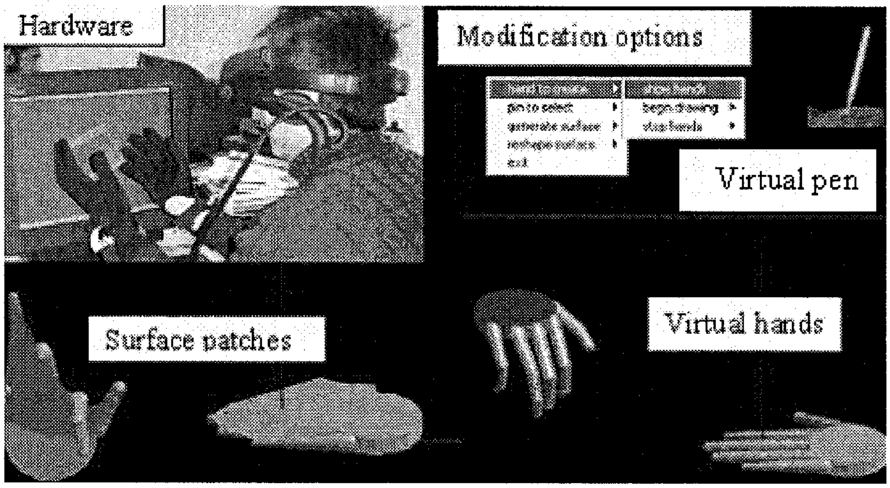

Figure 3.5: Overall architecture of the VR based sketching system.

In order to capture the initial shape of the designed object, data gloves are used. Two virtual hands are modeled to simulate the designer's hands in the virtual world. By using 27 connected triangular planes, the interior surfaces of the virtual hand are approximated. During the sketching process, these polygon meshes that replicate the inside of human hands are used to generate desired object shapes in 3D world. A simple distance calculation algorithm is developed to avoid high overlapping of these polygon mesh patches. If the distance between existing patches and current location of the virtual hand is satisfactory, a new set of polygon meshes is placed in 3D space. Using both left and right hands freely, rough image of the desired object shape is constructed in 3D world. The generation of these rough sketches of the desired object shape is quite fast. Other than distance calculation between exiting patches and the current location of the virtual hand, no other computation is required during the whole sketching process. Since the process is fast, designer's inspiration can be caught accurately. Capturing the designer's inspiration is the key to successful concept design. Inspiration of an artist or an industrial designer is the one that greatly makes the prospective product stands out from other comparable ones.

Although it is fast and intuitive, the surface patches made of small triangular planes are not geometrically complete. Other than visualization, not much can be done using these sketches. Yet, triangular plane patches create a perfect base for generating smooth and mathematically accurate parametric surfaces. As a result, once the rough sketching is completed, sufficient number of control points on the plane of the existing sketches is selected using a virtual pen as shown in Figure 3.6

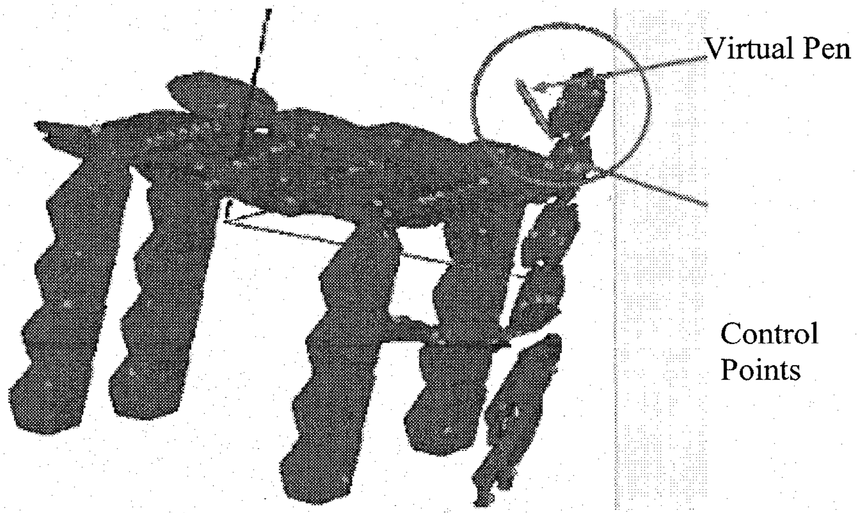

Figure 3.6: Selection of control points on a sketched surface.

Once sufficient number of points is selected on the planes of the sketches, parametric representations of the object are created. Details of the parametric surface generation are discussed later in the thesis. Naturally, generated parametric surfaces may not satisfy the expectations of the designers. In the next step of the described sketching system, designers are enabled to modify these parametric surfaces until they are perfected to satisfy designers. In this phase, the virtual pen is used to select and drag these points. Changing the location of these points modifies the shape of the object being designed. The initial rough sketch will be used as a benchmark during this phase since it is the true inspiration of the designer. Hence, while the modifications are made, designers will have chance to compare the parametric surfaces with the sketch of the product in 3D space. In 3D world, selecting and dragging points and observing their effects on the surface being designed are simple. Moreover the design process is friendly and extremely efficient. Our implementation also offers the designer to edit the data of the interpolating points directly in the saved data files.

The summary of the steps described in this thesis in order to create a successful product is given below:

1. create a rough shape of the designed product using designer's hands;
2. select sufficient number of points using a virtual pen;
3. generate NURBS surfaces interpolating these selected points;
4. modify parametric surfaces by:
    - moving interpolating points using a virtual pen;
    - editing the data of these interpolating points directly in the saved file.

3.2 Components of the Free-Form Sketching System

In this section we will introduce several physical and virtual components that we used to accomplish the goals described earlier in this chapter.

3.2.1 Virtual Hand and Surface Patch

The virtual hand which is an essential part of our system is constructed using OpenGL graphic library.

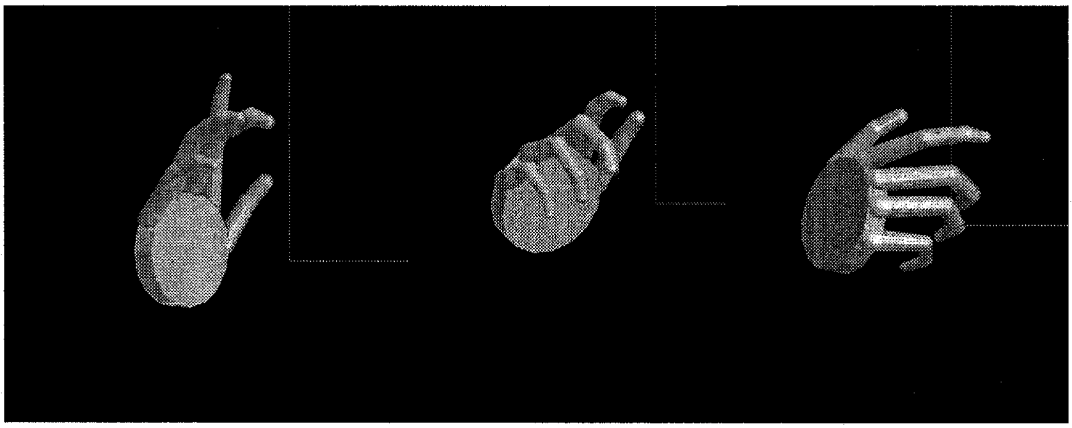

Figure 3.7: Virtual hands built with disks, cylinders, and spheres.

The virtual hand shown in Figure 3.7 is easy to build and has several advantages for our implementation. First, it simulates a human hand reasonably well. Second, the designed hand is able to replicate the actual hand with the data received from the data gloves. Finally, using simple geometries to create the shape of the hand increase the efficiency of the simulation. In any given VR simulation, the computer has to recalculate all the activities occurring at the given time frame including rotation and translation of all objects in the scene, rendering of geometries. Depending on the complexity of the model, in most cases physical phenomenon such as collision between virtual objects needs to be ensured during the simulation in each frame. On the other hand, 20 frames per second is the minimum tolerable frame update rate to achieve continuous movement effect in the VR simulations. Less than 20 frames per second causes visible flips on the display. Therefore, a simple hand design is an appropriate choice to replicate actual hand motions precisely in VR simulation without sacrificing from the frame update rate.

In OpenGL, object manipulations such as rotations and translations can all be represented by applying an appropriate 4×4 transformation matrix to the coordinates of object vertices. Let's denote v to represent a homogeneous vertex and M is a 4×4 transformation matrix, then Mv is the image of v under the transformation by M.

$$\begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} = [-yaw][-pitch][-roll][-translate] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where:

$$T_{11} = \cos(yaw) \times \cos(roll) - \sin(yaw) \times \sin(pitch) \times \sin(roll)$$
$$T_{12} = \cos(yaw) \times \sin(roll) + \sin(yaw) \times \sin(pitch) \times \cos(roll)$$
$$T_{13} = -\sin(yaw) \times \cos(pitch)$$
$$T_{14} = -TranslationX \times T_{11} - TranslationY \times T_{12} - TranslationZ \times T_{13}$$

$$T_{21} = -\cos(pitch) \times \sin(roll)$$
$$T_{22} = \cos(pitch) \times \cos(roll)$$
$$T_{23} = \sin(pitch)$$
$$T_{24} = -TranslationX \times T_{21} - TranslationY \times T_{22} - TranslationZ \times T_{23}$$

$$T_{31} = \sin(yaw) \times \cos(roll) + \cos(yaw) \times \sin(pitch) \times \sin(roll)$$
$$T_{32} = \sin(yaw) \times \sin(roll) - \cos(yaw) \times \sin(pitch) \times \cos(roll)$$
$$T_{33} = \cos(yaw) \times \cos(pitch)$$
$$T_{34} = -TranslationX \times T_{31} - TranslationY \times T_{32} - TranslationZ \times T_{33}$$

$$T_{41} = 0$$
$$T_{42} = 0$$
$$T_{43} = 0$$
$$T_{44} = 1$$

For a separate operation of translation or rotation:

translation is given by $T = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and its inverse $T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -z \\ 0 & 0 & 0 & 1 \end{bmatrix}$;

rotation about x, y and z are:

$$Rx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Ry = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Rz = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

respectively where α is the rotation angle about the given axis.

Graphic library OpenGL provides efficient functions to perform various animations in the simulation:

- glTranglate(TranslateX, TranslateY, TranslateZ) for translation;
- glRotate(RotateAngle, AxisX, AxisY, AxisZ) for rotation;

With these build-in functions, complex objects and scenes can be developed by applying following steps:

1. glPushMatrix() to save the current position;
2. glTranglate() and glRotate() to reach a desired position and orientation;
3. render geometric solids based on this current position;
4. repeat 2 and 3 to build the other parts of the object or the scene;
5. once rendering of all parts belonging to an object or scene is completed, using glPopMatrix() to update the current location of images;
6. repeat 1 to 5 to construct another scene or object groups.

The pair of virtual hands we use in our model are generated using these steps. Left and right hands are two independent object groups. The main reason we designed the virtual hands is to simulate the actual movements of designer's hands with the data captured through data gloves. A human hand consists of 16 highly flexible joints. Hence the pair of virtual hands is designed in such a way that relative positions of each joint on an actual hand in any given position are replicated in the VR simulation. Joints belonging to a hand are assembled together on the base of child-parent relationship. Motion of parent is followed by the children, yet each child has flexibility to perform its own transformation according to the pre-determined conditions (child can not desert parents during the simulation as such a joint of a finger cannot leave the hand/finger). Thus, joints are separate convex objects and attached to each other using several independent child coordinate systems according to forward kinematics principles in order to replicate an actual hand in real-time interactive simulations. The geometrical relationships between joints are demonstrated in the Figure 3.8.

