Real-time Parametric Surface Modeling For Conceptual Design

Lan Wu

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

August 2008

Your file Votre référence
ISBN: 978-0-494-45356-8
Our file Notre référence
ISBN: 978-0-494-45356-8

ABSTRACT

Real-time Parametric Surface Modeling For Conceptual Design

Lan Wu

Conceptual design is the initial stage of the product design cycle. In this stage, the structural details of the products require less attention than the functionality and style. Designers need an environment where their creative inspiration and imagination can be expressed intuitively and instantaneously without limitations. Paper-pen based traditional tools are still common in the practice. However, pen and paper based methods limit the expression of ideas (conceptual design) in the form of 2D images.

This thesis introduces a free-form 3D sketching technique with an objective to improve the conceptual design stage. Improvement comes from two characteristics: firstly, the designed system enables designers to use 3D digital space to summarize their imaginations; secondly, the toolkit provides an unconstrained environment where designers/artists can demonstrate their imaginations without dealing with details. Ultimate objective of this line of the research work is to develop a free-form sketching tool where users are not familiar with the system; rather they can naturally use the system in order to complete their initial design. In order to have a stereoscopic impression or called intuitive sensor of the designed products in the conceptual design stage, a direct 3D display is essential. Hence, in this research work, we introduce an unconstrained conceptual design environment that enables designers to display their ideas instantaneously and inspirationally without any constraints in computer in the form of 3D objects directly under the support of VR technology. The developed toolkit provides: a virtual space for designers to summarize their imaginations in 3D; a module to enable smooth connection of individual surface patches; finally, a method to give surface thickness to the images. Hence the resulting images are ready to be used within the existing CAD software, for design review purposes and manufacturing planning efforts.

ACKNOWLEDGEMENTS

I would like to acknowledge that without the assistance and support of many people directly or indirectly, I would not have completed this thesis.

First and foremost, I would like to express my thanks and appreciation to my supervisor Dr. Ali Akgunduz for his constructive advice, constant guidance, financial support and encouragement throughout this work. I am grateful the time he devoted to guide me to understand the concepts involved in this thesis. His constructive advice and constant guidance played the significant role. Without the assistance from him, this thesis would not have been existed.

I would like to thank my mother and all my family members for their continuous encouragement and support during my master. I express my gratitude to all faculty members and staffs of the Department of Mechanical and Industrial Engineering at Concordia University not only for providing me with a nice studying and working environment but also for contributing to my educations.

Finally, thanks to all those people who provided me help during my master.

Dedicated to my mother, and my family members…

Table of Contents

LIST OF FIGURES

LIST OF TABLES

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
VO	Virtual Objects
VE	Virtual Environment
CAVE	Cave Automated Virtual Environment

Chapter 1

Introduction

Virtual Reality (VR) is an interactive, real-time, immersive and 3-dimensional (3D) computer graphic simulation mainly used to replicate physical systems in computer. Among its several advantages, real-time interaction and stereoscopic display (enable immersion) characteristics are the two main features that enabled the realization of the project introduced in this thesis. In its simplistic fashion, our main objective in this research work is to replace paper-pen based sketching environment in Computer in 3D format. Real-time interaction enables performing the actions and immersion provides the feeling of presence in the Virtual Environment (VE).

These characteristics of VR enabled the technology to be used in various applications that include entertainment, medicine, architecture, and design and manufacturing. Although, we find large body of scientific results and industry applications of VR technology, the use of VR in conceptual design, particularly for sketching purpose is relatively new. Although the potential benefits of VR in free-form sketching is significant, there is much more work to be done before this technology can be effectively used to capture the feelings and imaginations of designers in 3D space.

1.1. Problem Area

Commercial Computer Aided Design (CAD) software provides powerful design tools and methods to replace the traditional blue-print based design methods and free designers from heavy calculation, checking, and image re-drawing works. CAD software has demonstrated its powerful functionality to perform the detail design. These capabilities not only support the 3D representation of the objects but also support various engineering and dynamic analysis as well as simulation studies to ensure the completeness of the design.

However, they are still awkward in the products' innovation and modification processes. For example, a local modification of products' 3D outline needs a series of modifications from 2D drafts, which will cause a series of recalculation, checking, analysis, redrawing works to be performed, and thus makes the design work to be a boring and heavy process. Besides, using the current CAD software to display the 3D images for the designed objects need to provide a series of relevant dimensions information. Therefore, current CAD software can't meet the conceptual design requirements. Conceptual design requires designers to be able to display their ideas quickly and accurately without limits and easy to achieve the results modification in real-time. Under this circumstance, current CAD software shows insufficient and inefficient support for conceptual design; hence the traditional paper-pen based method for conceptual design is still favorable among many designers.

Although the paper-pen based traditional method can reach the requirements of the conceptual design, which enable the designers to display their ideas, inspirations, and imaginations in real-time; however, this traditional design method only can only provide 2D images of the designed object. In conceptual design stage, designers are lack of stereoscopic impressions for designed objects. Designers have to work on their design process based on their spatial imagination for designed products. To some extent, the errors existed in this spatial imagination and real output results can't be avoided and will affect the following design process directly. Additionally, modifications are not possible in the same blue-print, rather in most cases duplications of the same effort are required to correct design errors. Consequently, the paper-pen based conceptual design is time-consuming and may lead to higher design and manufacturing cost. Lack of 3D view during the sketching reduces the perception; particularly those who involved in design process not familiar with the sketching and computer aided design may not be able to contribute to the design accurately.

Based on the above argument, we can conclude that, the traditional conceptual design tools and current CAD systems are not serving the needs of today's desi gn challenges. Virtual Reality, relatively a new technology featured by its real-time simulation, presentation, and interaction has shown a great potential in the area of applied science e.g. medical, education, communication, training and so on. Its advanced capability in real-time simulation, presentation, and interaction between human and machines makes it a good candidate for the conceptual design. From our research point of view, VR provides a unique environment for the design process such as: real-time virtual

3D model presentation, real-time simulation and analysis, as well as real-time design modification. Hence, the objectives of this research work, 3D Free-form sketching toolkit, are possible to be realized.

In today's highly competitive market-place, rapid response to changing factors is crucial. Reduced cycle-time of product development is one of the key elements to rapid response to market. The proposed VR based free-form sketching technique has great potentials to reduce the product development cycle-time. It is therefore important to develop methods to achieve the complete combination for the VR technology and conceptual design process to meet the practical requirements of design-cycle so that the whole design process can be performed in Computer Aided Design environment.

1.2. Objectives

Although commercial CAD software have replaced the traditional design method based on papers and pens in the detail design area, due to their inadequate support for conceptual design, paper-pen based sketching is still preferred by most designers.

Recent advancements on both hardware and software enabled the possibilities to develop methods to perform the conceptual design in an interactive computer system instead of using paper and pen to obtain the desired effects such as: i) direct 3D stereoscopic visible effects; ii) real-time engineering and dynamic analysis capability; iii) reducing the cycle-time for the innovation and development of the products. Therefore, our general objectives in this research work are to achieve the integration of conceptual design and detail design in order to perform the complete design cycle computerized. The sub-objective of our research is to investigate a natural expression method to be used for conceptual design so that the designers can concentrate on their creations rather than spending time to understand the constraints that are associated to the software being used and learning the relevant mathematics. The proposed VR based sketching technique on the other hand makes the design task enjoyable as the advantages listed above suggest. Guided by these objectives, this thesis tackles the free and intuitive 3D design method from three aspects:

1) We introduce a mathematical and practical immersive system to build real-time images using parametric surfaces.
2) In the second step, we introduce our solution to stitch several design segments (patches) using parametric continuity.
3) Finally, we show methods to handle the surface thickness problem. Our ultimate goal is to use the designed system as the initial step towards a CAD modeling. In order to export the results of a design effort to commercial CAD software, thickness of the designed parts need be handled.

The details of these three aspects are given later in the thesis.

1.3. Our Contributions

1.3.1. Real-time parametric surface modeling

This thesis introduces a real-time parametric surface modeling method to design 3D objects in VR environment. In order to directly display the surface shape for the designed objects, we use reverse engineering method to construct parametric surface modeling so that an intuitive visible surface shape can be displayed instantaneously without large range of errors. By now, the reverse-engineering methods are mostly used to recover the 3D images of real objects by fitting the points cloud retrieved from the 3D scanners in computer or in VR environment. On the other hand, the proposed VR based sketching approach is relatively new. A similar attempt with several limitations was introduced in a MASc thesis work at Concordia University by Hang Yu (2006). Our work provides a more complete platform than the one Yu proposed.

1.3.2. Smooth connection methods

A complete model usually includes several sub-components. Although the paper-pen based sketching mechanism does not require a special technique to stitch or assembly various components that make the products, the proposed sketching technique must address the problem of connecting various sub-parts. The sketching technique proposed in this thesis models components using NURBS surfaces, which we call patches. Designing a complex product/object using a single NURB surface is either impossible or its outcome is not accurate. Hence, in such CAD systems, components (surface patches) should be built in pieces and stitched together using easy and quick connection methods to achieve the completeness, accuracy and the smooth transition between different patches. In this thesis, we reach the C^1 smooth transition between the stitched NURB patches.

1.3.3. The discussion on surface-thickness issue

Physical objects are not only surface patches. They are materials with volume. Hence, a realistic design tool should enable the physical realism. Realism not only improves the completeness of the shape but also makes various engineering analysis possible. Furthermore, conversion to commercial CAD software becomes more realistic. For this reason, in this thesis we introduce an approximation method that adds desired thickness to the designed NURB patches. The developed approximation method provides a thickness for an arbitrary designed 3D object and a desired visible effect for the almost symmetric objects.

Chapter 2

Background Information and Related work

VR is an interactive 3D simulation technology which enables users to be present with the Virtual Objects (VO) in an immersive environment. A successful VR system relies on adequate configuration and reasonable utilization of VR hardware/software. VR's advantages include stereoscopic visualization, immersive feeling, and interaction with the virtual objects, which make VR superior to the other simulation methods. With the help of certain technologies and VR hardware environment such as data-gloves, Head Mounted Displays (HMD), position trackers, force-haptic devices, stereo glasses, etc., users can enjoy higher level of realism in their simulations. Powered by the combination of different disciplines, VR now as a simulation tool plays an important role in the multi-purpose applications.

From the industry point of view, VR has been used for machine-operator interaction, customer assessment, maintenance/assembly process simulation, plant layout, etc purposes. Recently, interactive computer simulations, particularly immersive environments, such as VR, have become popular in product design. Drafting, evaluation, testing, remote-collaboration (collaborative design reviews) are some areas where VR simulations are effectively used. One of the areas where the potentials of VR are significant, yet implementation is cumbersome in the VR based sketching for 3D drafting. This thesis introduces a parametric 3D sketching methodology that demonstrates the feasibility of using VR as a free-form sketching system for product design particularly during the conceptual design phase. Prior to giving the details of the sketching technique introduced in the thesis, let us outline the characteristics of VR to motivate the topic.

2.1. Virtual Reality Hardware

2.1.1. Stereo Display Devices

The popularity of VR mostly comes from its immersive 3D display characteristics in which a user can be present and interact with the VE. The immersive characteristics of VR are due to the availability of stereo displays.

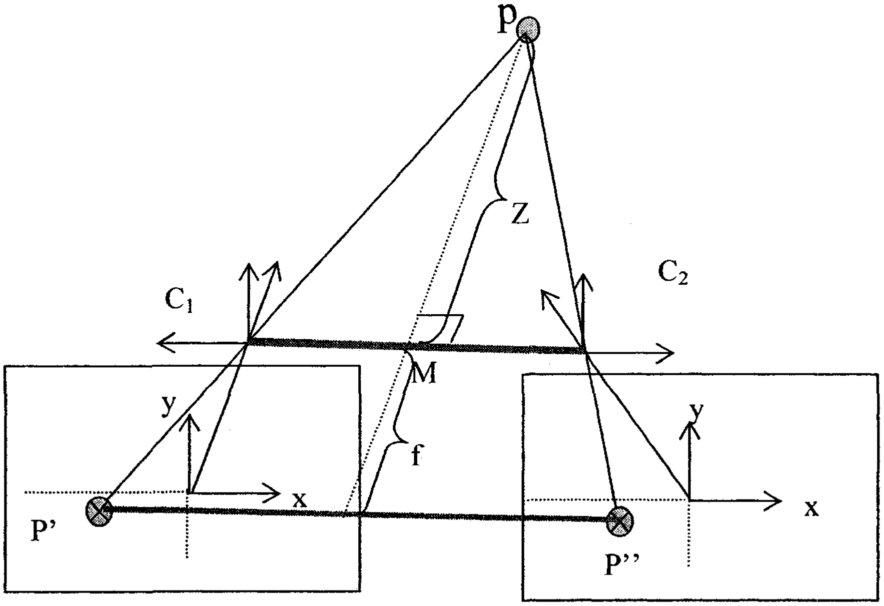


Figure 2.1: Stereo Displays and Immersion

Assume that the picture of a point P is taken from two locations, P' and P'' using an identical camera where the camera locations (C_1 and C_2) at each point and its focal length f are known (See Figure 2.1 for illustration). Then, it is possible to calculate the depth (Z) of the point P as:

$$\frac{Z}{Z+f} = \frac{d}{D} \Leftrightarrow z = \frac{df}{D-d}$$

Where d is the distance from C_1 to C_2 and the D is the distance from P' to P''.

The immersive displays also known as stereo displays take advantage of this simple rule in Figure 2.1 and display the same image in two different locations alternating at each frame (one image for left eye and the other image for the right eye). Due to the fast refreshment rate minimum 20 frames per second, user of such systems sees the environment in 3D even though the displayed scene is in 2D. Below various immersive systems are discussed.