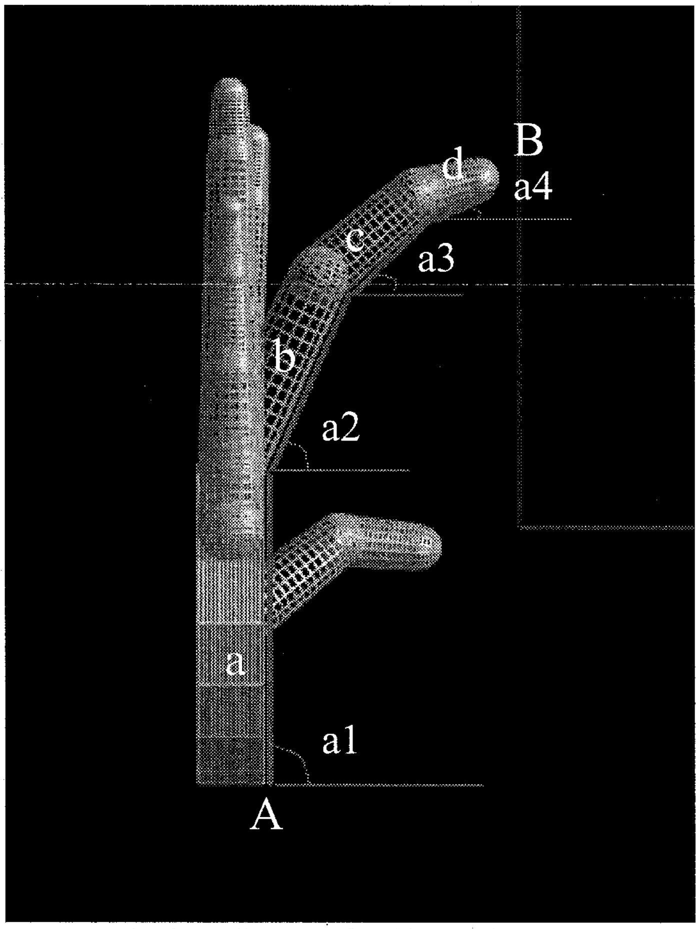

Figure 3.8: Coordinates of point B related to point A.

Where the coordinates of point B (x_B and y_B) can be found as:

$$x_B = a \times \cos\alpha_1 + b \times \cos\alpha_2 + c \times \cos\alpha_3 + d \times \cos\alpha_4 + x_A$$
$$y_B = a \times \sin\alpha_1 + b \times \sin\alpha_2 + c \times \sin\alpha_3 + d \times \sin\alpha_4 + y_A$$

In the free-form sketching system described in this thesis, total 20 vertices are used to generate 27 triangular planes as shown in Figure 3.9. These flexible triangles cover the area an actual hand follows.

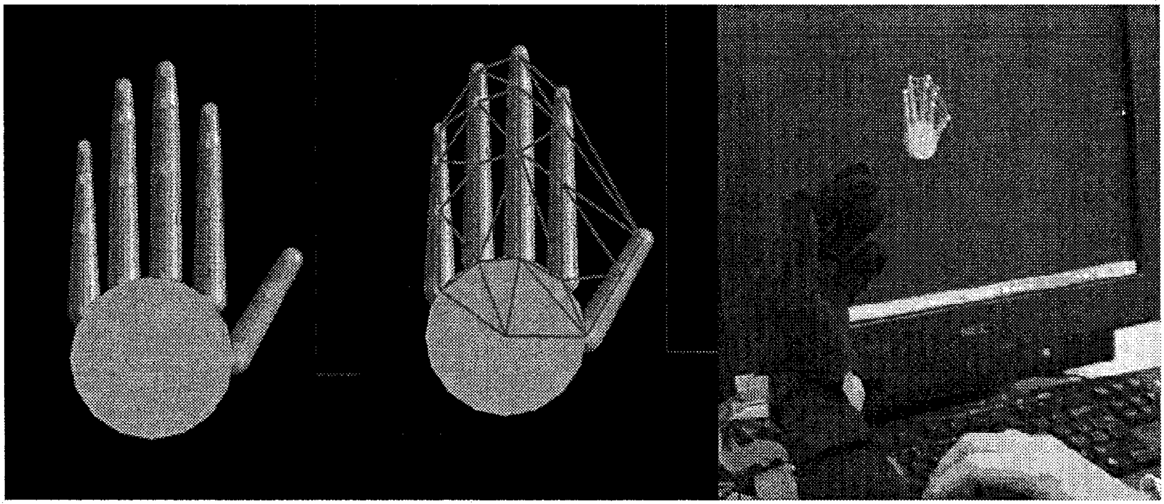

Figure 3.9: Virtual hand with joints, triangulation of surface patch and the animation of actual hand in VR.

5DT data-gloves (shown in figure 3.9) send 7 signals to track the hand gestures. The five of them are the flexures for the five fingers, and the other two are the pitch and roll of the palm. Five integer values of the curvature for each finger are in the range of $0 \sim 4096$. Because the range of the flexure of each joint is $-110^\circ \sim 20^\circ$ (unwind finger as the original position, bend is a negative degree and elevation is a positive degree), so we set a corresponding relation between the received data-glove signals and flexure range $(-110^\circ \sim 20^\circ)$. Table 3.1 displays the corresponding relationship between these relationships.

Table 3.1 Corresponding relations between the input data and output angles.

Input	Output
0 ~ 4096	flexure degree for finger joints −110° ~ 20°
0 ~ 31	−110°
32 ~ 62	−109°
63 ~ 93	−108°
……	
3380 ~ 3410	−1°
3411 ~ 3441	0°
3442 ~ 3472	1°
……	
4000 ~ 4030	19°
4031 ~ 4061	20°
4062 ~ 4096	20°

In this way, virtual hands can animate user's hands. Triangulation of virtual hands as described earlier enables us to define surface patches of the product being designed as shown in Figure 3.10.

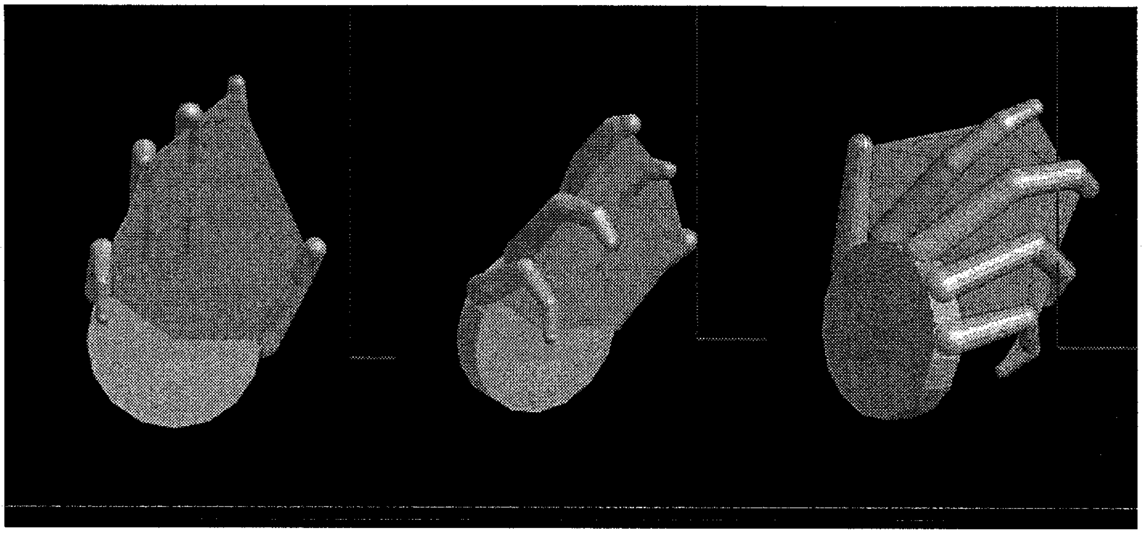

Figure 3.10: 27 triangle planes attach tightly to each hand.

3.2.2 Control Point Selecting

Once the rough sketches of the design are created, a sufficient set of control points are located for use in parametric surface generation in the next step. The virtual pen controlled by the mouse and key-board inputs is used to define points on the sketched surface. Figure 3.11 illustrates the point selection process.

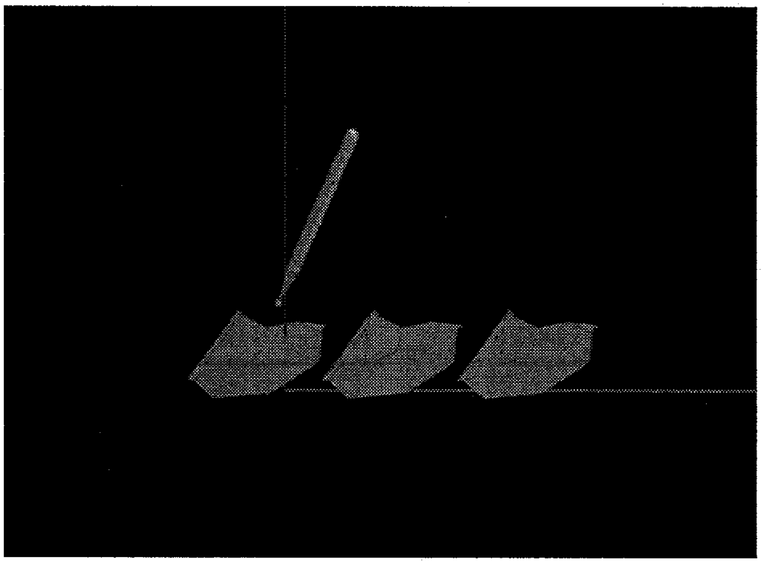

Figure 3.11: Control point selection process using virtual pen.

Generated surface patches may be located in 3D space in any rotation and translation. Moreover, depending on the hand gesture at the time a patch is generated, the shape of the patch includes several convex and concave curvatures. Hence, a mathematical model is required to identify the exact location of the virtual pen and the surface patch intersection point.

- First, the location of the end point of the virtual pen is calculated as:

  $$v' = Mv$$
  $$M = [R : T]$$

  where R is the rotation matrix, T is the translation matrix from the origin, v is the initial coordinates of the end-point and v' is the current location;
- Next, the triangular surfaces and the current location of the virtual pen are projected to one of the 2D planes (XY, XZ or YZ) as shown in Figure 3.12;
- If the projection of the virtual pen is inside of any one of these 2D planes, there is a possibility that virtual pen and the triangular surface are intersecting;
- Finally, by using plane equation of the surface, we check the distance between virtual pen and the surface. If the distance is zero, the virtual pen has contact with the surface. In practice, instead of zero distance, a small tolerance value (0.0001) is used.