- HMD is an optical device used for stereo demonstration (Figure 2.2). It consists of two scenes and one optical instrument which can transfer images between two scenes and eyes to generate virtual senses of the third dimension. A motion tracker is used to capture and record the current position and orientation of users' head so that the generated images can be viewed from current viewpoints. As a result, virtual stereo images and environment can be experienced by its users.

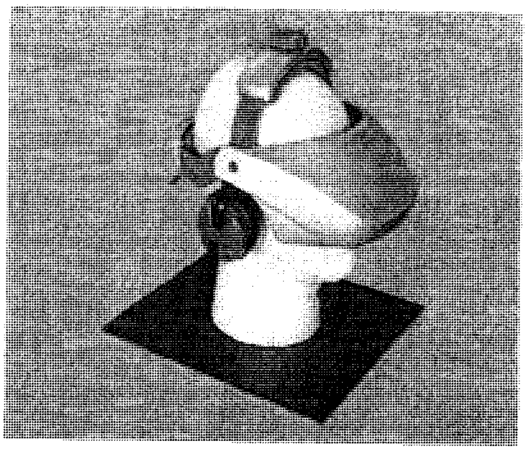


Figure 2.2: HMD from 5DT technology

- CAVE (Cave Automated Virtual Environment) provides users an immersive sense in an average room size cubical space (See Figure 2.3). The image generated by computer is projected on the wall and the floor. Users wearing stereo glasses can view the stereo image and experience the immersive sense. A head tracking device is used to track the change of the viewpoints and modify the projection of the image following the current position of the viewers.

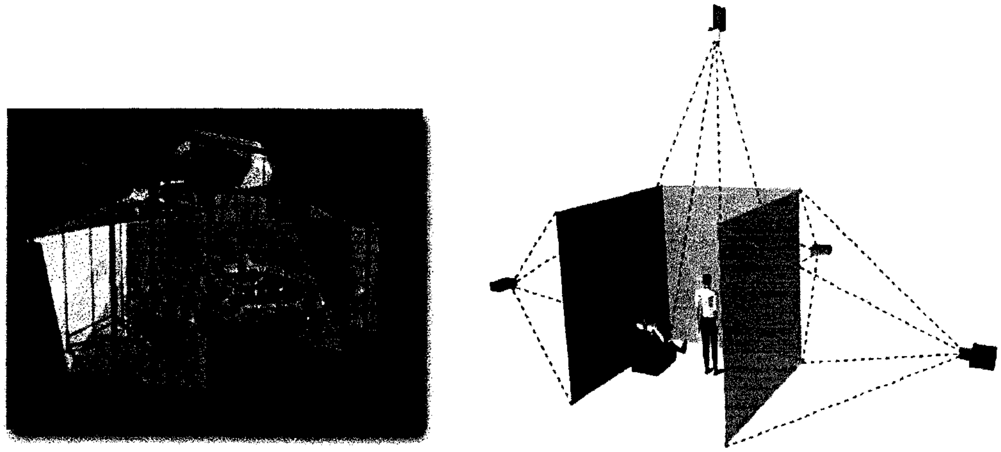


Figure 2.3: a) Projection structure in the PC environment; **b)** Normal configuration of the cave

- Boom is another stereo display instrument in VR system (See figure 2.4). Its mechanism is to fulfill some control functions and, with the help of optical instrument, display screen to create 3D effects. Its working principle is similar to that of HMD.

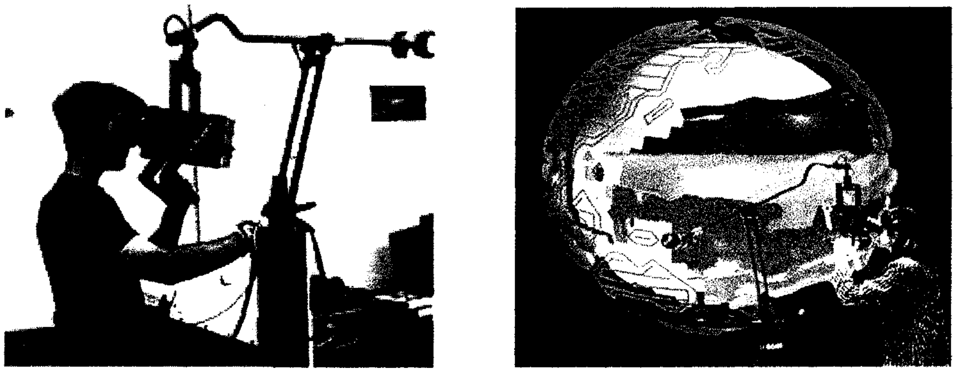


Figure 2.4: a) Boom stereo display device; **b)** Boom--Binocular Omni-Orientation Monitor

2.1.2. Input Devices

Input device as an information transporter between physical world and VR system device is a must to perform high quality simulation tasks. Common VR input devices include 3D mouse, track-pen, data-gloves, keyboard, video, camera and 3D scanner.

- **Data-gloves**

  Data-gloves can capture and convert real hand information into digital signals, thereby performing all kinds of simulation associated with the hand motion. For instance, users can build the virtual hand modeling in computer by certain algorithm. Data-gloves companies also provide relevant software to help users achieve real hand motion simulation.

  In our system, we use 5DT data-gloves (Figure 2.5) as input device to capture the motion and gesture information of the index finger or the tip of the index finger. And, in terms of conversion and program process, we use collected information as input data to fulfill our drawing purposes.

Figure 2.5: 5DT data-gloves also used in our experiments

- **Motion trackers**

  Motion trackers are 3D tracking devices used to capture the motion of real objects. They can provide position and orientation tracking data based on electromagnetic principle. In our system, 6-DOF trackers (from Flock Birds) are used to capture and transfer the position and rotation information of the real hand to assist the data-gloves to perform graphic design task in VR environment.

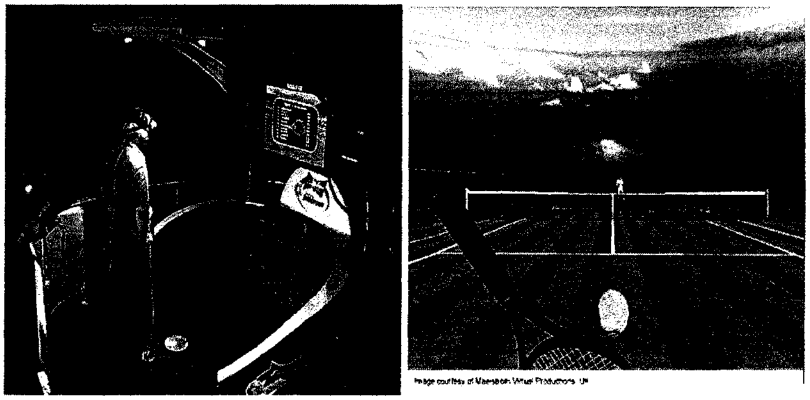


Image courtesy of Maelstrom Virtual Productions, UK

Figure 2.6: Flock of Birds 6-DOF tracker and its applications form Ascension technology

2.2. Virtual reality and Design

2.2.1. Traditional design analysis & checking method

A typical product design process consists of several steps: 1. Conceptual design. 2. Detail design. 3. Analysis and Verification; 4. Modification and Approvals. Prototyping, simulation and optimization are some of the frequently used tools to generate and test ideas and to select the best design that satisfies the requirements at the highest level. Designers interact with the physical prototype through touching, observing, manipulating

Designers interact with the physical prototype through touching, observing, manipulating and testing, with which they can intuitively evaluate and compare their design ideas with real output results. However, physical prototyping is extremely time-consuming and expensive. Hence, the time and resource limitations prevent all the design ideas from being tested in physical prototypes. Furthermore, in today design philosophy, the physical prototyping is one undesired stage of product design and companies omit this option by adapting more advanced virtual prototyping techniques.

2.2.2. Digital designs CAD

In recent years due to the reason stated above, virtual prototypes are preferred as the replacements of old physical prototypes to perform concerned simulation and test tasks to provide intuitive support for the design process in Computer Aided Design (CAD). Virtual prototyping provides various advantages such as quick reproduction, easy modification and low cost, etc. However, in current CAD systems the generation of virtual models should be realized through series processes of 2D to 3D conversion. As a result, in CAD system, the generation of the 3D virtual model comes after the detail design process. Hence, the evaluation and redesign of products unavoidably lag behind processes (2D). In this sense, such a design sequence is not different from the traditional design path except using computer and mouse rather than pens and papers. CAD, therefore, cannot provide sufficient support for the generation of virtual 3D model in the early design stages and lacks sufficient data link support between conceptual design and detail design.

2.2.3. Application of VR in design

In recent years, VR applications and studies have shown great potential in many areas such as education, architecture, training, and medicine, etc. In design field, VR has combined with current CAD technology to support products' 3D evaluation/test and animation tasks to provide designers and customers with intuitive real-time simulation results. However, the products' evolutions, modification and re-design work still represent a redundant process. Such operations can only be done using 3D prototypes. However, prototypes can only be generated after the conceptual design on 2D space, possibly sketches on a paper is completed. Hence there are significant amount of works being done to convert initial design ideas from 2D to 3D to test ideas and verify their fitness to the objectives. In this thesis, we show that the recent advancements on both VR hardware and software enable us to generate 3D virtual models of initial design ideas in conceptual design stage. The resulting product provides 3D parametric models of design ideas include:

- Virtual prototypes
- Surface thicknesses
- Assembly operations: Union, Intersect, Add or Subtract $(\cup, \cap, +, -)$.

Below we summarized the related works in the area of free-form sketching and conceptual design.

2.3. Literature review

Many conceptual design works have been done by the pioneers in the field. In general, literature in the field can be categorized in two distinct groups: i) Spline Based methods; ii) Surface based methods. The first group works on generating sketches in the computer environment by using surface patches which are based on parametric or non-parametric spline patterns. The second group of research work in this area focuses on generating initial sketches of design using surface patches. In addition to these two groups, there are recent works that combine surface patches and spline generation techniques to better serve the objectives of the conceptual design process. Below, we discuss the noteworthy works in these two classes of design techniques and provide comparative insights about their practicality and usefulness in design.

Wesche, G. Droske, M. (2000) developed a spline based free-from sketching toolkit to perform conceptual design on a virtual environment. The developed toolkit is known as responsive workbench. They used a stylus as an input device to draw B-spline curves in VR directly, then edit the curves to generate a curve network which can be used to generate surface model by filling the multisided path with closed boundary surrounding it. Their method provides a well defined editing function. Moreover, using G^ncontinuity along the patch boundaries through the surface modeling, the responsive workbench provides smooth finishing as shown in Figure 2.7. However, there is a disadvantage in their surface modeling in that the inside surface cannot be modified due to the coon surface properties. Also, there is no mention of depth generation which is an essential part of CAD modeling for defining material properties and to conduct various engineering and dynamic analysis including finite element analysis, etc. Also, their editing functionality does not include subtraction operation which is again very important to generate complex geometries.

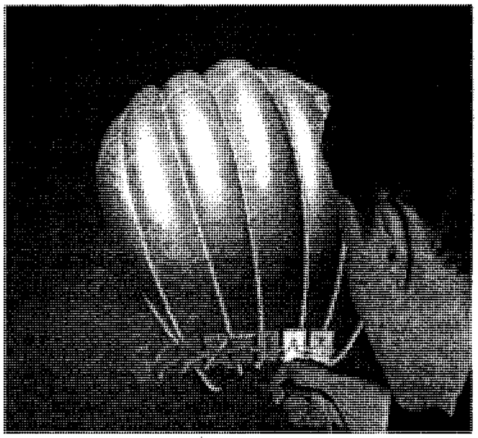


Figure 2.7: 3D styling of surfaces on the Responsive Workbench(RWB) TM(Wesche, and Droske,2000)

A Different method was introduced by Liu, X.C. and Gadh, R.. In this technique, without using parametric curves and surface representations they directly go to conceptual design in VR environment. They chose to fully utilize VR support to construct a skeleton of the designed object. With the usage of CAD technology such as cover, loft, skin, sweep and stitch, a 3D object can be created with related solid representation or volume representation (Figure 2.8). This is a big advantage from which they can provide thickness information for designed object. However, the method is not far different from traditional CAD systems where limited flexibility is provided. In other words, images can only be generated using the library of available shapes. This may be useful in detail design, however when it comes to the conceptual design where capturing the designers feelings and ideas is far more important than details of the image, the proposed method shows limitations.

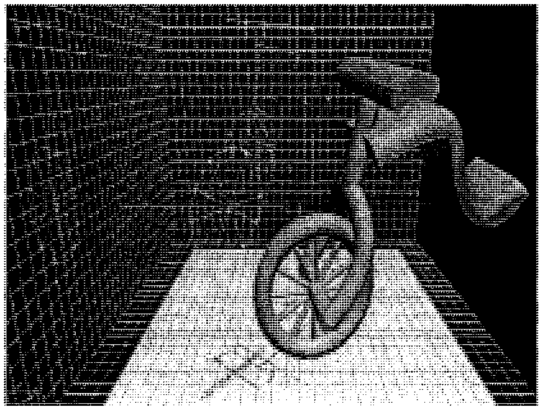


Figure 2.8: Skeleton Based Representation and the Model(Liu and Gadh unpublished)

Virtual clay modeling method was presented by Krause, F-L, et al. (2004). They divided the creation process into 3 steps. First, using a pointer as input device, similar to the method used in [1], they create curve networks by editing B-spline curves with a pointer and fitting the curve network to the desired parametric surface. Second, they introduced the clay modeling method to achieve the intuitive expression of the design ideas. Finally, they converted their geometries into volume representation (Figure 2.9). This method, however, is not adequate in location deformation, and interactive aspects.

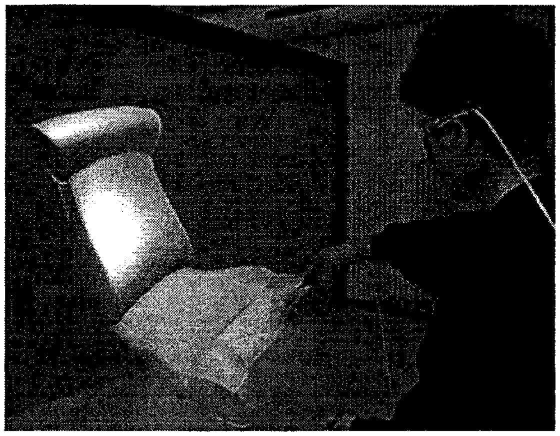


Figure 2.9: Editing the surface(Krause, F-L, et al., 2004)

In their paper, they also suggested that 3D image of an existing real object can be captured from the physical prototype (reengineering) using a pointer to track the object's surface information to get relevant data points instead of using expensive 3D scanners or other expensive equipment. The reengineering of existing objects is a common practice and studied extensively in the literature. Zetu and Akgunduz (2005), Corbo et. al (2004), and Pollefeys et al. (2000) are some of the examples where images of existing objects are reconstructed from the sequence of camera images.

Wesche,G. and Seidel, H-P (2001) proposed a tracking stylus to capture input data. The data can be used to create a curve networks to descript the skeleton of the designed model. The major drawback of this method is not supporting the surface generation. The designed shapes are only in the form of wireframes (Figure 2.10) which are not suitable for engineering analysis.

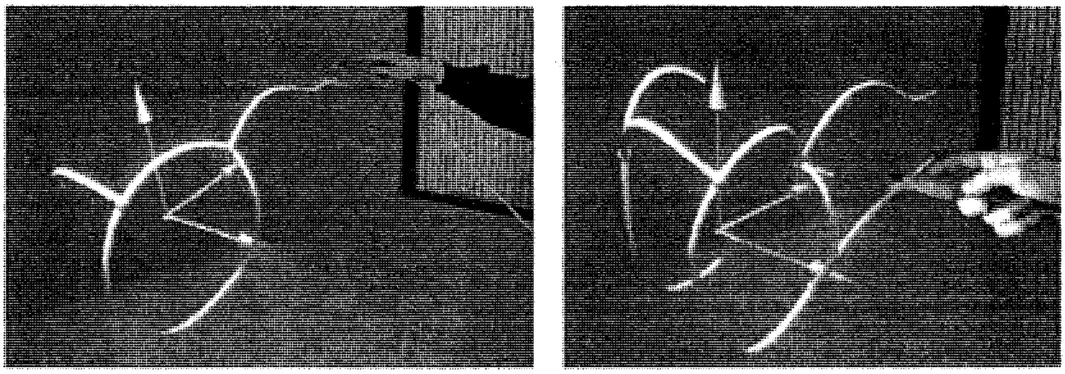


Figure 2.10: Tracking Stylus for Wireframe modeling in 3D(Wesche and Seidel, 2001)

Data-glove as input device was used by Xu, S., Fok, S, C. Tor, S. B. (2000) to construct curves based on the collection of the data points. However, their suggestion also does not support surface generation.

Using hands to draw images in 3D, obviously, is the closest technique to the real and natural expression of thoughts and feelings of the designer. In Schkolne (2006), surface patches are directly placed in 3D using tracked hands. In his work a 6 degrees of freedom tracker, data-gloves and stereoscopic glasses are used. A polygon surface-based image was demonstrated which perfectly fitted the trajectory of the hand, providing users with intuitive feelings (Figure 2.11).

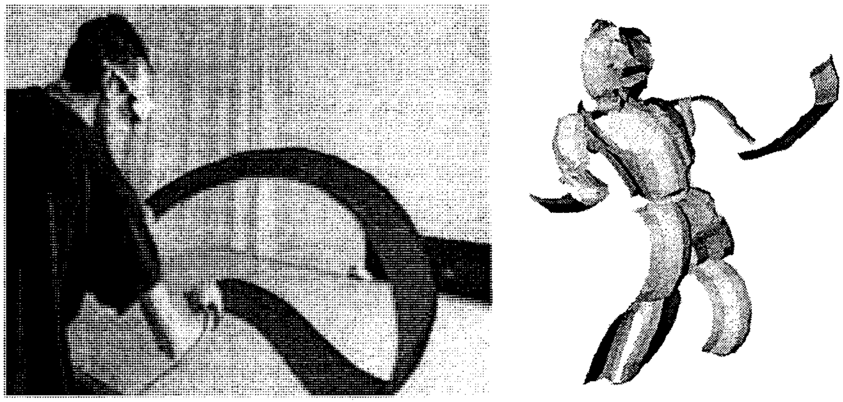


Figure 2.11: Free-form sketching (Schkolne 2006)

The path of the hand in this figure is rendered as a geometric object and the curvature of the hand defines the stroke/movements. A large amount of data is specified in each sweep of the hand. This work also did not mention the possibility of going to parametric surfaces. Moreover, thickness of the surfaces and Boolean operations are not supported.

Yu and Akgunduz (2005, 2004) introduced a three-step approach to perform the free conceptual design in 3D. In this method, first approximate shape of the designed images is drawn in 3D using data-glows. Next, various reference points are selected on the surface of the initial design and selected reference points are triangulated using parametric surfaces. Finally, various modifications are enabled to improve the parametric surfaces using the initial design as the benchmark.

Figure 2.12: 3-Step Free-form Sketching technique (Yu and Akgunduz 2004, 2005)

One of the important aspects in free-form sketching in 3D is fitness to the parametric modeling. Continuity between different patches may become cumbersome, as the project gets larger. Etienne Beeker (1986) investigated the feasibility of G^n continuity for connecting free-from surfaces. He used normal vector theory to guarantee the smoothness from patch to patch by forcing the direction of the normal vector along the boundary line to verify continuity.

Barian A. Barsky and Tony D. DeRose (1990), mentioned to use Beta constraints obtained by applying equivalent parameterization or reparameterization methods to the original parametric representation of the curve to verify G^n continuity for the two joining parametric curves at the common joints. He proposed if G^n continuity holds between these two curves, then Beta constraints (called shape parameters) exist between them, which not only can be used to verify the G^n continuity but also can be used in a CAD environment to modify the shape of the curve without causing the change of the control polygon.

Jana Pilnikova et al. (1998) mentioned the method to stitch the triangle Bezier patches with geometric continuity by applying the polar forms on the parametric continuity. In terms of their method, if the control points for both adjacent Bezier patches are coplanar, then the corresponding Bezier parametric patches will enjoy the geometric continuity features. Using their method, they successfully connected the two adjacent Bezier patches along their common boundary and obtained C^1 andC^2continuity.

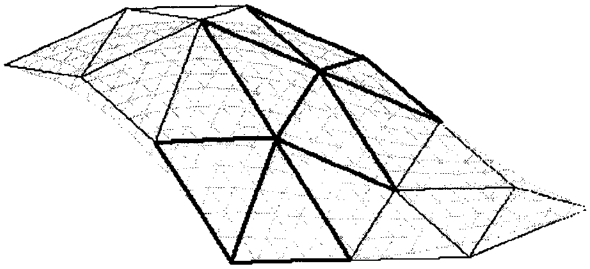


Figure2.13: Geometric interpretation of C^1 continuity (Jana Pilnikova et al., 1998)

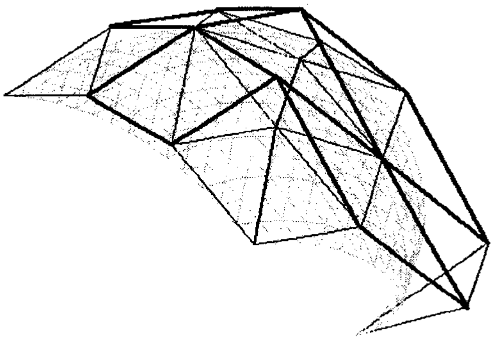


Figure 2.14: Geometric interpretation of C^2continuity (Jana Pilnikova et al., 1998)

Hongwei Lin and Hujun Bao (2007) proposed combining reverse engineering technology to construct the smooth surface. According to their method, first, a triangle mesh for the object is constructed from the point cloud by reverse engineering method; then, this triangle mesh is separated into a set of quadrilateral patches, and normal vector curves will be constructed along the boundaries where form as a normal curve net so that the G^1smooth conditions can be guaranteed along the patches' boundaries ; next, re-fitting the patches to make it satisfy the pre-determined tolerance and G^1 continuity conditions defined by boundary normal curve net, in this method, they accomplish the reconstruction for a smooth surface over a triangle mesh model.

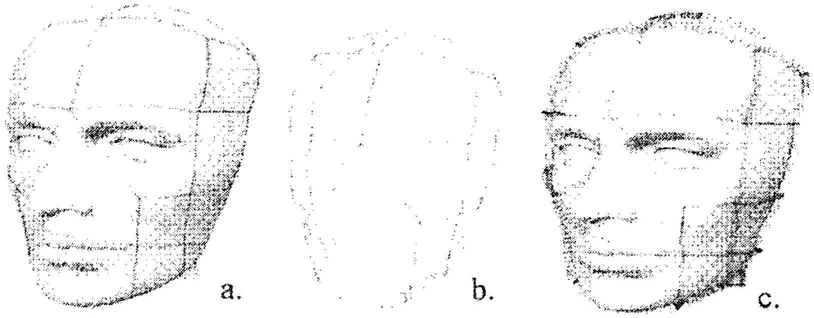


Figure2.15: a). A triangular mesh is segmented into a series of quadrilateral patches; **b).** Quadrangle curve mesh formed by the boundaries of the quadrilateral patches; **c).** Normal curves, like a fence, enclose and separate each quadrilateral patch. (Hongwei Lin and Hujun Bao, 2007)

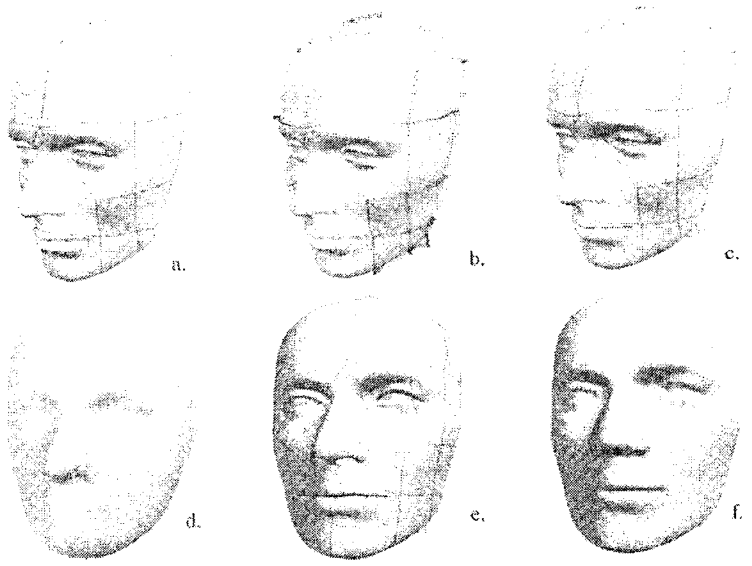


Figure2.16: a) Quadrilateral patches; **b)** normal curve mesh; **c).** boundary fitting curve mesh; **d)** initial G^1 continuously stitched Bezier patches fitting the quadrilateral patches; **e)** complete curve mesh after adaptive fitting; **f)** the final adaptive fitting result with precision 2.1×10^{4}. (Hongwei Lin and Hujun Bao, 2007)

The rest of this thesis is presented in the sequence as follows: chapter 3 3.1 Image Capturing: Hardware and Software, 3.2 Real-time parametric surface modeling, 3.3 Mathematical expression; chapter 4 Advance problems in free-from sketching and examples for displaying our method; chapter 5 Conclusions and future works; the last chapter will be bibliography.

Chapter 3

Free Form Conceptual Design: Theory and Discussion

This thesis introduces a 3-dimensional free-form sketching tool to be used mainly in conceptual design. The proposed model is designed and tested in an immersive virtual environment. Users of the designed system, designers or artists, are provided with a virtual environment in which they are able to draw 3D images of a product, directly into the computer. Immersive VR enables interaction between the users of the system and the simulated environment through data-gloves by which the location and the orientation of the hands are captured. In addition, the designed sketching tool enables users to use other traditional computer hardware such as keyboard and the mouse. Hence the designers are able to sketch their ideas in 3D using data-gloves. Various support functionalities are available in the system. Support commands can be handled via mouse or keyboard inputs. Limited number of hand gestures is also recognized as input commands in the system to enable further flexibility and the reality for the user.

Advantages of the sketching tool described in thesis are:

- Immersive Environments replace the paper-pen based traditional sketching.
- Using hand is the most natural tool that a designer has. The proposed toolkit enables users to sketch their thoughts using data-gloves.
- Images are transformed into parametric surface equations, hence storing the data files; modifying the image and using the results with other CAD software are very easy.

- Our system enables the modeling of surface thicknesses which is one of the first attempts in this filed. Parametric surfaces with surface thicknesses give better realisms to the images and enable more accurate engineering analysis such as finite element analysis.
- Finally the proposed free-form sketching technique allows computational geometry operations: Union, Intersect, Add and Subtract operations.