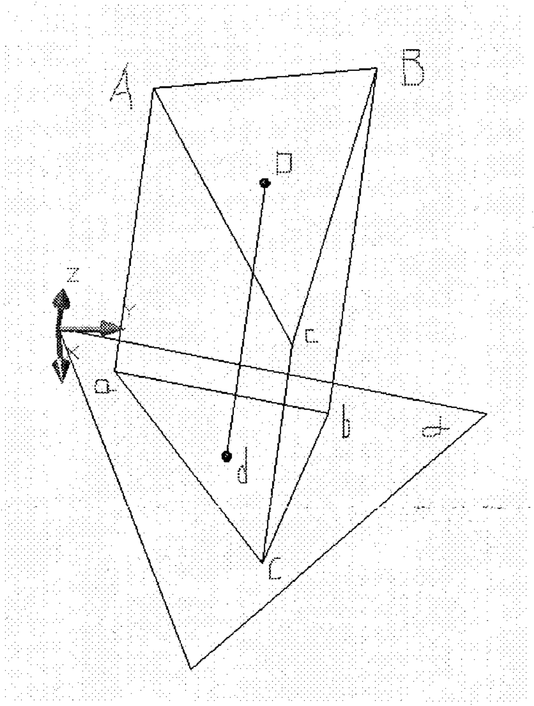

Figure 3.12: Projection of the triangular surface ABC and the virtual pen D on XY plane.

Following functions are called systematically to perform intersection test.

- "Same_Side()": Identifies if two points are on the same side of a line.
- "Point_In_Triangle()": Calls function "Same_Side()" three times. If all these calls return true, it performs distance calculation between virtual pen and the surface.

Details of these functions used in the free-form sketching system are given below.

```
Same_Side (p1, p2, a, b)
{
    Computes whether points p1, p2 are on the same side of line defined by $ab$.
    Returns: $\begin{cases} true & \text{points are on the same side of the line } ab \\ false & \text{otherwise} \end{cases}$
}
```

Point_In_Triangle (p, a, b, c)

```
{
    Compute whether a point $p$ in 3D is inside of a triangular surface $ABC$.
    Returns: $\begin{cases} \textit{true} & \text{point is inside of } ABC \\ \textit{false} & \text{otherwise} \end{cases}$
}
```

This is a very fast way because if some condition is not satisfied, the checking will be terminated. One may argue that, when the virtual pen's translational speed is fast, it may penetrate through a surface rather than touching it as shown in Figure 3.13. Eventually the described method would fail to detect a contact and its location when such case occurs. Although we did not implemented in our system, such problems could be solved by checking collision between a surface and line segment constructed by the location of virtual pen in two consecutive frames (t_1 and t_2). Solution methodology for such a problem is mathematical straightforward yet computationally more expensive.

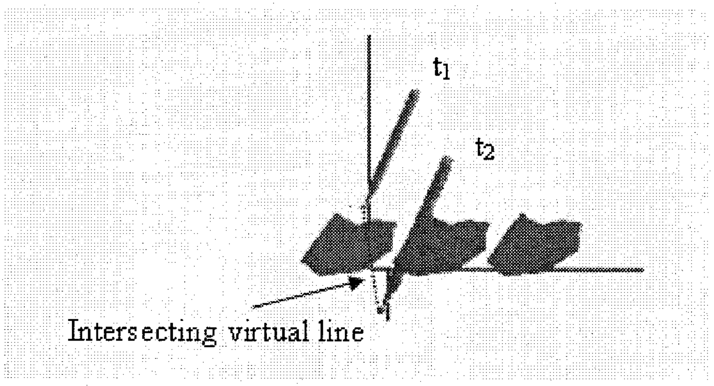

Figure 3.13: Intersection between virtual pen and surface under fast translation.

3.3 Parametric Representation of Design Surface

There are many kinds of parametric equations to be used for geometric representation of computer images. Among them, NURBS is considered to be the most complete option to represent surfaces of computer images. It is a unique technique which combines characteristics of other parametric representation techniques such as Bezier or B-Splines curves and surfaces. Almost exclusively all the CAD/CAM systems use NURBS to store geometric entities (Zeid, 2005). It has become the de facto industry standard for the representation, design, and data exchange of geometric information processed by computers. NURBS allows conversion from one system to another through well structured exchange standards such as STEP and IGES.

NURBS can create continuous, smooth curves and surfaces from control points. These points do not have to be in regular intervals. Possibility of using non-uniform knot-vector gives the freedom to create regular or irregular continuous curves and surfaces. Connecting the control points of NURBS curves and surfaces gives a convex hull which strongly contains the shapes of the curves or surfaces which are an approximation of the actual shapes. Furthermore, NURBS representations of curves or surfaces allow developers to utilize both control point movement and weight modification to attain shape control which is not possible with other parametric equations. For example, it is difficult to edit a B-spline curve or surface by modifying control points. In order to create a noticeable change on the shape of B-spline curve or surface, control points must be stretched significantly.

Contrarily local modification of curves or surfaces is quite flexible in NURBS format. If the control points $\{P_i\}$ is moved to a new location or the weight $\{w_i\}$ is modified, it affects only that portion of the curve on the interval $u \in [u_i, u_{i+p+1})$ (For surface, if $\{P_{i,j}\}$ is moved, or $\{w_{i,j}\}$ is changed, it affects the surface shape only in the rectangle defined by the control parameters u and v --$[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1}))$. Changing the weights of control points affects their gravitational impact on the curve and the surface. Qualitatively, increasing the weight pulls the curve or the surface toward the control point, decreasing weight pushes away the curve or surface from the control point. Thus, a curve or surface can be edited without moving control points significantly. Between moving control points and adjusting their weight, NURBS provide a much more flexible tool than regular B-spline. NURBS provides a unified mathematical basis for representing both analytic shapes, such as conic sections and quadric surfaces, as well as free-form entities, such as car bodies and ship hulls. Designing with NURBS is intuitive because almost every tool and algorithm has an easy-to-understand geometric interpretation. Designers do not have to spend their times and energies to study the mathematical properties of NURBS in order to work with them. For application in computer, the algorithm of NURBS is fast and numerically stable. These advantages ensure the application of NURBS in large and complex 3D scene such as representations of auto-body, airplane engine etc. designs which requires significant computation power for rendering, animation and various engineering simulations. An important property of NURBS for interactive design, which is common to most parametric representations types, is that they are invariant under common geometric transformations, such as translation, rotation, parallel and perspective projections because these transformations are applied to the control points only.

In our program, we focus on building NURBS surface that interpolates some specified points selected from the previous rough sketches of the design. In the previous step, selection of these control points was discussed in details. A series of points are selected by using the virtual pen. The goal is now to generate NURBS surface that interpolates these selected points. In following sections, the developed algorithm for generation of NURBS surfaces is discussed in details.

3.3.1 Computing One Point on NURBS Surface

First, we need to computer one point on NURBS surface. Hence, the algorithm allows us to generate as many points as we desired on the surface of the design and we will get this surface by connecting these points. The function "Point_On_Surface" is designed to generate such points.

```
Point_On_Surface (m, p, U, n, q, V, P, u, v, S)
{
   Compute one point on NURBS surface.
   The output is S which is a point on the surface.
}
```

The m and n are the numbers of the control points in u and v directions. This does not mean that these control points must be arranged in rectangular form in 3D space. These points can be anywhere. There is no limitation on the order and position of these points. Symbols p and q are the degrees in u and v directions. In our application, we set them to 3 because using a cubic degree of freedom will likely yield to a surface with much fewer control points than a quadratic function using C^1 continuity. U and V are the knot vectors of u and v respectively. P is the set of the control points. When u and v changing from 0 to 1 under a given augment for each one, desired number of points are computed. Connecting these points yields to a surface. The smaller the augment is set to, the smoother the surface is.

In function "Point_On_Surface", we use two other important functions "Find_Span" and "Basis_Function". They are for computing the basic functions $\{N_{i,p}(u)\}$ of NURBS surface.

```
Find_Span (m, p, u, U, I)
{
    Determine the knot span index
    The output is I which is the knot span index.
}

Basis_Function (i, u, p, U, N)
{
    Compute the non-vanishing basis functions.
    The output is N which is the non-vanishing basis functions.
}
```

$U = \{u_0,..., u_m\}$ is a nondecreasing sequence of real numbers, i.e., $u_i \le u_{i+1}$, $i = 0,..., m-1$. The u_i is called knot, and U is the knot vector. The i^{th} B-spline basis function of p-degree (order $p+1$), denoted by $N_{i,p}(u)$, is defined as

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u \le u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

For NURBS curves and surfaces, in any given knot span, $[u_i, u_{i+1})$, at most $p+1$ of the $N_{i,p}$ are nonzero, namely the functions $N_{i-p,p},..., N_{i,p}$. All other functions are identically zero and it is wasteful to actually compute them.

Hence, the first step is to determine the knot span in which *u* lies. Function "Find_Span" finishes this task by using a binary search. Assuming *u* is in the i^{th} span, we can compute the nonzero basis functions as illustrated below. Function "Basis_Functions" computes all the non-vanishing basis functions and stores them in the array *N*.

$$\{N_{i,0}\} \Rightarrow \begin{Bmatrix} N_{i-1,1} \\ N_{i,1} \end{Bmatrix} \Rightarrow \{\cdots\} \Rightarrow \begin{Bmatrix} N_{i-p,p} \\ \vdots \\ N_{i,p} \end{Bmatrix}$$

Figure 3.14: Nonzero basic functions in an inverted triangular.

With these basic functions, we can compute one point in function "Point_On_Surface". By a bidirectional circulation of call function "Point_On_Surface", a series of point on the surface are computed and connecting these points yields to a surface.

3.3.2 Computing U, V, and P from Point

Computing one point on a NURBS surface requires the knowledge of *U*, *V*, and *P* where *P* is the control point and generally not on the surface (See Figure 3. 15).

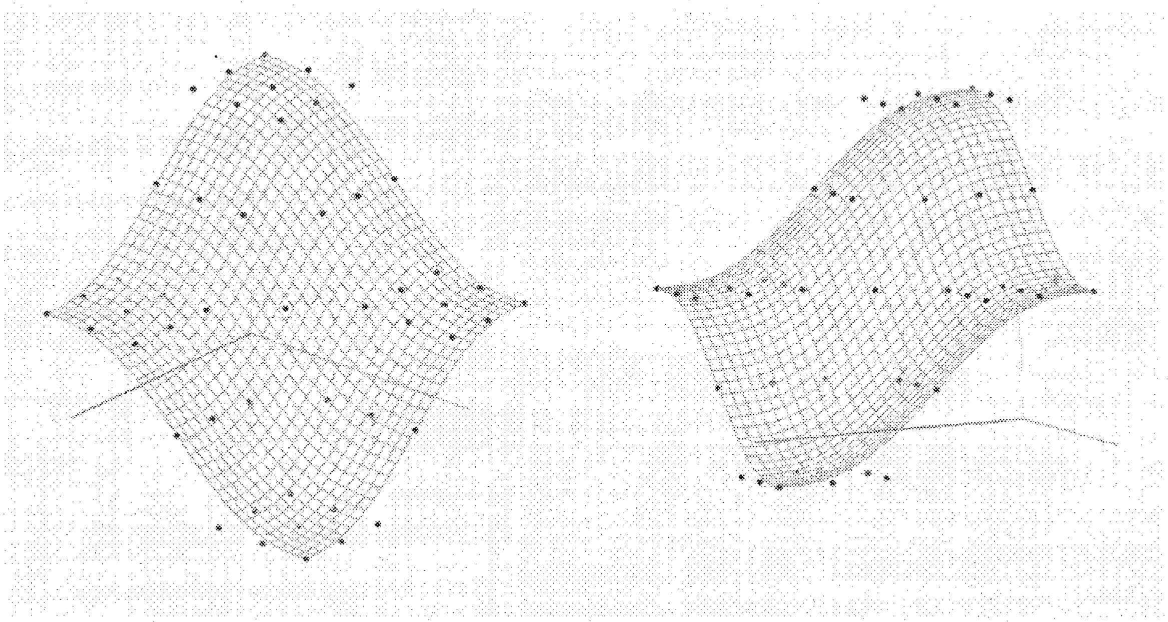

Figure 3.15: A NURBS surfaces with control points.