The details of the developed system are discussed below.

3.1. Image Capturing: Hardware and Software

The proposed free-forms sketching tool is designed using Open GL library and C++ programming language. Meanwhile, 5DT data-gloves are chosen as input device and command control device attached by Flock birds which provide six degrees of freedom motion tracking. Now, let us introduce the hardware and software used in the toolkit.

3.1.1. Capturing images from hand gestures: Data-gloves

The main component of our technique is the data-gloves which enable us to track the hand gestures precisely. The hardware first records the motion information of the hands and converts into position information and then transmits it into the computer in the form of digital signals. The exact locations of the hands, including the fingers are calculated from these signals (See figure 3.1 for illustration).The given hand image was created in our lab for various other VR applications.

Figure 3.1: Virtual hand- Replication of user's hands in VR

By using the combinations of the different gestures of the left hand fingers, a mixture of commands are performed under the control of the left hand, which makes designers to work completely with their hands without using keyboard and mouse during the drawing process. The command gestures are recognized by the application through simple geometric checks. For instance, bending a finger larger than a threshold implies that a segment is completed. Similarly, a new segment can be started by the same finger motion. On the other hand, the right hand is used to draw the desired shape.

Since creating a shape in an empty space is tedious work, usually the initial image is not satisfactory for the product design. Rather, it only qualifies as a benchmark image that includes the fine-details (ideas) of its creator. The final image is built through patches of parametric surfaces. The overall idea is to generate such parametric surfaces that fit the initial sketches. Guarded by this idea, we choose reverse engineering method (apply least-square method into our application) to achieve our objective.

3.1.2. Data Format: Processing the collected data for image generation

In our examples, the surface patches are represented as 4*4 matrixes for control points array in the reverse engineering method in order to obtain the cubic spline surfaces, i.e., we choose the smallest size patches to build a whole image, which has the following advantages:

1. Easy to modify or change in a small range;
2. Reduce the calculation burden for CPU;
3. Save the limitation memory space;
4. Easy to display the different drawing methods such as single-point drawing and non-single-point drawing. (The types of drawing styles will be discussed later in the thesis in details)

Hence, we use C++ programming language and openGL as the working platform in our system to build our modeling by using least square method and NURB parametric surface equations so that we enjoy the flexibility in expressing the shape of objectives to dramatically reduce the error between the generated image and the desired one. Now, we introduce the mathematical approach below.

3.2. Real-time parametric surface modeling

Real-time image generation is an essential need for our system to enable designers a naturalistic work environment. Designers would like to see the outcome of his/her inputs as part of the designed object in real-time. Just like it would happen if he/she is using the pen/paper based sketching technique. However, in our case, the final image is not the one the designer is inputting the computer monitor through his/her hand but, it is the result of the post-processing algorithm imbedded into our system which converts the user inputs into a parametric image. The main connection between the user inputs and the parametric representation is that the parametric image is the best fitting surface patch to the 3D data-points collected from user's strokes.

3.2.1. Generation of input data

The ultimate objective of the user of the system is to deliver his/her thoughts and imaginations to the interested parties in the best effective way. We considered the best form to represent images of design and use this for further analysis in parametric surface forms. Parametric surfaces are constructed from various reference points. This characteristic of parametric modeling enables us to model a free-form sketching system that produces an intuitive and flexible working condition for the designer. It is intuitive because a designer would be able to mimic the sketching method that his/her would follow in the paper-pen based traditional method. A designer using a paper-pen based sketching platform would use several lines of stroke to shape the image in his/her mind into the paper. A sculptor would pull, push, stretch, add, remove material, etc to shape the design from clay. In the proposed 3D sketching tool, the user can act in the same way. Hand movements would generate line segments. Changing the location of control points or data points would reshape the existing image. The user would see the changes in real-time.

Below, the overall architecture of the proposed sketching toolkit is provided in figure 3.2 and discussed in details.

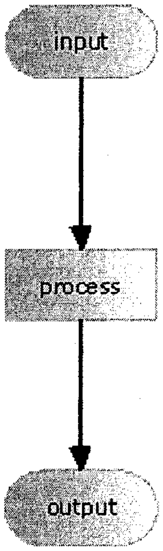


Figure 3.2: Overall architecture of the proposed sketching toolkit

- **Resources for the input data**

In our system, the input data is acquired in two distinct manners.

1. We use data-gloves and flock-of-birds position trackers to capture and record the motion of the tip of the index finger of the right hand, and transfer and convert it as the input data. Since the captured reference points are from single point generating source, the tip of a finger, the drafting style is called "single-point-generation". And, data acquired by this method is called "single-point-generation data".
2. We use the same equipments and skills shown above to track and record the motion of the joints of the index finger of the right hand, and transfer and covert it as another input data. Since the source of input generates multiple reference

points, each joint corresponding to a reference point, the drafting style is called in our system "multi-point-generation". And, data acquired by this method is called "multi-point-generation data".

After capturing, transferring and converting these data information from the physical space to computer, we arrange these data to make them satisfying by the demand of mathematical expressions.

The image shown in Figure 3.3 is the representation of right hand in VR. Total of 20 data points, shown in small green color points, are possible to generate using a single image. Hence it is possible to generate non-linear geometries in 3D in a single pass. They are the source of the data. To minimize the computational complexity and to ensure the practicality of the system, only the index finger joints are used as the source in our system

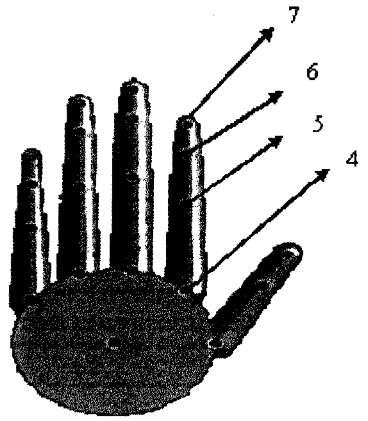


Figure 3.3: The hand image used to replicate human hand in VR. Indexes 4, 5, 6, 7 are index numbers of the joints of the index finger of the right hand

- **Independent motion control**

Motion control is one of the most important issues to be handled in user-VE interaction. In order to provide the immersion, in our case a natural working environment for the designer, relative positions of user – includes location, view direction and the field-of-view – and the designed object should be controlled in real-time. The control occurs in many forms. First, simple visual contact with the VE should be sustained with a certain level of realism. Next, the user should be able to interact with the VE. In our case, the term VE concerns the images formed in 3D space.

The above objective can be sustained by the free control of the hand. Furthermore, enabling the hand to interact with the designed object, without losing the geometrical properties of the design, should also be sustained. Objects should be:

i) Treated as a single object which can be moved and rotated with respect to the hand independently;

ii) Sub-components, in our case the surface patches, should be handled independently. It is worth to note that these free from motion and rotation for hand and designed objects, the data for designed surface and data adhered on the joints of the hands shouldn't interfere with each other during this process.

In order to make the data adhered on the hand's joints not follow the rotation motion of the designed objects; we multiply with the R_{GL}^{-1} with the hand data $\mathbf{H}$ to achieve the rotation motion for the designed object independently from the hand. The hand data ($\mathbf{H}$) includes coordinates of the joints in desired coordinate system (may change between world coordinate system and object coordinate system). Additionally, for the sake of keeping the total scene relatively static with respect to the global coordinates system, inverse rotation matrix of the designed objects should be used efficiently to the total scenes e.g. coordinates system frame in our case. Furthermore, if we apply the motion control (including rotation and translate) to the object, we need to apply its reverse matrix (including rotation and translate) to other objects which enable these objects' motion control independent with respect to that of the object applied motion control. Now, let us introduce the specific method below.

.

A general transformation of an object that includes rotation and translation in all three axis, T, is:

$T = LR$ or $T = RL$ 3.1

Where L be the translation matrix and R be the rotation matrix in X, Y, and Z axis respectively.

Let R be defined as:

$R = R_x R_y R_z$ 3.2

And, an independent motion of hand can be achieved by

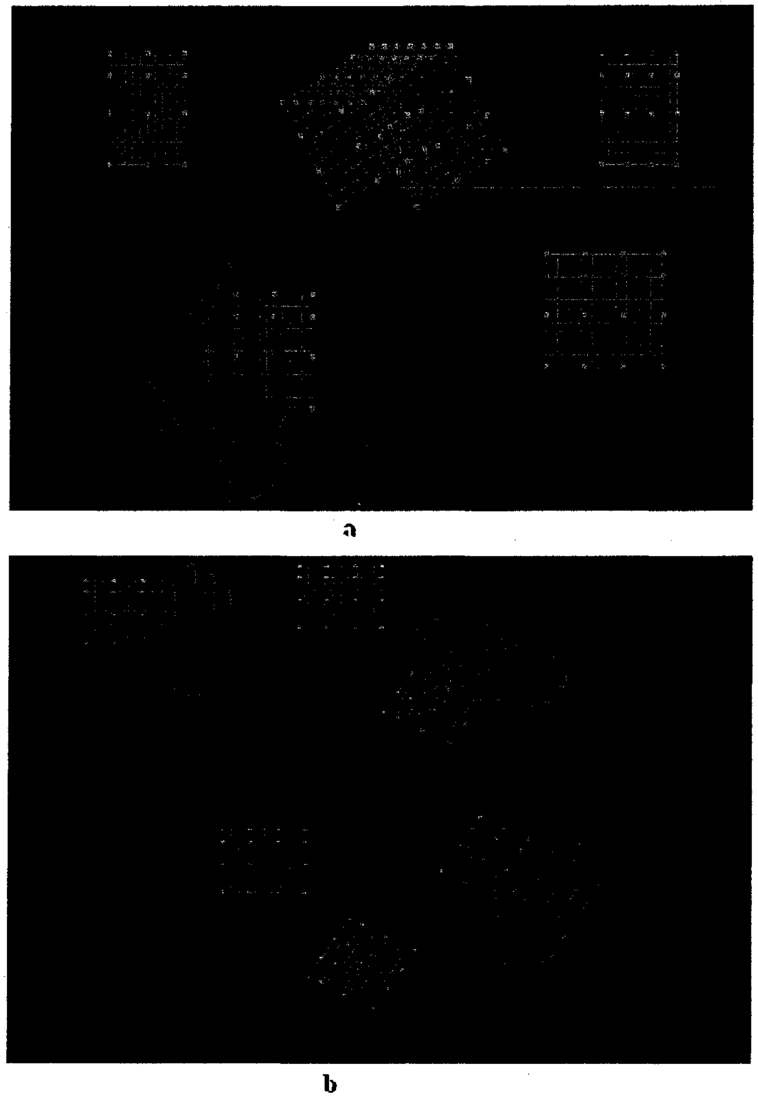
$V' = T^{-1} V$ 3.3

Where

$T^{-1} = R^{-1} L^{-1}$ or $T^{-1} = L^{-1} R^{-1}$ 3.4

After multiplying this reverse matrix (Eq.3.4) with virtual hand, as well as the adhered data H on the hand the independent motion control for the hand and the designed objects as well as the reference point's data are obtained. It worth to note that equations (3.1, 3.2, 3.3, 3.4) we give are the simplest expressions, in fact, the motion control for the designed objects is the free combination of the rotation and translation motions.

Now, we use the following images (figures 3.4 a. and b) to demonstrate the corresponding results generated by using the above mathematic operation.



a

b

Figure 3.4: Display the independent motion relation between designed objects and hand.

- **Processing the acquired data**

After separating the motion relations and achieving the free control purpose for the hand and the designed objects, we begin processing the acquired data. First, we named *u* direction along the local *X* axis [see process 1 and process 2 in figure 3.5], *v* direction along the local Y axis [see process 1 and process 2 in figure 3.5], and chose the right hand index finger as drawing tool , then edit the acquired data matrix in terms of the inputted sequence of data points and index number of the right hand index finger joints (not enjoyed by single point drawing method) as well as data input methods(single-point-generation and multi-point-generation). In this method, we obtain a 4*4 data point's matrix which completely responses to the motion and gestures of the joints of the index finger of the right hand or that of the tip of index finger of the right hand.

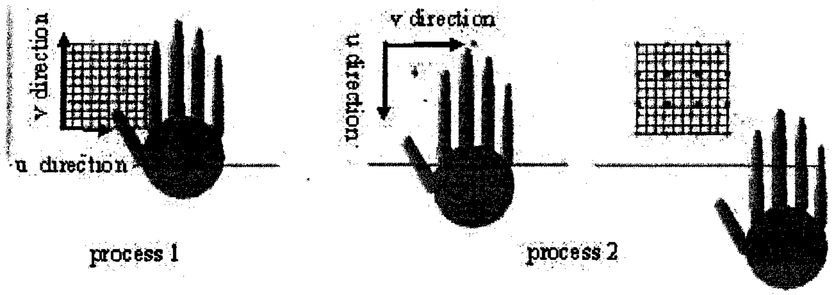


Figure 3.5: Process 1 is the multi-point-generation and Process 2 is the single-point-generation

Let us introduce the specific editing methods and processes below:

In our system, as mentioned above, we classify the data input methods as two categories:

- Single-point-generation data.
- Multi-point-generation data.

Hence, we use two ways to edit the acquired data as a 4× 4 matrix.

First, for single-point-generation data, we collect the data following the drawing sequence shown on figure 3.6 below:

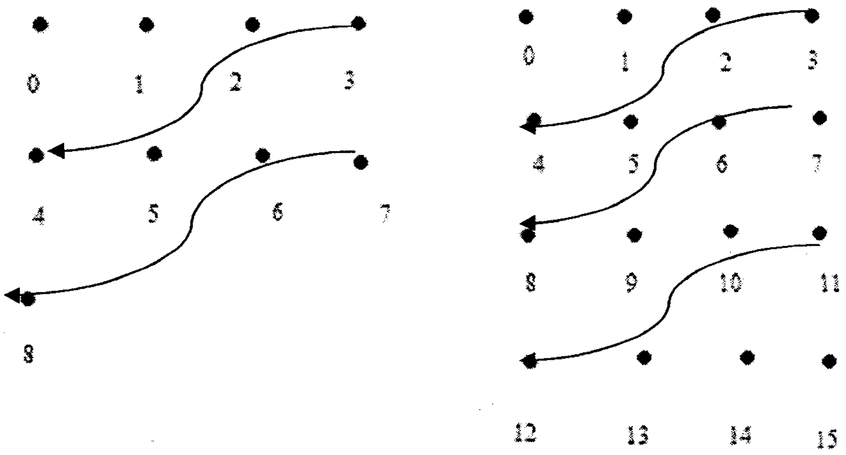


Figure 3.6: Data drawing sequence for single-point-generation

Editing the acquired data matrix is demonstrated in the figure 3.7:

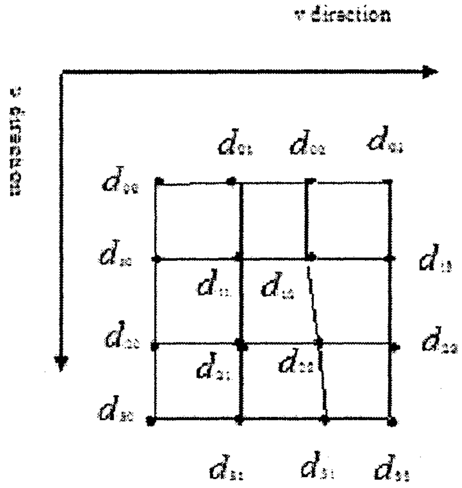


Figure 3.7: Edited acquired data matrix for single-point-generation

Second, for multi-point-generation data, we draw the data below and follow the sequence as the arrow pointing at. Drawing sequence

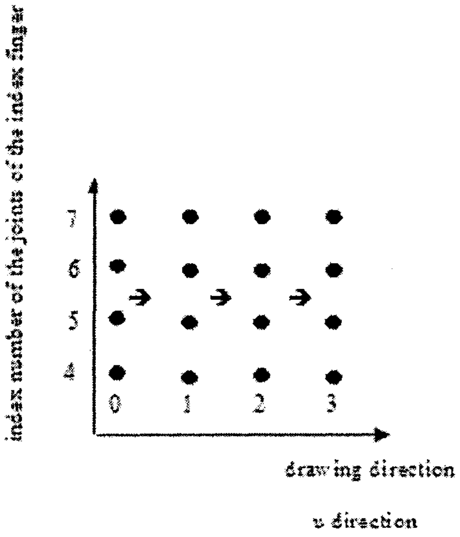


Figure 3.8: Data drawing sequence for multi-point-generation

Editing the acquired data matrix is shown in figure 3.9.

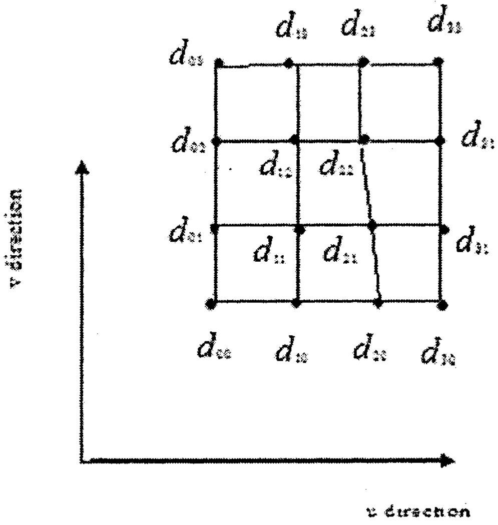


Figure 3.9: Edited acquired data matrix for multi-point-generation

According to the methods mentioned above, we obtain the acquired data matrix. Below, we present the specific mathematical process by means of its mathematical equations.

3.3. Mathematical expressions

In this section, we use mathematical expressions to present our methods which are used in our system for the free conceptual design purpose. Our presentation sequence is as follows:

1. Reverse-engineering method (least-square method) for the parametric surface's generation
2. Smooth connection method
3. Thickness problem
4. Discussion on the trimming methods

3.3.1. Least-square method

As mentioned above, in our system we have two ways to collect input data: "single-point-generation data"; and "multi-point-generation data". Besides the data acquiring techniques, the required mathematics to generate the surface patches including the least-square method are the same. Now, let us address them.

Obtaining the acquired data matrix, in the proposed method above, the initial inputs to the designed object are the data points. Let **d** be the data points retrieved from the motion of a virtual had shown in figure 3.4 and 3.5, and *n* be the number of data points in the data set **d**. An object, O, is the parametric surface constructed from control points **C**. Control points are calculated from collected data points **d**.

The overall architecture of the discussed technique is given in the below flow chart, figure 3.10:

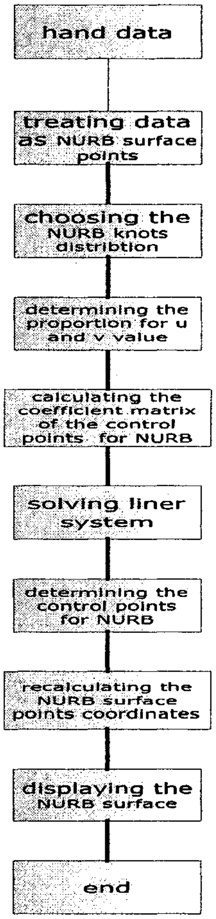

hand data
treating data as NURB surface points
choosing the NURB knots distribtion
determining the proportion for u and v value
calculating the coefficient matrix of the control points for NURB
solving liner system
determining the control points for NURB
recalculating the NURB surface points coordinates
displaying the NURB surface
end

Figure 3.10: Architecture of the Proposed Surface Building Algorithm

Given: a set of data points $\Omega = \{ \boldsymbol{d_{mn}} \}$ where $m = \{0,1,\cdots,m\}$ and $n = \{0,1,\cdots,n\}$ suppose that we have a set of control points C_{mn} where $m = \{0,1,\cdots,m\}$ and

$n = \{0,1,\cdots,n\}$. Let the number of order for the NURB function to be equal to the number of control points, and let the number of control points to be equal to that of data points, then the NURB surface equation in its general form is given as:

$$P(u,v) = \left[R_0^{m+1}(u)R_0^{n+1}(v) \quad \cdots \quad R_m^{m+1}(u)R_n^{n+1}(v)\right]\begin{bmatrix} C_{00} \\ \vdots \\ C_{mn} \end{bmatrix} \qquad 3.5$$

$$0 \le \mathrm{u} \le 1; 0 \le \mathrm{v} \le 1;$$

Where,

$R_{ik}(u)$ is the rational B-Spline basis functions:

$$R_{i,k}(u) = \frac{W_i N_{i,k}(u)}{\sum_{i=0}^{n} W_i N_{i,k}(u)} \qquad 3.6$$

w_i represents the weight for each control point.

$N_{i,k}(u)$ is the B-spline functions which can display as follows:

$$N_{i,k}(u) = (u - u_i) * \frac{N_{i,k-1}(u)}{N_{i+k-1} - u_i} + (u_{i+k} - u) * \frac{N_{i+1,k-1}(u)}{u_{i+k} - u_{i+1}} \qquad 3.7$$

Where k is the order of the B-spline curve and u_i is called knot vectors or knot value which depend on if the B-spline curve is open curve or closed curve. Equation 3.7 is the parametric equations for the open B-spline curve.

In our case, we arrange the knot values as follows:

$$\mathrm{KV} = \left[u_0 u_1 \cdots u_v\right]^{\mathrm{T}} = \left[u_0 \cdots u_0 u_k \cdots u_{m-k} \cdots u_v\right]^{\mathrm{T}} = \left[\underbrace{0 \cdots 0}_{k} u_k \cdots u_{m-k} \underbrace{1 \cdots 1}_{k}\right]^{\mathrm{T}} \qquad 3.8$$

Where k is the coefficient determines the degree of the NURBS curve. It keeps the NUBRS curve to interpolate the end points. By choosing k (order of the NURBS curve) we arrange the full multiplicity (order-1).

m = n + k + 1 n is the number of control points -1; k is the order of B-spline curve and m is the length of knot vector.

We arrange the knots to be spaced averagely with an increment $u = \dfrac{1}{m-2k+1}$. We suppose that the data points collected from the data gloves and sensors to be on the surface of the designed object, thus we have the equation as follows:

$$\begin{bmatrix} d_0 \\ \vdots \\ d_k \end{bmatrix} = \begin{bmatrix} R_0^{m+1}(u_0)R_0^{n+1}(v_0) & \cdots & R_m^{m+1}(u_0)R_n^{n+1}(v_0) \\ \vdots & \ddots & \vdots \\ R_0^{m+1}(u_k)R_0^{n+1}(v_k) & \cdots & R_m^{m+1}(u_k)R_n^{n+1}(v_k) \end{bmatrix} \begin{bmatrix} C_{00} \\ \vdots \\ C_{mn} \end{bmatrix} \qquad 3.9$$

Where the data points set $\{d_{\downarrow}(0,) \quad d_{\downarrow}(1,) \quad \cdots \quad d_{\downarrow}(k)\}$, k = {0, 1, 2...$k$} is the reedit of the data set of Ω $\{d_{m,n}\}$ where $m = \{0,1,\cdots,m\}$ and $n = \{0,1,\cdots,n\}$.

In our case, k =15 for 4*4 matrix, m = {0, 1, 2, 3} and n = {0, 1, 2, 3}.

In the above equation 3.9, each data point d_k has a corresponding pair of parameters (u_k, v_k).The values of u_k, v_k take the proportions of the k_{th} data point on the curves on both u and v directions, and are calculated by using the accumulated length method. In our case n = 3 and m = 3. See figure 3.11 for illustration for the u_k, v_k, we calculate them using the accumulated length method.

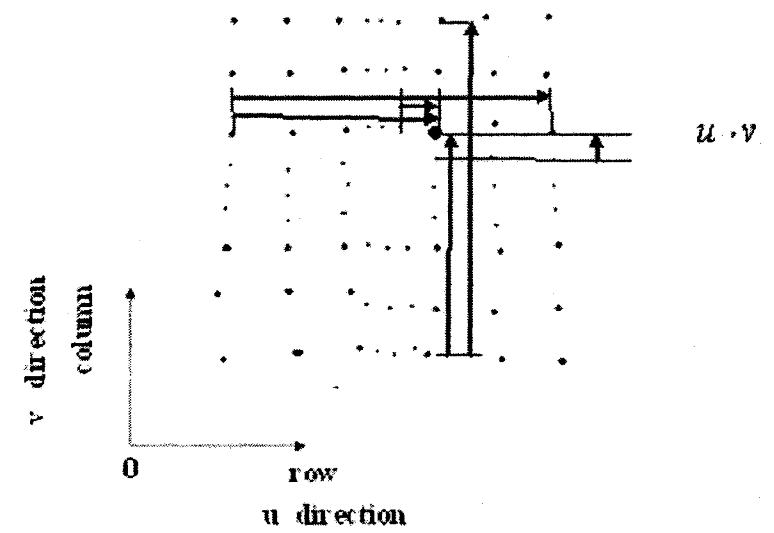


Figure 3.11: Illustration for the calculation of u_k and v_k values

Calculation of u_k and v_k values

i. Along u direction (m rows)

$$u_k = \frac{\sum_{i=0}^{k} L_i}{\sum_{i=0}^{m} L_i} \qquad \text{where } \begin{cases} 0 \le \mathrm{i} \le \mathrm{m} \\ 0 \le u \le 1 \end{cases} \qquad 3.10$$

Along u direction, L_i is obtained as shown below in equation 3.11:

$$L_i = \sqrt{\left(x_{i,j} - x_{i-1,j}\right)^2 + \left(y_{i,j} - y_{i-1,j}\right)^2 + \left(z_{i,j} - z_{i-1,j}\right)^2} \qquad 3.11$$

ii. Along v direction (n column)

$$v_k = \frac{\sum_{j=0}^{k} L_j}{\sum_{j=0}^{n} L_j} \qquad \text{where} \begin{cases} 0 \le \mathrm{j} \le \mathrm{n} \\ 0 \le v \le 1 \end{cases} \tag{3.12}$$

Along v direction, L_j is calculated using the expression given below:

$$L_j = \sqrt{\left(x_{i,j} - x_{i,j-1}\right)^2 + \left(y_{i,j} - y_{i,j-1}\right)^2 + \left(z_{i,j} - z_{i,j-1}\right)^2} \tag{3.13}$$

Now, let us simplify the equation 3.5 in the below simplified form:

$$d = MC \tag{3.14}$$

where

$$M = \begin{bmatrix} R_0^{m+1}(u_0)R_0^{n+1}(v_0) & \cdots & R_m^{m+1}(u_0)R_n^{n+1}(v_0) \\ \vdots & \ddots & \vdots \\ R_0^{m+1}(u_k)R_0^{n+1}(v_k) & \cdots & R_m^{m+1}(u_k)R_n^{n+1}(v_k) \end{bmatrix} \tag{3.15}$$

Multiplying both sides of the equation 3.14 by the transpose of M (M^T) below expression is obtained.

$$M^T d = M^T MC \tag{3.16}$$

Solving linear system given in equation 3.16, we obtain C which is the m*n matrix giving the coordinates of control points in the NURB parametric surface. Once we obtain the C from equation 3.16, we use the equations 3.6, 3.7 and 3.8 to arrange and calculate the knots' parameters to obtain $R_{i,k}(u)$ and $R_{j,k}(v)$, and then by substituting C, $R_{i,k}(u)$ and $R_{j,k}(v)$ into the equation 3.5 we obtain the final coordinate points of the NURB surface. Two pictures shown in figure 3.12 below are two NURB parametric surface images which are generated by using lease-square with different data point input methods and different input data point matrix's editing methods.