In our program, we need the NURBS surface to interpolate a series of specified points. Obviously, control points are not these points. We have a series of specified points and what we want is a surface to interpolate them. The overall process of our algorithm is given in Figure 3.16.

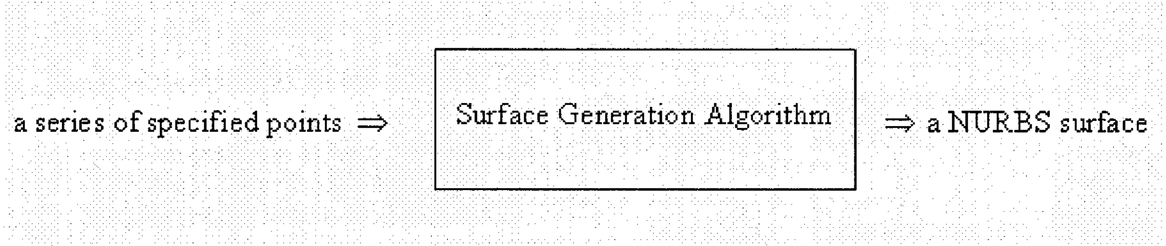

Figure 3.16: Surface generation process.

The surface generation can be divided into two parts: computing U, V, and P from previous selected points and computing points on NURBS surface using U, V, and P. The process of converting control points into NURBS surfaces is given in Figure 3.17.

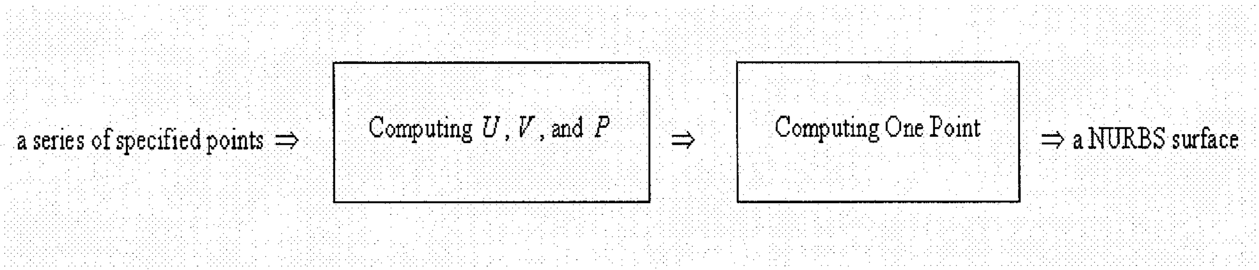

Figure 3.17: Two steps of surface generation process.

Up to this point, generation of NURBS surfaces from series of selected control points is shown. Yet, the first part, computing the U, V, and P from a series of specified points is not discussed. In this section, we compute U, V, and P based on a set of points which are selected from the rough image.

This is the fitting problem, i.e. NURBS curves and surfaces are constructed to fit an arbitrary set of geometric data such as points and derivative vectors. There are two types of construction, interpolation and approximation. In interpolation, we construct a curve or surface which satisfies the given data precisely, e.g., the curve or surface passes through the given points. In approximation, we construct curves and surfaces which do not necessarily satisfy the given data precisely. Rather they only approximate the given data. In some applications, such as generation of point data by use of coordinate measuring devices or digitizing tablets, or the computation of surface and surface intersection points by matching methods, a large number of points need to be generated. Such point generation techniques may include significant measurement or computational noise. In this case, it is important for the curve or surface to capture the shape of the data, but not to wiggle its way through every point. In approximation, it is often desirable to specify a maximum bound on the deviation of the curve or surface from the given data, and to specify certain constraints, i.e., data which is to be satisfied precisely.

Curve fitting is a challenging problem in computational geometry. Its complex nature has attracted many researchers to search efficient solutions to the problem. Majority of such works are heuristics, and there are usually no unique or clear–cut accurate answers. A fundamental problem is that the given data never specifies a unique solution. There is infinite number of NURBS curves and surfaces which can interpolate the same data set, or can approximate them. A designer using fitting algorithm software often find that the resulting curves or surfaces are not the ones he/she desires to achieve rather the geometric shapes which are mathematically correct and satisfies the input data.

Input to a fitting problem generally consists of geometric data, such as points and derivatives. Output is a NURBS curve or surface, i.e., control points, knot vectors, or weights. In our implementation, we write the function "Surface_Interpolation" to calculate the necessary control points and knot vectors in order to write the parametric surface equation.

```
Surface_Interpolation (m, n, Q, U, V, P)
{
        Compute the knot vectors and control points for the NURBS surface to
        interpolate through $mn$ points.
        The outputs are the U and V which are the knot vectors for $u$ and $v$
        directions and P which are the control points.
```

```
}
```

In our program, we use the selected points from rough image as the Q and let the user give a reasonable m and n size. Reasonable means that the product of m and n is equal to the number of the selected points. We input these three data into the function "Surface_Interpolation" and the output is the knot vectors U and V and control points P. Then we call the "Surface_Point" function to calculate some amount specified by the user of points on the surface. The NURBS surface is constructed by connected these points. The previous selected points are on the surface. The final result looks like that the NURBS surface interpolates these selected points.

3.4 Surface Modification

As described in section 3.3, the most important advantage of working with parametric equations is that they are quite flexible to modify. Up to this point, the generation of a rough sketch of the product in 3D space and its representation in parametric surfaces are completed. In this section, the image perfection process is discussed. The initial parametric representation of the rough sketch is a good approximation, yet it includes some degree of discrepancy. Now, parametric surface can be modified based on the rough sketch until the satisfactory shape of the object is achieved. Parametric surface modification process and use of rough sketches as the benchmark during modification process are demonstrated in Figures 3.18 and Figure 3.19 respectively.

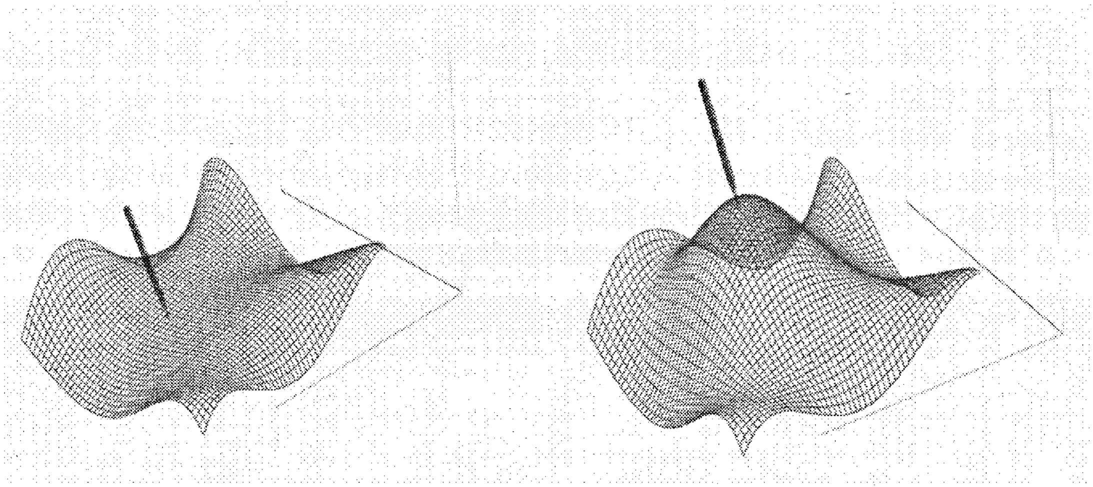

Figure 3.18: A NURBS surfaces modifies automatically when its interpolated points are modified.

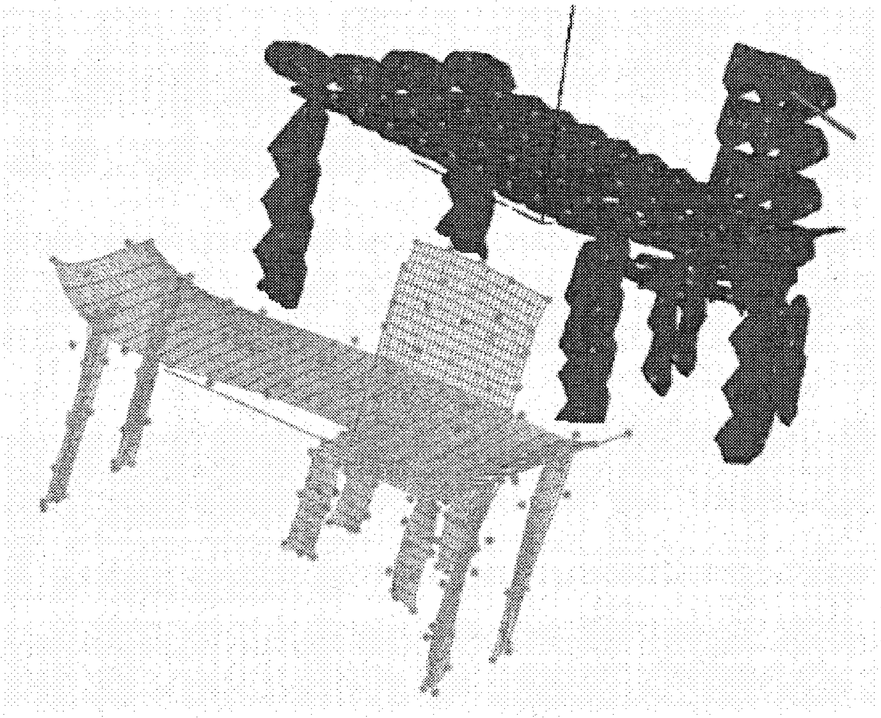

Figure 3.19: Comparing parametric surfaces with the rough sketch during the modification.

3.5 Program Execution

Once the free-form sketching program is executed, the user will have several options to choose from. Available options in the current version of the software are shown in Figure 3. 20 which is a snapshot of the software while being used.