In this method, we accomplish the NURB parametric surface drawing. However, In the process of drawing 3D objects, only using one parametric surface to represent a complex object is insufficient and inefficient. Usually more patches are required to accomplish the design task. In our system, we use NURB patch scheme to construct the 3D objects. Therefore, we develop methods below to achieve the smooth transitions with C1 continuity between the stitched parametric patches. See figure 3.12 for the illustration.

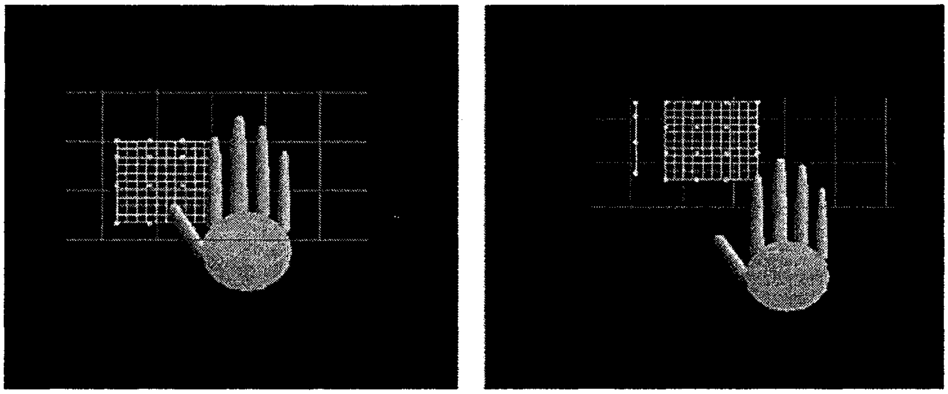


Figure 3.12: Drawing NURB Surfaces generation: **a)** Single point generation; **b)** Multi-point generation

3.3.2. Smooth connection of surface patches: Continuity Rules and exception handling

As mentioned above, only one parametric surface is not sufficient to build 3D objects in VR. Several surface patches are required to cope with our objective. Thus, in this section we introduce the so-called smooth connection methods. The smooth transition between multiple patches in order to sustain the smoothness or continuity is achieved by utilizing NURB parametric surface properties which let the C1 continuity smooth transition between stitched patches. We present the mathematics and its implementation in the following sequence:

- To construct fillet surface to accomplish C1 smooth connection between two stitched NURB patches.
- To build up NURB patches to fill in the holes to achieve the C1 smooth connection along their common boundaries when more than two patches are stitched together surrounding at their common joint point.

i. Construct fillet surface to achieve the C1 smooth connection between two stitched NURB patches

Our method in its general form can be presented below:

Given: stitched NURB *patches 1 and 2*, as shown in figure 3.13

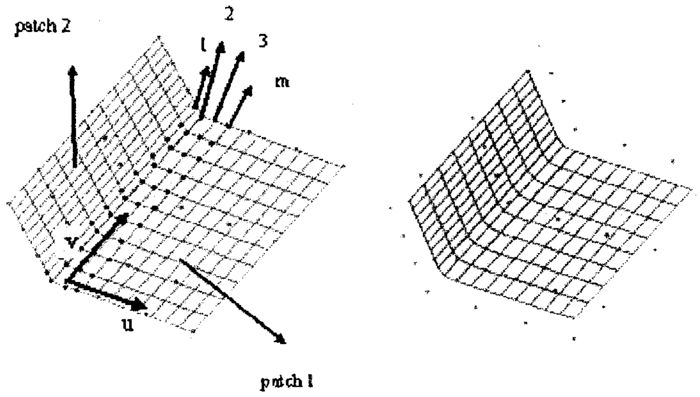


Figure 3.13: Stitching NURB Patches: **a)** Stitching process; **b)** C1 Continuity after stitching

The left image in the figure 3.13 includes dots in red colors which were used as control points in order to build NURB fillet surface between patch 1 and 2. Please note that those points are the local surface data on both of the *patch 1* and *patch 2*.

First, record these data as $\{C_{i,j}\}, i = \{0,1,......m\}, j = \{0,1,2,......n\}$, and then we calculate the data for the fillet surface after denoting the fillet surface points as $p(u,v)$:

$$p(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} C_{i,j} R_{i,k}(u) R_{j,k}(v) \tag{3.17}$$

By using the equations 3.6, 3.7, 3.8, and letting k = 2 along their boundary *v* direction [figure 3.13: a)] as suggested in Zeid (2005) and Farin (2000), we obtain the fillet NURB surface which consists of the following properties:

1. If the full multiplicity properties can be held by the end knots, the control polygons of the parametric surface boundary curves will be the boundary polygons of the control polygon mesh. (Guaranteed by equation 3.8)
2. When the order *k* of the B-spline is equal to 2, the parametric mesh will be its control polygon mesh itself;
3. The boundary polygons of the parametric mesh pass through the four corner points of the control polygon mesh and tangent to the boundary polygon of control polygon at these four points if the end knots have full multiplicity.

These three properties guarantee to maintain the C1 continuity (first order derivates are the same at their common joints) for fillet NURB surface and *patch 1*, and the fillet NURB surface and the *patch 2* due to:

1. Boundaries being overlapped along the *v* direction;

2. At the four end points of the control polygon, the boundary conditions for C1 continuity being held.

Therefore, we conclude that this new fillet NURB surface accomplishes C1 smooth connection task and make the C1 smooth transition in a local range (two adjacent surface segments) without changing the shape of the designed objects outside this range, as explained below by the special case.

The following, figure 3.14 is a special case used in our system. These three columns of patch surface points shown on the left side are used as the control points. The right image shown in blue region is the new fillet NURB surface defined by the left control Polygon and equations 3.6, 3.7, 3.8and 3.17.

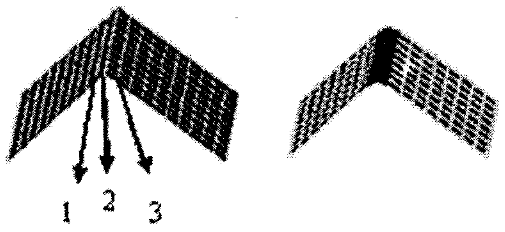


Figure 3.14: Illustration of a special case

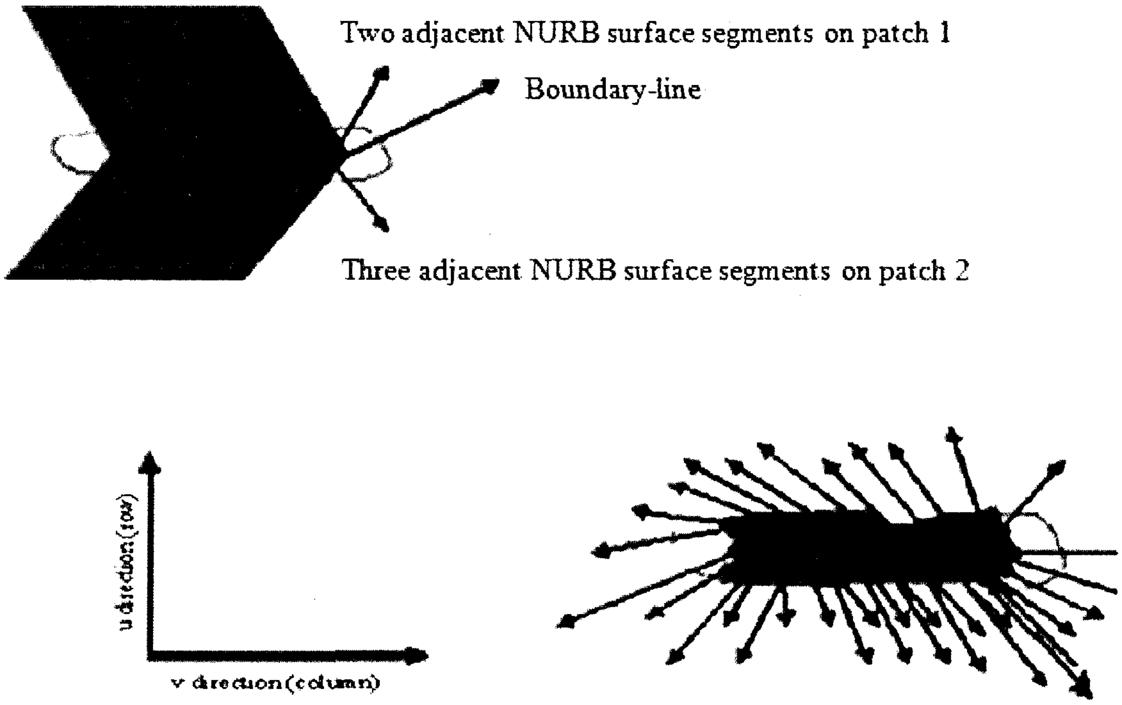
We now address our methods as follows:

In our system, our NURB patches are presented as 4×4 matrixes for control polygon array and 11×11 matrixes for the generated parametric surface points array. In term of the aforementioned three properties of the NURBS surface, we construct fillet parametric surface (NURB) between two stitched patches to make the combination of them (two NURB patches and fillet NURB surface) satisfy the C1 continuity conditions at their common boundaries. As two patches are connected together, they have the overlapped data points along the boundary. We use this column of data [see figure 3.14 arrow 2] and two adjacent columns of data [see figure 3.14 arrow 1, 3] to build acquired control polygon for the fillet NURB surface. It is a 3 × 11 control polygon array arranged by using local NURB surfaces' points. Then, along both directions *u* and *v*, we construct a 3 × 2 order (stitched along *v* direction) or 2 × 3 order (stitched along *u* direction) NURB parametric surface using this control polygon and equations 3.6, 3.7 and 3.8. After generating this NURB parametric surface, we don't show the image for the control polygon but show the fillet NURB parametric surface image instead, thus the construction of the C1 smooth connection is achieved between two stitched patches.

Now, we present the above idea in a mathematic way as discussed below:

According to figure 3.15, from two stitched NURB patches, we use data including one column of boundary data points [noted by number 1 shown on figure 3.15 below], one adjacent column of NURB surface points on *patch 1* [noted by number 2 shown on figure 3.15 below], and one adjacent column of NURB surface points on *patch 2* [noted by number 3 shown on figure 3.15 below] to edit control point matrix **C** as a 3 ×11 matrix below:

$$C = \begin{bmatrix} C[0][0] & \cdots & C[0][10] \\ C[1][0] & \cdots & C[1][10] \\ C[2][0] & \cdots & C[2][10] \end{bmatrix} \tag*{3.18}$$



Two adjacent NURB surface segments on patch 1
Boundary-line
Three adjacent NURB surface segments on patch 2
u direction(row)
v direction(column)

Figure 3.15: Illustration of two stitched patches

Substituting this control point matrix into equation 3.17 and using equations 3.6, 3.7, 3.8 and letting the order as 3 × 2 (order 3 for *u* direction→ row 3 points/row, order 2 for *v* direction→ column 11 points/column), we obtain the fillet NURB parametric surface which achieves the C_1 smooth connection between *patch 1* and *patch 2*.

Let us now explain the method in a more detail fashion below:

Shown in figure 3.15, the two patches are stitched along the *v* direction, thus along *u* direction, we have control point array as 3 rows, and 11 columns along v direction [see figure 3.15]. Due to the *k*=2 along the *v* direction, the parametric surface (NURB) boundaries curves will be the control polygon curves themselves in terms of the properties of NURB parametric surface mentioned above, the new generated fillet NURB surface boundary curves are overlapped with its control polygon boundary curves along *v* direction and due to this control polygon boundary be the NURB patch segment itself, hence, along *v* direction the parametric boundary curve for both fillet and connected NURB patch [suppose *patch 1*] are overlapped and satisfying the C1 continuity conditions [the same for fillet surface and *patch 2*]. Along the *u* direction, in terms of the NURB surface properties, the boundary parametric curves pass by the four corners' end points of the control polygon and tangent to this four end points of the control polygon, therefore, the parametric boundary curves of the fillet NURB surface tangent to its control polygon boundary curves along *u* direction, and due to this four end points not only the end points of the fillet surface's control polygon but also the local points and end points of the stitched NURB patch surface, hence, the boundary parametric curves of the fillet NURB surface tangent to connected NURB patch surfaces 1 and 2.

Hence, reconsidering the mentioned properties, i.e .

1. Along the *v* direction, their boundaries are overlapped together.
2. C1 continuity constitutions hold at four end corner points between fillet surface and *patch 1*, and fillet surface and *patch 2.*

We reach the conclusions that this new fillet NURB surface accomplishes the smooth connection task with C1 continuity and achieve the C1 smooth transition for two stitched NURB parametric surfaces in a local (two adjacent surface segments) range without changing the shape of the designed objects outside this range.