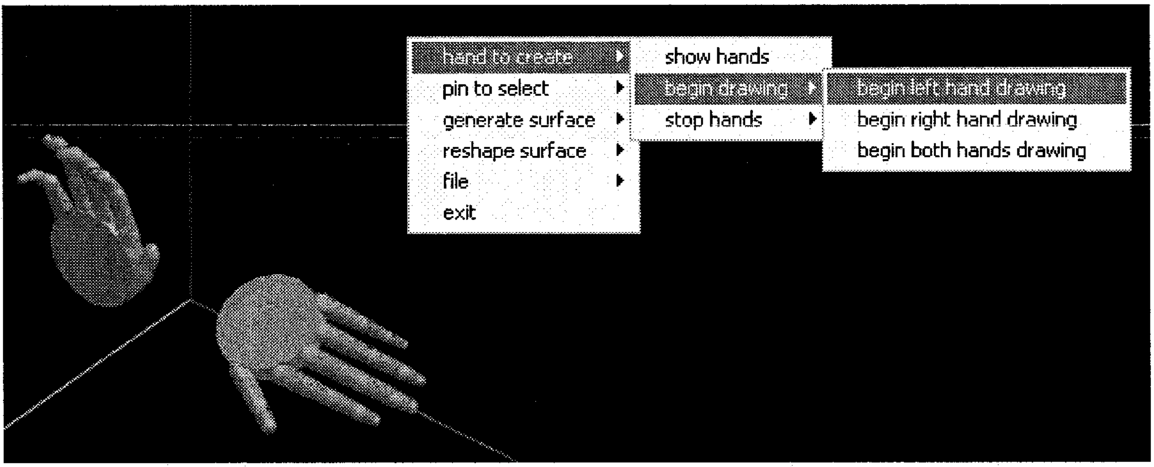

Figure 3.20: Options in free-form sketching system.

Once an option is chosen, designer can start performing his/her desired actions using all three dimensions available in the virtual world. For example, series of convex surface patches are drawn to the 3D space as shown in Figure 3.21. In this figure, the advantage of the described method comparison to the work of Krause et al. (2004) (right image on Figure 3.21) can be distinguished easily. In their work, Krause is attempting to reach the object shapes by creating 3D segments. From the figure, it is clear that connecting these independent splines to generate surfaces of objects is extremely challenging. Whereas in our implementation, placing surface patches directly into an empty space naturally generate a reference point to work with.

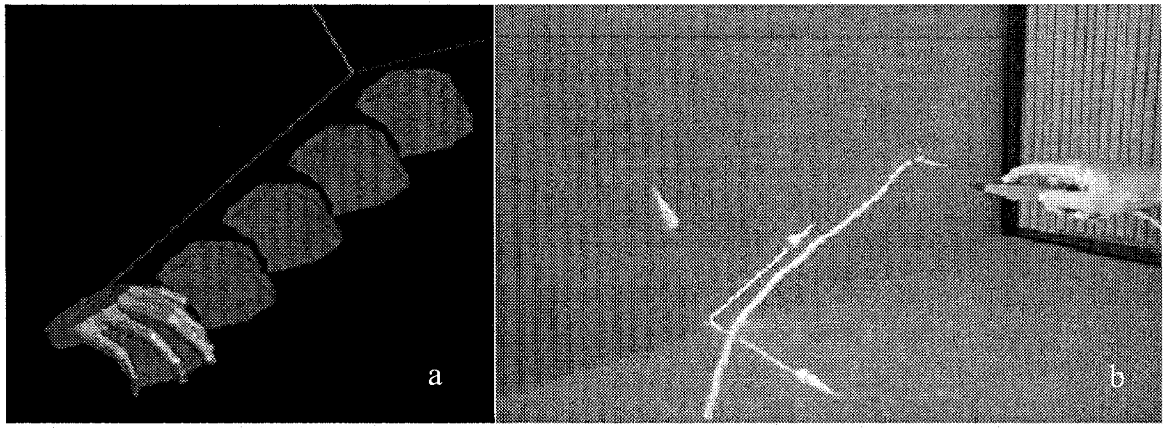

Figure 3.21: a. Surface generating by free-form sketching system; b. Spline generating by Krause (2004).

In order to complete the design process, the free-form sketching system provides additional functionalities which can be controlled by mouse or keyboard actions. Activating the control point selection process, control of virtual pen, selection of control points (accepting a found intersection), generation of the surfaces and modification of the initial surfaces are all controlled by selecting various options through graphical user interface. In Figure 3.22, the snapshots of some of these functionalities are presented.

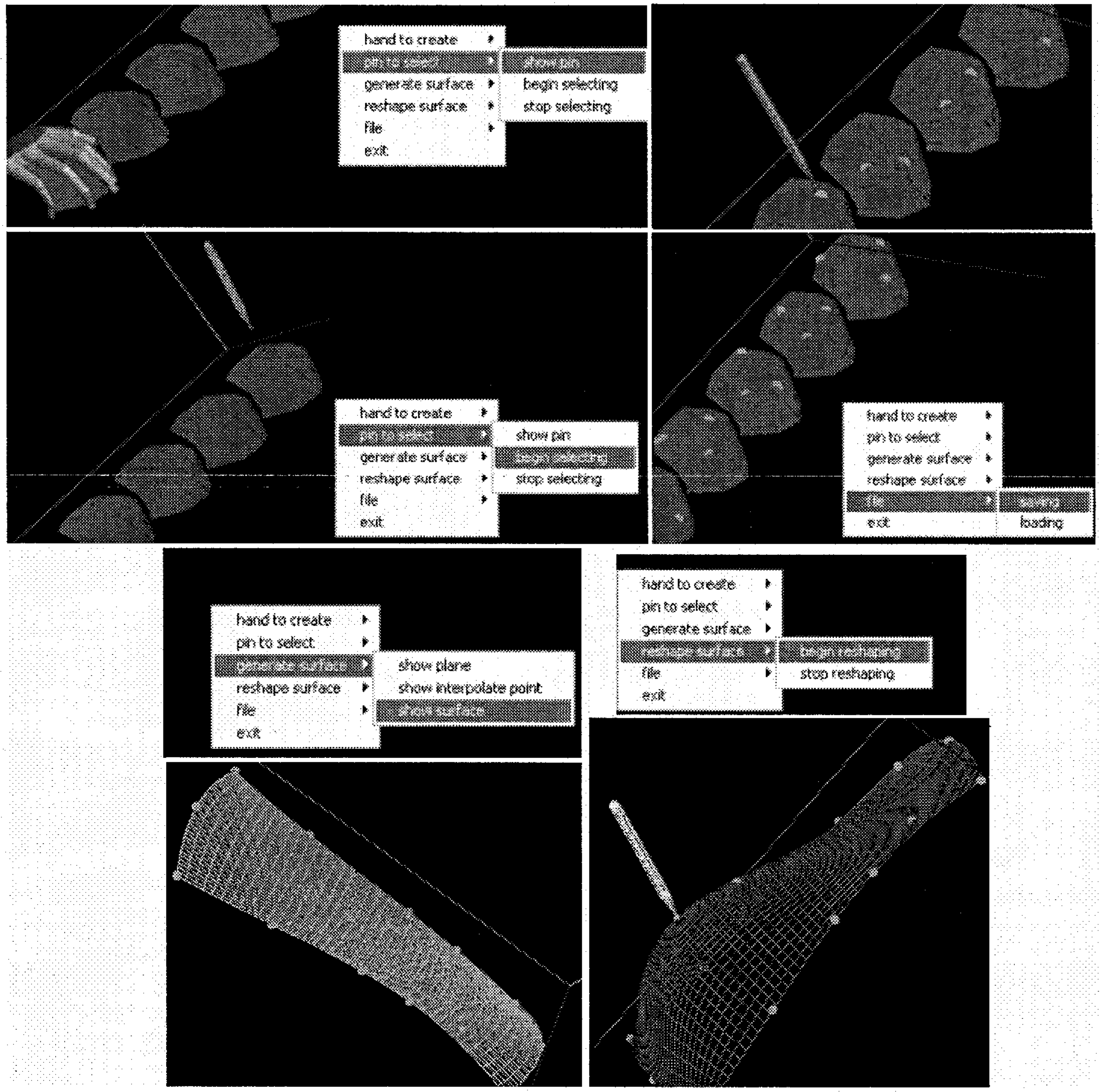

Figure 3.22: Series of options supported by the system to reach the final design.

Chapter 4

Results

In this chapter, we demonstrate the capability of our free-form sketching system in three different examples. We have used series of figures to explain the development of each design. Forming the rough sketch, selection of control points, generation of the initial parametric representation of the sketch and finally perfecting the parametric surfaces are shown in consecutive figures for each example.

4.1 Table and Chair

The rough sketching of the table and chair example is given in Figure 4.1. In figure 4.2 the control point selection process is displayed. Finally, in Figure 4.3 the parametric surface representation and the surface perfection processes are illustrated.

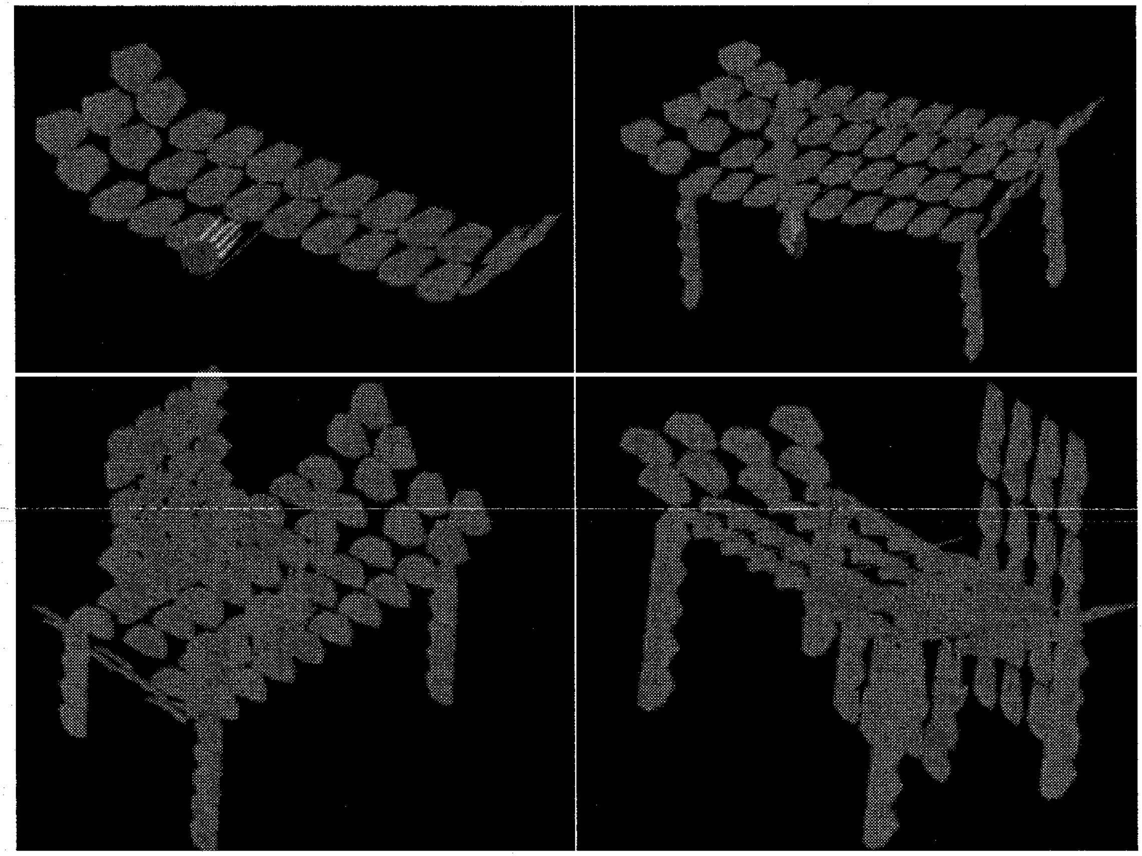

Figure 4.1: Rough sketching of table and chair are created.

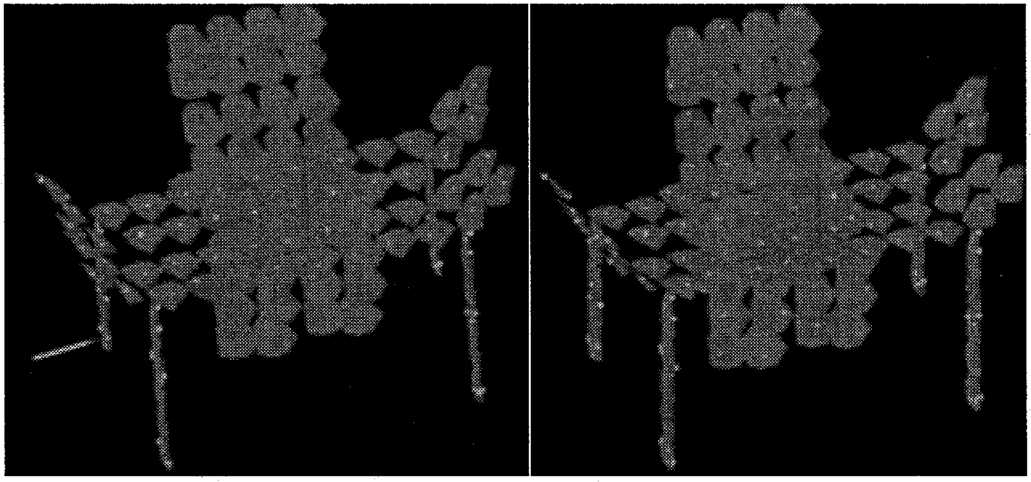

Figure 4.2: Control points selection on the surface of rough sketch.

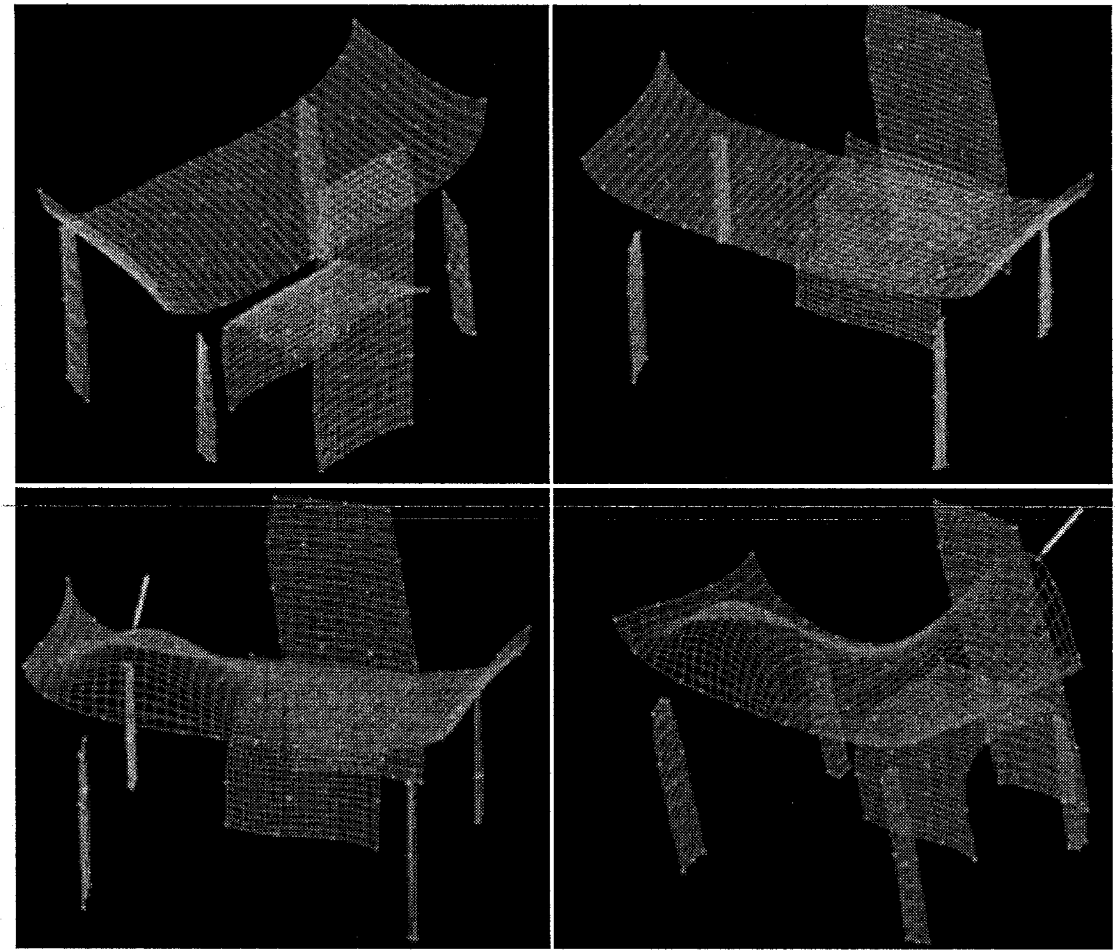

Figure 4.3: Parametric surface representation of table and chair example.

4.2 Car Body

In the next example we demonstrate the process of designing an automobile body from scratch. Figures 4.4, 4.5 and 4.6 demonstrate the rough sketching, control point selection and parametric surface creation processes respectively.

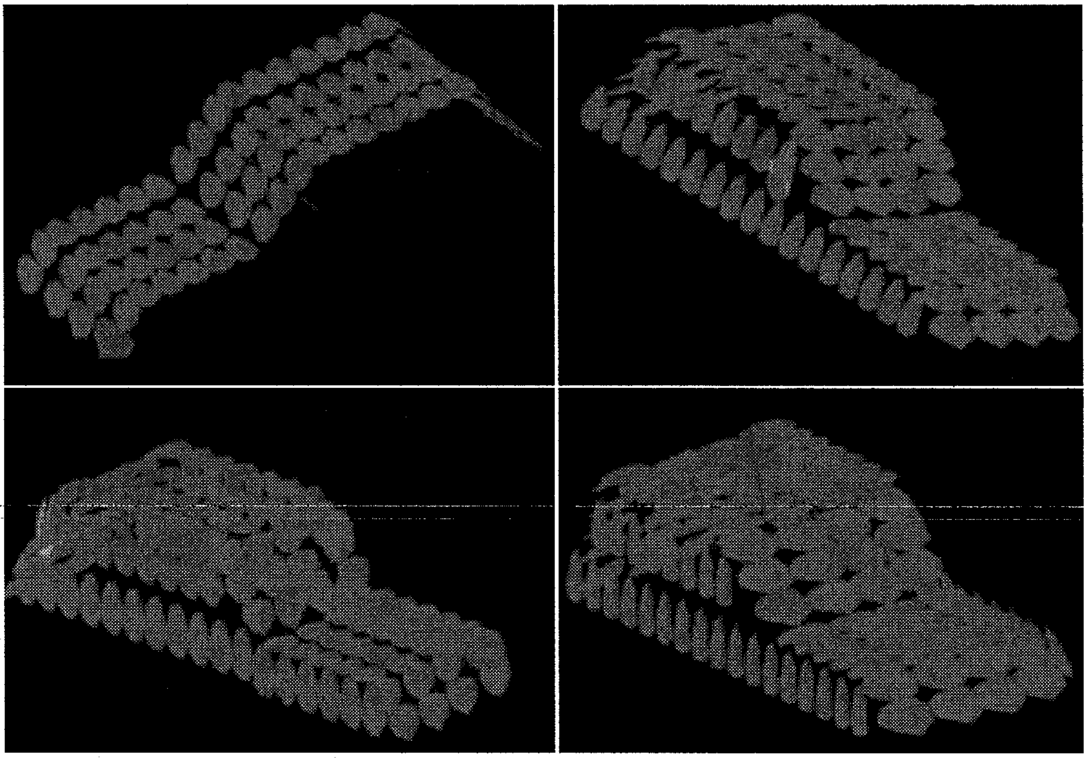

Figure 4.4: Rough sketch of automobile body.

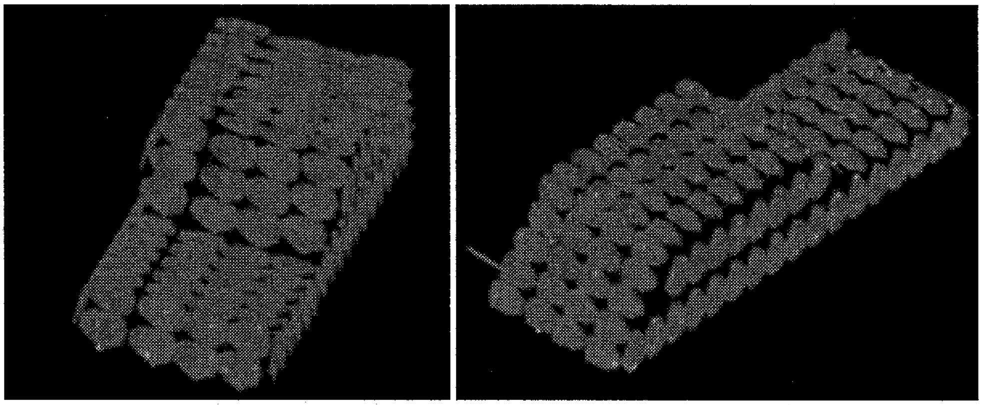

Figure 4.5: Control point selection from the surface of rough sketches of the automobile.