It is worth noting that using this method to construct our C1 smooth connection, that we abandon the first and the last segments of images surrounding the NURB patch (including *u*=0, *u*=1, *v*=0, *v*=1) for each stitched NURB surface [see right image on figure 3.13]. For example, in our system, we show the images of (10×10) NURB parametric surface data instead of showing the images of (11×11) NURB parametric surface data points. Hence, along the stitched boundaries, suppose 3 patches are connected, after C1 smooth connection, there is a hole, existing along the three patches connection boundaries surrounding the common joint point in a segment range. The same problem exists for multi-patches case, for example, 4 patches case or more. Hence, in order to use patch scheme to build complete object with C1 smooth connection conditions, we need to develop methods to fill the holes so that we can obtain a desired image for the designed objects.

ii. Fill in the holes

Using the method presented in section 3.3.2, we construct a NURB fillet surface to achieve C1 smooth connection for stitched patches in a local range. However, due to the discarding of the last and the first segments of the image along the stitched patches boundaries, if stitched patches' number is greater than two supposing 3, 4, 5 or over 5 which is possible in a realistic application, there will be a hole existing among these patches surrounding their common joint.

In what follows, we will present a method to construct a NURB patch to fill in the hole to obtain a complete smooth surface for the designed 3D objects to guarantee the C1 continuity holds along their boundaries.

Our method is in its general form as follows:

Let's denote $\{C_{i,j}\}, i=\{0,1,......m\}, j=\{0,1,2,......n\}$ as control points. They are the points which lie on the stitched NURB patches.

Now, we calculate the data for the NURB surface, we note the NURB surface points, $\{H(u,v)\}$ as:

$$H(u,v)=\sum_{i=0}^{m}\sum_{j=0}^{n}C_{i,j}R_{i,k}(u)R_{j,k}(v) \qquad 3.19$$

where:

The order k in equation 3.19 along the u and v directions should be the same as that of their adjacent fillet NURB surface along their u directions in equation 3.17. By interpolating the control points and using equations 3.19, 3.6, 3.7, and 3.8, we obtain the NURB patches which can be used to fill in the holes. And C1 continuity conditions can be held along their boundaries. Now, take our system as an example.

In our cases, we interpolate the control polygon matrix to be a 3×3 matrix by using the local NURB patches' surface points. Let's denote this 3×3 control polygon matrix in its general form as follows:

$$C = \begin{bmatrix} C[0][0] & C[0][1] & C[0][2] \\ C[1][0] & C[1][1] & C[1][2] \\ C[2][0] & C[2][1] & C[2][2] \end{bmatrix} \qquad 3.20$$

We keep the order $k = 3$ to be equal to that of the fillet NURB surface for the following three cases.

Case 1: As the four patches are connected together surrounding their common joint (figure 3.16), we process it using the method presented above:

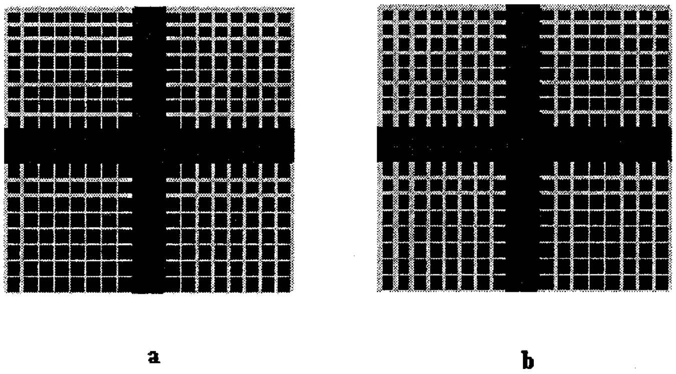


Figure 3.16: a) Image before the hole is filled; **b)** Image after the hole is filled

Along the connected boundaries, four patches intersect at one point (number 5) [shown in figure 3.17 below]. Now, we use the boundaries data and this intersection point data to construct a control point matrix to generate a NURB parametric patch to fill in this hole. We take data from the four boundaries [see figure 3.17 below] and arrange them as a 3×3 control polygon array, then build a 3×3 order NURB surface and let this NURB surface satisfying all above listed properties of the NURB surface by using equations 3.6, 3.7 and 3.8.Hence, this new NURB surface used to fill in the holes can completely satisfy C1 continuity smooth connection condition. We demonstrate its mathematic approach below:

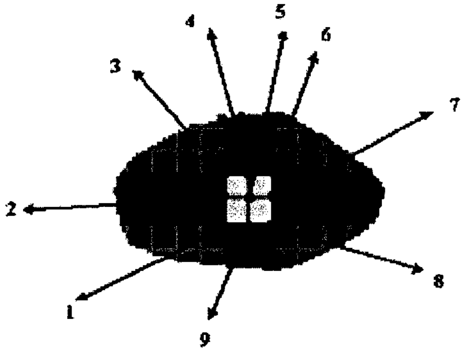


Figure 3.17: Selecting the control points for four patches' connection case

For the hole generated by the four fillet NURB surfaces boundary parametric curves, we utilize 9 points data [shown above on red color dots, these 9 points lie on the four NURB patches surrounding their common joint] as control points to build a NURB surface to fill the hole and make it satisfy the C1 smooth connection conditions . First, we arrange these local NURB patches' surface points as the regular 3×3 control point matrix C (shown in equation 3.20) where the coefficient of the matrix becomes as shown in equation 3.21.

C[0][0]= point coordinates(number 3),C[0][1]= point coordinates (number 4),C[0][2]= point coordinates(number 6)

C[1][0]= point coordinate (number 2),C[1][1]= point coordinates (number 5),C[1][2]= point coordinates(number 7) 3.21

C[2][0]= point coordinates(number 1),C[2][1]= point coordinates (number 9),C[2][2]= point coordinates(number 8)

Substituting this control point matrix into equation 3.19 and using equations 3.6, 3.7, 3.8 and letting the order be 3×3, we obtain the parametric NUBR surface (**H(u, v)**) used to fill in the hole which keep all properties of the NURB surface mentioned above. Hence, we conclude that we fill in the hole by using this NURB patch and achieve the task of C1 smooth transitions between four fillet NURB surfaces and this new NURB patch.

<u>**Case 2:**</u> As the four connected patches to be degenerated as three connected patches

See figure 3.18.

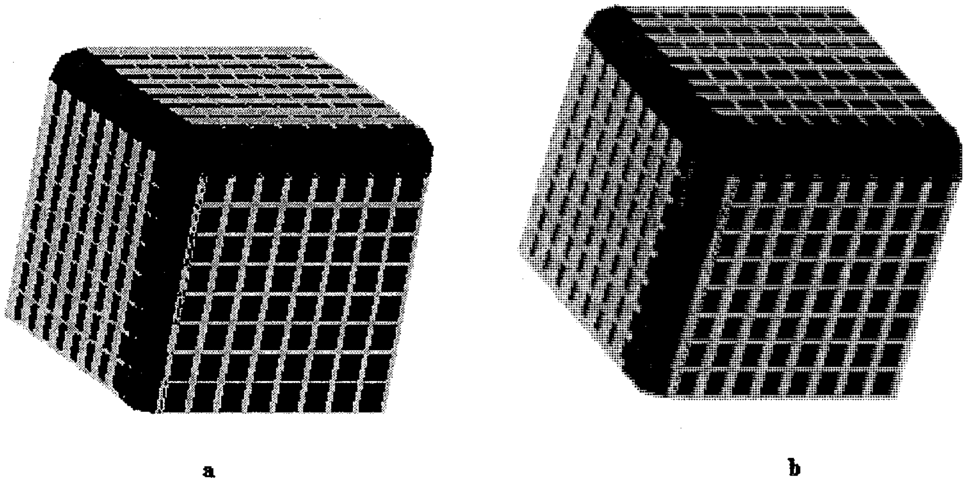


Figure 3.18: Illustration of case 2 – Blue color surfaces are filled surfaces: **a)** Shows an existing hole around the intersection point with the fillet boundary polygon (parametric boundary curves) as outline; **b)** Shows the smooth connection effects after the hole is filled in by NURB patch.

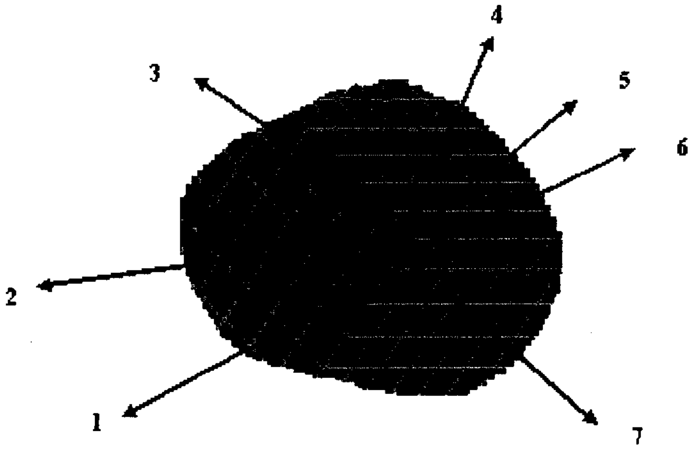


Figure 3.19: Interpolating the control points for three patches' connection case

As three patches are stitched along the boundary surrounding their common joint points [shown on figure 3.19 number 5], we use these 7 data points [red dots above on figure 3.19] to build the control point matrix C in order to generate NURB surface. In this case, we use equation 3.22 to edit the elements of the control point matrix C (its general form is shown in equation 3.20).

C[0][0]=point coordinates (number 4),C[0][1]=point coordinates (number 4),C[0][2]= point coordinate (number 4)
C[1][0]= point coordinates(number 3),C[1][1] = point coordinates (number 5),C[1][2]= point coordinate (number 6) 3.22
C[2][0] = point coordinates(number 2), C[2][1]= point coordinates (number 1),C[2][2]= point coordinate(number 7)

By interpolating the control points using equation 3.22, we obtain the control polygon matrix. Then we substitute this control matrix into equation 3.19 and use equations 3.6, 3.7 and 3.8 and make the order be 3×3 for NURB surface, we obtain the parametric NURB surface (**H(u, v)**) used to fill in the hole which maintains all properties of the NURB surface listed above.

Hence, those C1 continuity conditions are shared for this new generated NURB surface and fillet surfaces along their common boundaries due to the reasons that:

- Boundary parametric curves are overlapped for fillet surfaces and new NURB surface.
- The first order derivatives for both fillet NURB surfaces and this new NURB patch are same at their connected end points.

Therefore, we conclude that:

1. By using this new constructed NURB patch to fill in the hole [shown in the figure 3.18 a)], we obtain the complete smooth NURB surface [shown in the figure 3.18 b)].
2. C1 continuity conditions are held along their boundaries.

<u>**Case 3:**</u> Up to 5 patches are connected together

An example for the stitching of 5 patches is illustrated in the figure 3.20. The figure 3.20-a. illustrates the degenerated patch that includes a hole in the middle. Whereas, in figure 3.20-b, we show the corrected image.

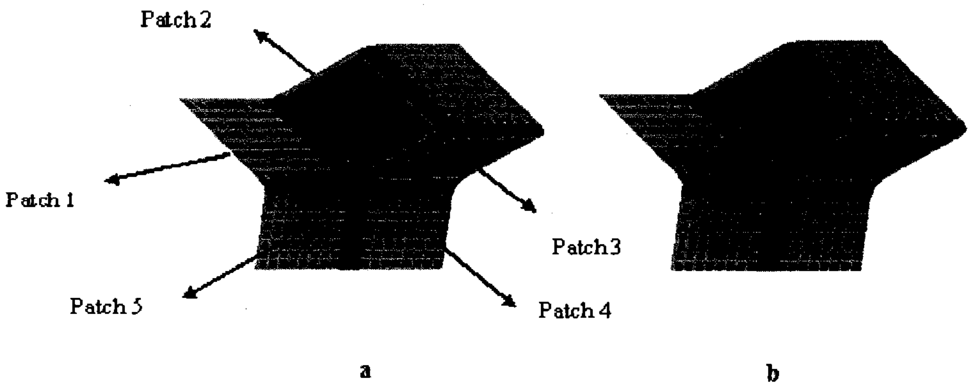


Figure 3.20: Stitching of 5 patches: **a)** Above shows image effect before the hole is filled in; **b)** Above shows a filling effect for a 5- side holes. See the blue area around the intersection point

From the figure 3.20 a, we can see that a 5-sides' hole is created by the boundaries of the fillet NURB patches. Therefore, in order to obtain a complete and smooth NURB surface, we will construct a NURB patch to fill in this hole and let it share the C1 continuity boundary conditions with these five filled NURB patches at their common boundaries and common end points. Although based on the current existed method, S-patch proposed by T. DeRose and C. Loop (1989) can be addressed for solving this problem. The generalized smooth patching method, used in our examples (car modeling), illustrated in the figure 3.21 below.