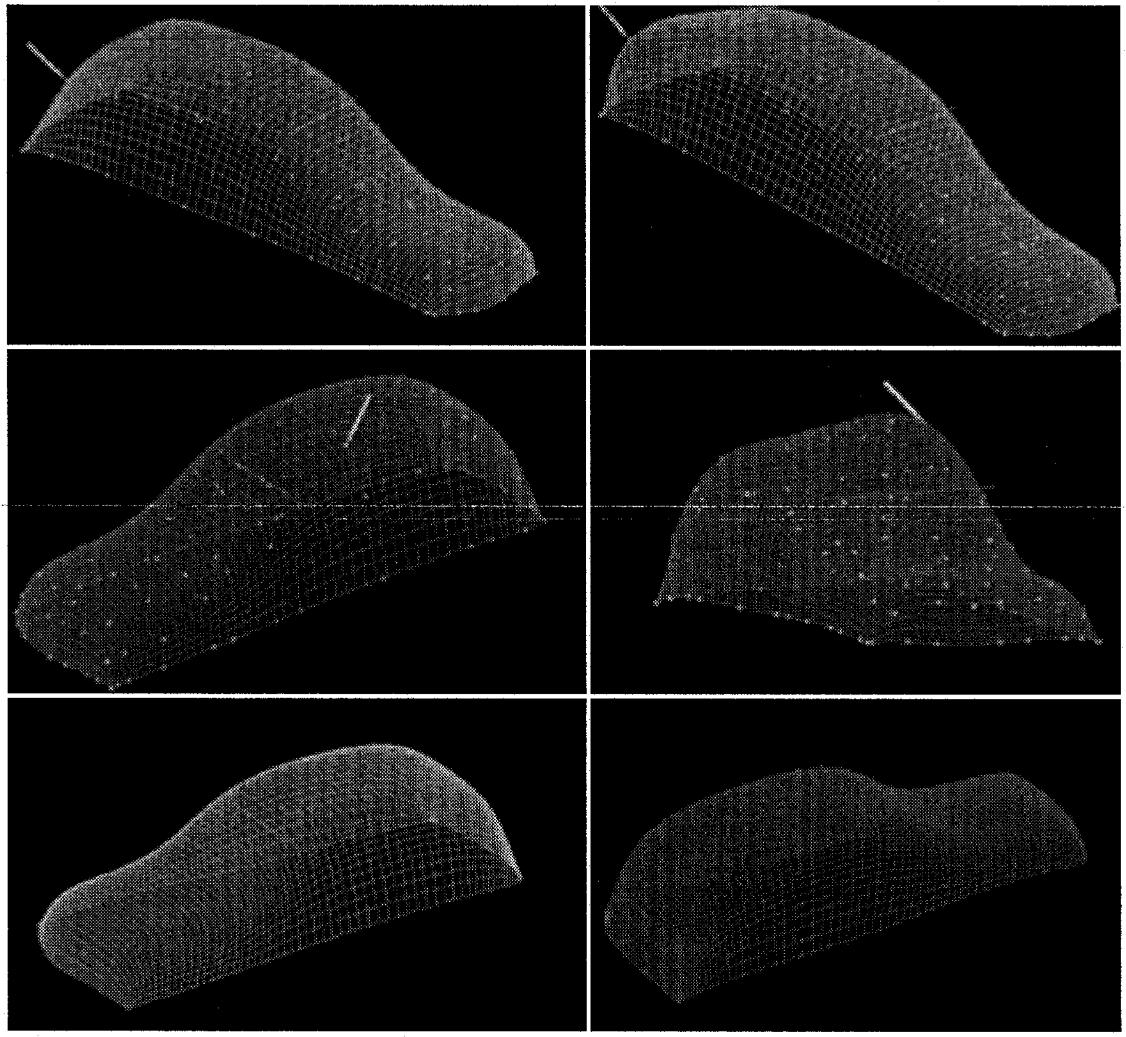

Figure 4.6: New auto body design in parametric form.

4.3 Plane Body

In the final example, the design of an airplane model is illustrated. The idea generation in rough sketch format, control point selection, generation of parametric surfaces and perfecting the design are demonstrated in Figure 4.7, 4.8 and 4.9 respectively.

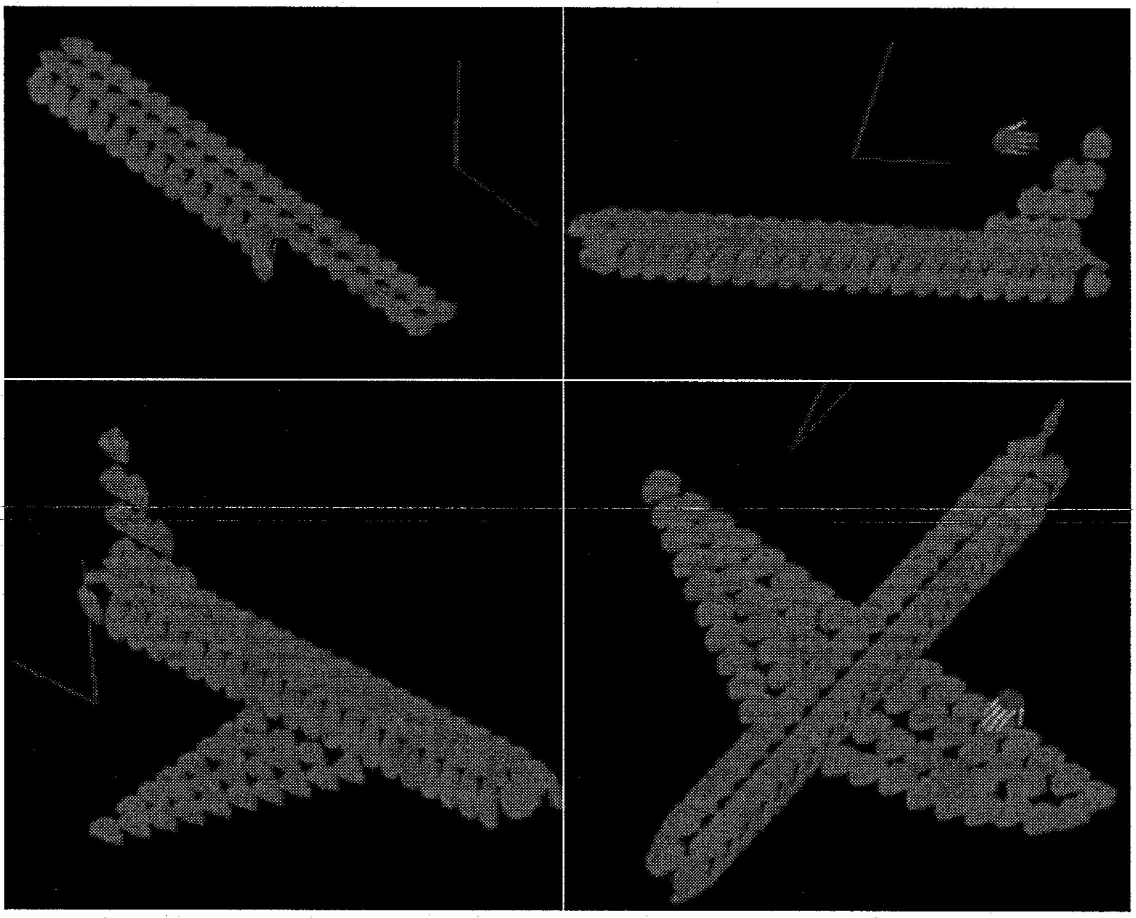

Figure 4.7: Idea generation in rough sketch form for initial airplane design.

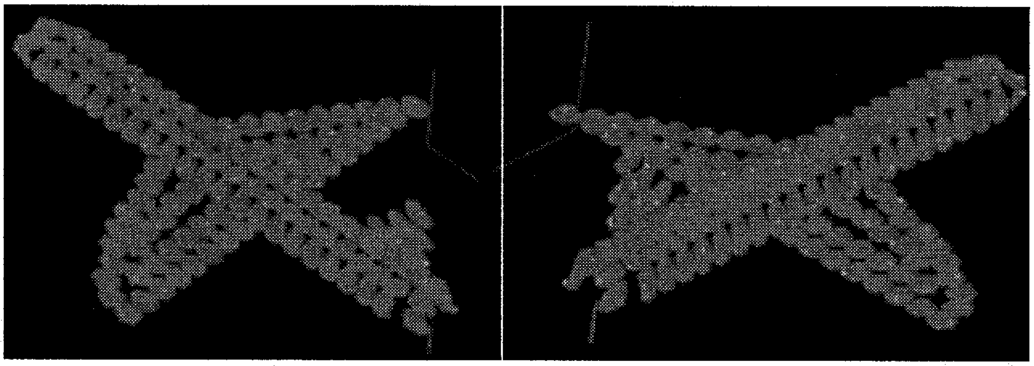

Figure 4.8: Control points selection on the surface of the initial sketch.

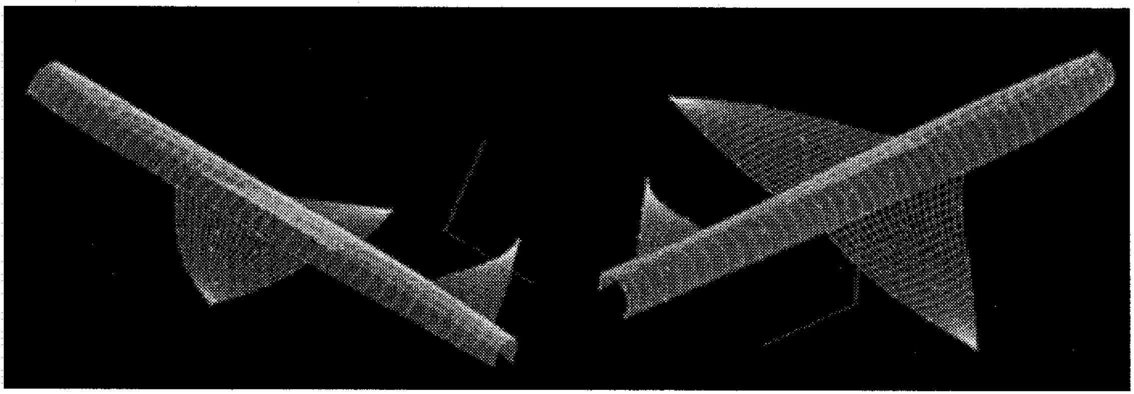

Figure 4.9: Initial parametric representation of the airplane.

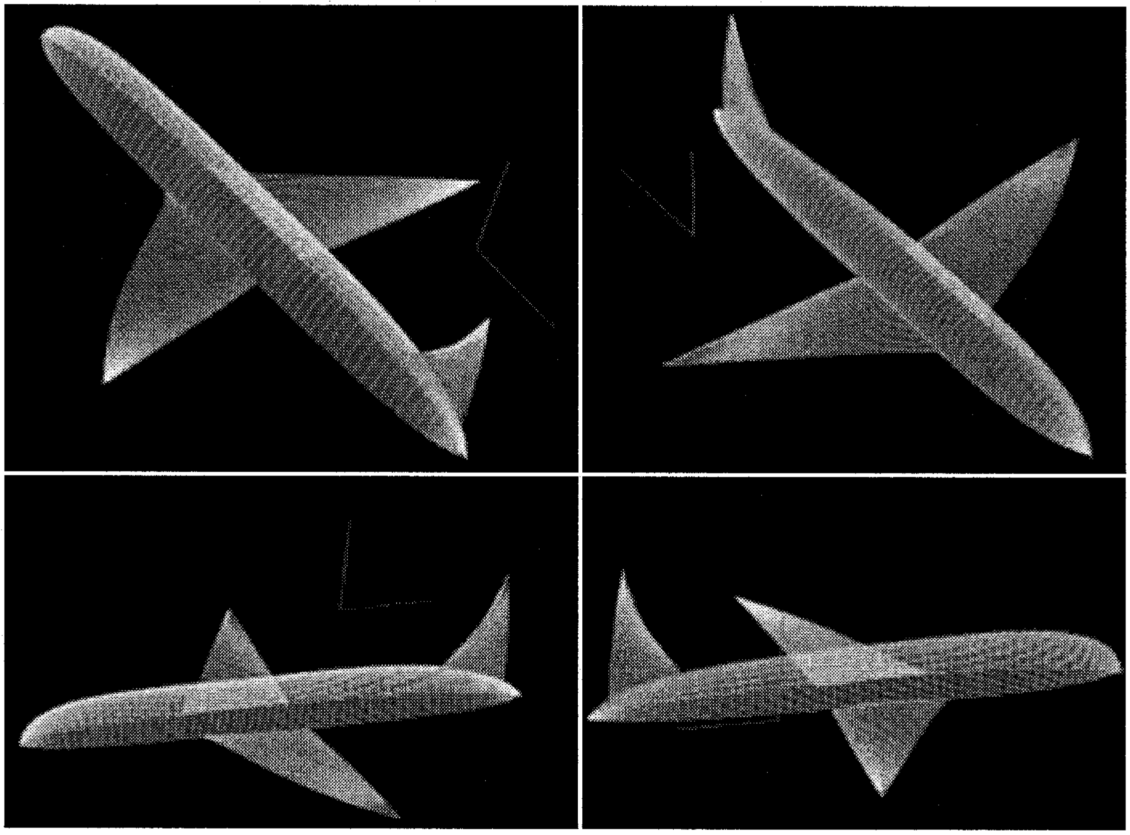

Figure 4.10: Final shape of the newly designed airplane.

Chapter 5

Conclusion and Future Work

We have introduced an approach that enables industrial designers to create conceptual product models directly in 3D space freely and intuitively using VR technology. The objective of this work is to computerize the concept design process. Concept design is the only design stage that is not done in the computer. Concept design is different from mechanical design. It requires natural and intuitive design environment. With the described VR based free-form sketching system, the designers can exercise their inspirations rapidly and intuitively in 3D space by directly creating computerized representations of the design. This system is realized by using a two-step sketching approach. In the first step, polygon meshes are used to generate the rough images of the designed products. In the second step, parametric surfaces are generated to represent the product shapes based on rough images created in the first step. By using polygon meshes and parametric surfaces in two consecutive steps, we simply realize the computerization of concept design in 3D space. The difference between our work and the previous works in this area is that they just use polygon mesh or parametric surface in their system as the representation method. In most cases, drawing splines or connection different spline segments in 3D space is extremely challenging if it is not impossible. Although, pen and paper based sketching platforms provide a physical boundary for designers to start and end their sketches, in VR environment such natural condition does not exist. Hence, in order for designers to work in VR, they have to make several assumptions to deal with the absentees of a base work-bench. On the other hand the described sketching system in the thesis first generates a base which is the rough sketch for designers to use as a bench, an example and also a benchmark to create their design. In this regard, the proposed method is a naturalistic 3D space in VR for designers to present their ideas as product geometries.

Although we have developed the framework for the proposed 3D free-form sketching tool, there are still limitations. The free-form sketching system allows the designer to generate parametric surfaces of a product in segments. For example as shown in the table example first the surface of the table is generated, then the legs are attached to it. These sub-components have to be dealt individually. Yet, connectivity of assemblies should be supported by a number of rules such as sharing same control points, snapping two geometry, aligning geometries etc. that are frequently used in many CAD software. The second issue which was not addressed in this work is to add thickness to surfaces. Although conceptually, adding thickness to the object shapes is straight forward yet in practice this is still a challenging issue to address.

Beside some design related shortcomings of the developed system, there are several other performance related issues have not been addressed in the current version of the system. Various features can be included to increase the efficiency of the system. For example, mouse and keyboard commands can easily be replaced by voice activated commands. In the current version, mouse and keyboard oriented command selection is destructing the designers' concentration on the design, and also decreasing the overall speed of the design process. A voice recognition system can assists the designers in much more efficient way with minimal effect to the designers' concentration.

The other difficulty we have experienced during image generation, the use of hand or fingers is not always the best option. There are cases where small details are handled, the size of hand or finger becomes a challenge to work with. Integration of more accurate tools such as haptic devices can assist addressing these challenges. Haptic devices can also be used to replace the virtual pen we used to select control points and modify the surfaces. Haptic devices enable users to feel touching feeling. When control points are selected, natural sense of touching will increase the speed of design process.

Finally the ultimate goal of this project is to integrate the whole product development process from concept design to mechanical design into a computerized system.

Bibliography

[1] A. Akgunduz, H. Yu, "*Two-Step 3-Dimensional Sketching Tool for New Product Development*", 2004 Winter Simulation Conference (WSC '04), Washington, DC, USA, December 5-8, 2004

[2] A. Cohen, C. Rabut, L. L. Schumaker, "*Curve and Surface Fitting: Saint-Molo*", VU Press, 1999.

[3] A. G. Hauptman, "*Speech and Gestures for Graphic Image Manipulation*", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Wings for the Mind, Pages 241-245, 1989.

[4] A. Stork, O. Schimpke, D. R. Amicis, "*Sketching Free-Forms in semi-immersive Virtual Environments*", ASME 2000 Design Engineering Technical Conferences and Computer and Information in Engineering Conference, Baltimore, Maryland, 2000.

[5] C.-C. P. Chu, T. H. Dani, R. Gadh, "*Multi-sensory User-interface for a Virtual-reality-based Computer Aided Design System*", Computer-Aided Design, Vol. 29, No. 10, 709-725, November 1997.

[6] C. Luciano, "*Animations of Tele-collaborative Avatars with Real-Time Inverse Kinematics Algorithm Using Dual Quaternion*", Master thesis, Industrial Engineering and Operations Research, University of Illinois at Chicago, 2000.

[7] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE" Proceedings of

SIGGRAPH 93 Computer Graphics Conference 08/01/1993 - 08/01/1993, ACM SIGGRAPH, pp. 135-142.

[8] D. Browning, Cruz-Neira, C., Sandin, D., DeFanti, T., "The CAVE Automatic Virtual Environment: Projection-Based Virtual Environments and Disability" Proceedings of the First Annual International Conference, Virtual Reality and People with Disabilities 06/01/1993 - 06/01/1993.

[9] D. Sandin, D. Browning, "*Evaluating the Visual Aspects of Product Design: Simulating Ergonomic Studies From Within Virtual Realities*", Manufacturing Research Center at the University of Illinois, May 1991.

[10] D. Schmalstieg, L. M. Encarnacao, Z. Szalavari, "*Using Transparent Props for Interaction with the Virtual Table*", ACM SIGGRAPH Symposium on Interacitve 3D Graphics (I3DG'99), Atlanta, GA, ACM Press, 1999.

[11] E. Sachs, A. Roberts, D. Stoops, "*3-Draw: a Tool for Designing 3D Shapes*", IEEE Computer Graphics& Applications vol. 11, no. 6, pp. 18-26. November-December 1991.

[12] F. Bruno, M. L, Luchi, M. Muzzupappa, S. Rizzuti, "*The Over-Sketching Technique for Free-Hand Shape Modelling in Virtual Reality*", In Proceedings of Virtual Concept 2003, Biarritz – France, November 5-7 2003.

[13] F-L. Krause, M. Gobel, G. Wesche, T. Biahmou, "A Three-stage Conceptual Design Process Using Virtual Environments", The 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2004, Plzen - Bory, Czech Republic, February 2 - 6, 2004

[14] G. Burdeau, M. Srinivasan, "*Haptics in Virtual Environments Workshop*", Workshop Notes of the Annual IEEE Virtual Reality '2000 Conference, Rutgers University, New Brunswick, N.J., March 2000.

[15] G. Wesche, "*Freedrawer: a Free-Form Sketching System on the Responsive Workbench*", ACM symposium on virtual reality software and technology, Banff, Alberta, Canada, November 2001.

[16] G. Wesche, M. Droske, "*Conceptual Free-Form Styling on the Responsive Workbench*", Proceedings of the ACM symposium on Virtual reality software and technology, Pages 83-91, Seoul, Korea, 2000.

[17] H. Yu, and A. Akgunduz, "*Free Form Sketching in 3D Using Uniform Spherical Reference Points*", HCI International 2005, July 2005 in Las Vegas , US

[18] I. E. Sutherland, "*The Ultimate Display*", Proceedings of IFIP Congress, pp.506-508, 1965.

[19] I. Zeid, "*Cad/Cam* Theory and Practice", McGraw-Hill College, 1991.

[20] J. Dionisio, V. Henrich, U. Jakob, A. Rettig, R. Ziegler, "*The virtual touch: Haptic Interfaces in Virtual Environments*", Computers and graphics, 21 (4): 459-468, July-August, 1997.

[21] J. M. Zheng, K. W. Chan, I. Gibson, "A New Approach for Direct Manipulation of Free-Form Curves", in UROGRAPHICS 98, volume 17, pages C327–C334, 1998.

[22] L. Cutler, B. Frohlich, P. Hanrahan, "*Two-Handed Direct Manipulation on the Responsive Workbench*", Symposium on Interactive 3D Graphics, 1997.

[23] L. M. Encarnacao, O. Bimber, D. Schmalstieg, S. D. Chandler, "*A Translucent Sketchpad for the Virtual Table Exploring Motion-based Gesture Recognition*", 1999 Eurographics Conference, Milan, Italy, 1999.

[24] M. F. Deering, "*HoloSketch: A Virtual Reality Sketching / Animation Tool*", ACM Transactions on Computer-Human Interaction, Vol. 2, No. 3, Pages 220-238, 1995.

[25] M. Tovey, J. Owen, "*Sketching and Direct CAD Modeling in Automotive Design*", 2000 Elsevier Science Ltd, Pages 569-588, 2000.

[26] O. Bimber, L. Miguel, A. Stork, "*A Multi-layered Architecture for Sketch-based Interaction within Virtual Environments*", Computers and Graphics, 24(6), Pages 851-867, 2000.

[27] O. Tolba, J. Dorsey, L. Mcmillan, "*A Projective Drawing System*", Proceedings of the 2001 Symposium on Interactive 3D Graphics, Pages 25-34, 2001.

[28] P. Dierckx, "*Curve and Surface Fitting with Splines*", Computer Science Department, Katholieke Universiteit Leuven, Belgium, 1995.

[29] P. Frederick, "*What's Real About Virtual Reality*", IEEE Computer Graphics and Applications, Vol. 19, Pages 16-27, 1999.

[30] P. Lancaster, "*Curve and Surface Fitting: An Introduction*", Academic Press, September 1986.

[31] Q. Cai, A. Mitiche, J. K. Aggarwal, "*Tracking Human Motion in an Indoor Environment*" Proceedings of Second International Conference, Image Processing, pp. 215-218, Washington, D.C., October 1995.

[32] R. C. Zeleznik, K. P. Herndon, J. F. Hughes, "*Sketch: An Interface for Sketching 3D Scenes*", SIGGRAPH 1996 Proceedings, Pages 163-170, 1996.

[33] S. K. Card, J. D. Mackinlay, G. G. Robinson, "*The Design Space of Input Devices*", in Proceedings of the CHI'90 Conference on Human Factors in Computing Systems, Pages 117-124, 1990.

[34] S. Steven, P. Schroder, "*Surface Drawing*", Technical Report CS-TR-99-03, Caltech Department of Computer Science, 1999.

[35] S. Yoshida, S. Miyazake, T. Hoshino, T. Ozeki, J. Hasegawa, T. Yasuda, S. Yokoi, "*Spatial Sketch System for Car Styling Design*", International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII, Part B5, Amsterdam, 2000.

[36] T. Baudel, "A *Marked-based Interaction Paradigm for Free-hand Drawing*", UIST 94, pages 185-192, 1994.

[37] T. H. Dani, R. Gadh, "*Creation of Concept Shape Designs via a virtual Reality Interface*", Cmoputer-Aided-Design, Vol. 29, August 1997.

[38] T. Igarashi, S. Matsuoka, H. Tanaka, "*Teddy: A Sketching Interface for 3D Freeform Design*", ACM SIGGRAPH'99, Los Angels, 1999.

[39] V. Popescu, G. C. Burdea, M. Bouzit, "*Virtual Reality Simulation Modeling for a Haptic Glove*", Proceedings of Computer Animation 99 Conference, 195-200, 1999, University Geneva, Switzerland, May 26-29, 1999.

[40] W. Buxton, G. Fitzmaurice, R. Balakrishnan, G. Durtenbach, "*Large Displays in Automotive Design*", IEEE Computer Graphics and Applications 2000, Pages 68-75, July-August 2000.