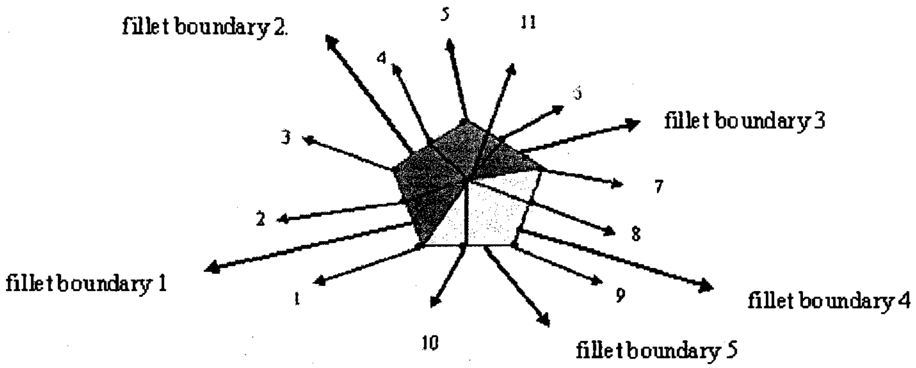


Figure 3.21: Interpolating control points for 5 patches' connection case

As five patches are stitched along the respective boundary surrounding their common joint point (shown by number 11 on figure 3.21), we process *Patch 1*, *Patch 2*, *Patch 3*, *Patch 4*, and *Patch 5* using the C1 smooth connection so that these patches are combined into one smooth parametric surface by means of the joining of the fillet NURB patches along the boundaries (fillet boundary 1, fillet boundary 2, fillet boundary 3, fillet boundary 4, and fillet boundary 5). Thus, a hole (shown in figure 3.20 a) is formed by these fillet patches' boundaries. Therefore, we construct the NURB patch to fill in this hole so that the C1 continuity conditions can be held along their boundaries and at the common end points. We now explain our method in mathematic way below:

We separate the hole into two parts colored by blue and grey shown on figure 3.21.We now interpolate the data points 1, 2, 3, 4, 5, 6, 7, 11 and let points 1, 2, 10 be collinear and let point 1 be the center point of the point 2 and point 10 (note 1*) as control points to build up a NURB patch to fill in the blue hole and interpolate data points 1, 11, 7, 8, 9, 10 and let points 6, 7, 8 be collinear and let point 7 be the center point of the point 6 and point 8 (note 2*) as control points to build up another NURB patch to fill in the grey hole. We edit the control point matrix C (shown in equation 3.20) for blue area by using the equation 3.23 shown below to interpolate its elements:

C[0][0] = point coordinates (number 3), C[0][1] = point coordinates (number 4), C[0][2]= point coordinates (number 5)

C[1][0] = point coordinates (number 2), C[1][1] = point coordinates (number 11),C[1][2]= point coordinates (number 6) 3.23

C[2][0] = point coordinates (number 1), C[2][1] = point coordinates (number 11),C[2][2]= point coordinate s(number 7)

And, we use equation 3.24 below to edit the control point matrix C (shown in equation 3.20) for the grey area:

C[0][0]= point coordinates (number 1), C[0][1]= point coordinates (number 11), C[0][2]= point coordinates (number 7)
C[1][0]= point coordinates (number 10),C[1][1]= point coordinates (number 11),C[1][2]= point coordinates (number 8) 3.24
C[2][0]= point coordinates (number 9), C[2][1]= point coordinates (number 9), C[2][2]= point coordinates (number 9)

Note 1: C[2][0] = point coordinates (number 1) = 0.5×(coordinates of number 2+ coordinates of number 10). Points number 1, 2, 10 are collinear.

C[2][2] = point coordinates(number 7) = 0.5×(coordinates of number 6 +coordinates of number 8). Point number 6, 7, 8 are collinear. (Equation 3.23)

Note 2: C[0][0] = point coordinates (number 1) = 0.5×(coordinates of number 2+ coordinates of number 10). Points number 1, 2, 10 are collinear.

C[0][2] = point coordinates(number 7) = 0.5×(coordinates of number 6 + coordinates of number 8). Point number 6, 7, 8 are collinear. (Equation 3.24)

After editing the control point matrixes for two NURB patches individually, we substitute these two control point matrixes into the equation 3.19 respectively, and utilize equations 3.6, 3.7 and 3.8 for both of them to arrange the knot parameters so that the generated NURB surfaces completely satisfy the properties of the NURB surface listed above. In this way, we obtain two NURB surfaces to fill in two holes. In terms of the properties of the NURB surface,

1. NURB surface for filling in the blue area shares the C1 boundary conditions with the fillet NURB *patch 1, 2, 3* and NURB surface for filling in the grey color area shares the C1 boundary conditions with fillet NURB *patches 3, 4, 5*;

2. And, these two (blue color NURB surface and grey color NURB surface) share the common boundary (shown by number 1, 11, 7).Meanwhile, due to the arrangement of note 1 and note 2, these two NURB patches along their common boundary and at their two common end points 1, and 7 satisfy the C1 continuity boundary conditions .

Therefore, we conclude that the NURB patch combined from the blue color NURB patch and grey color NURB patch has the C1 smooth transition with the fillet NURB surfaces 1, 2, 3, 4 and 5. Hence, by the joining of this NURB surface and fillet NURB patches, the five stitched NURB patches finally are converted as one smooth parametric NURB surface sharing the C1 continuity conditions along the fillet NURB patches' boundaries and common joining points.

Below, we use a group of images to display C1 smooth connection methods with respect to the following mentioned four cases used in our system.

Case 1: Two stitiched NURB patches.

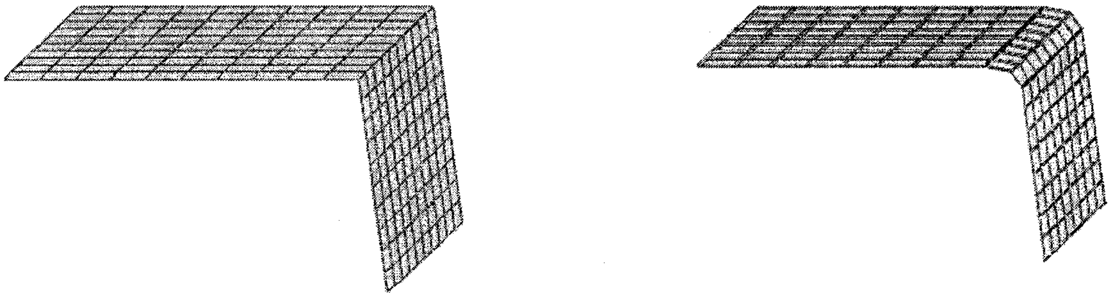


Figure 3.22: **a)** Two stitched NURB patches; **b)** C1 smooth connection for two stitched NURB patches

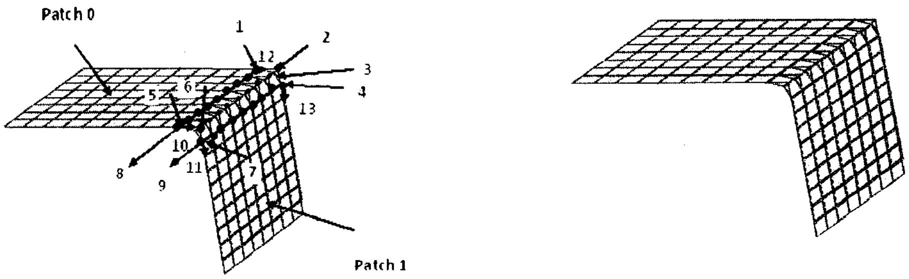


Figure 3.23: a) The construction of the C 1 continuity between two stitched patches. (the left image); **b)** Fillet surface under the control polygon defined by the local NURB patches' points shown in the left image and colored by the red dots.(the right image)

The example provided in figure 3.22 further detailed in figure 3.23. Below, we summarized the mathematix of the figure.

Table 3.1: Details of the figure 3.23

Reference Point	**Explanation**
1	$p0_{m-1,n-1}(C_{0,n1}, H_{0,n1})$
2	$p0_{m,n-1}(C_{1,n1}, H_{1,n1})$
3	$p1_{1,n-1}(C_{2,n1}, H_{2,n1})$
4	Fillet NURB surface defined by the control polygon shown control points in red color dots
5	$p0_{m-1,1}(C_{0,0}, H_{0,0})$
6	$p0_{m,1}(C_{1,0}, H_{1,0})$
7	$p1_{1,1}(C_{2,0}, H_{2,0})$
8	Common boundary between the fillet surface and Patch 0. (overlapped) k=2
9	Common boundary between the fillet surface and Patch 1. (overlapped) k=2
10 and 12	Common end point between the fillet surface and Patch 0 Along the *u* direction pointed by the arrow, the first order derivatives at the common end points are: $p0_{m,1} - p0_{m-1,1} = C_{1,0} - C_{0,0}$ $p0_{m,n-1} - p0_{m-1,n-1} = C_{1,n1} - C_{0,n1}$
11 and 13	Common end point between the fillet surface and Patch 1. Along the u direction pointed by the arrow, the first order derivatives at the common end points are: $p1_{1,1} - p0_{m,1} = C_{2,0} - C_{1,0}$, $p1_{1,n-1} - p0_{m,n-1} = C_{2,n1} - C_{1,n1}$

Note: Symbol Pi is the local point of the ith NURB patch.
Symbol C is the control point of the fillet NURB patch denoted by H.
Symbol H is the fillet NURB point.

Case 2: Four stitched NURB patches.

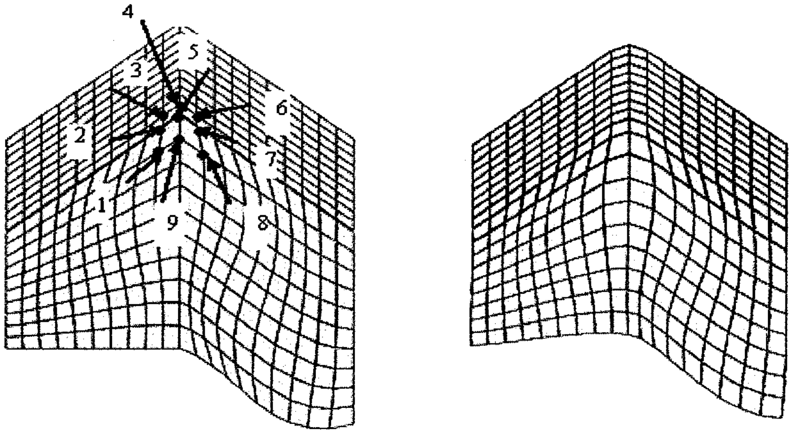


Figure 3.24: a) Four stitched NURB patches; **b)** C1 smooth connection for four stitched patches

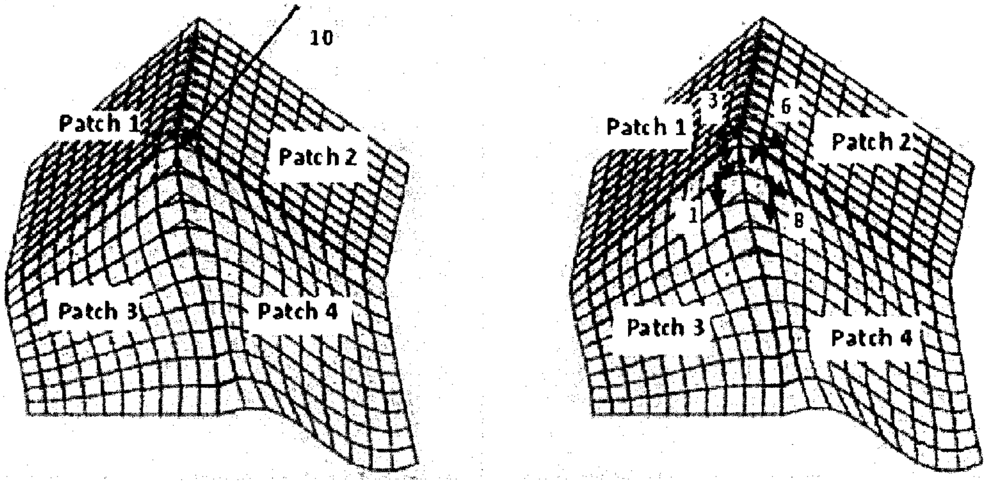


Figure 3.25: a) The construction of the NURB patch used to fill in the hole between four stitched patches with C1 continuity smooth transition

In figure 3.25 the control polygon of the NURB patch consists of the local points (numbers 1 to 9 shown in figure 3.24-a) of these four NURB patches shown in the red color. Points (number 1,3,6,8 in figure 3.25) shown in the pink color are the common end points between the NUBR patch and the four fillet NURB patches. The arrows represent their first order derivatives at these four end points.

points between the NUBR patch and the four fillet NURB patches. The arrows represent their first order derivatives at these four end points.

Denote: Point 1 as $p3_{1,n-1}$ (number 1), point 3 as $p1_{m-1,1}$ (number 3), point 6 as $p2_{1,1}$ (number 6), Point 8 as $p4_{1,1}$ (number 8). Using equation 3.20 to construct the control points matrix for the NURB patch. Thus:

$$\begin{aligned}
&p3_{1,n-1} - p3_{0,n-1} = C_{2,0} - C_{1,0} \\
&p3_{1,n} - p3_{1,n-1} = C_{2,1} - C_{2,0} \\
&p1_{m-1,0} - p1_{m-1,1} = C_{1,0} - C_{0,0} \\
&p1_{m,1} - p1_{m-1,1} = C_{0,1} - C_{0,0} \\
&p2_{1,1} - p2_{0,1} = C_{0,2} - C_{0,1} \\
&p2_{1,0} - p2_{1,1} = C_{1,2} - C_{0,2} \\
&p4_{1,1} - p4_{1,0} = C_{2,2} - C_{2,1} \\
&p4_{1,1} - p4_{0,1} = C_{2,2} - C_{1,2}
\end{aligned}$$

Note: We use $pi_{m,n}$ to represent the local NURB surfaces' points where i : patch number, *m*: row, *n*: coloumns. We use $C_{i,j}$ to denote the control points of the NURB patch used to fill in the hole. Thus, in terms of the above equations, the left sides are the first order derivatives of the local NURB surfaces' points 1, 3, 6, and 8. The right sides represent the first order derivatives of the NURB patch at these points 1, 3, 6, and 8 (which are common end points). We let $k = 3$ along both *u* and *v* directions to gurantee the overlap of the boundaries. Therefore, C1 continuity boundary conditions are satisfied between the new NURB patches used to fill in the hole and the four stitched NURB patches which have been connected by the fillet NURB surfaces.

<u>**Case 3:**</u> Three stitched NURB patches.

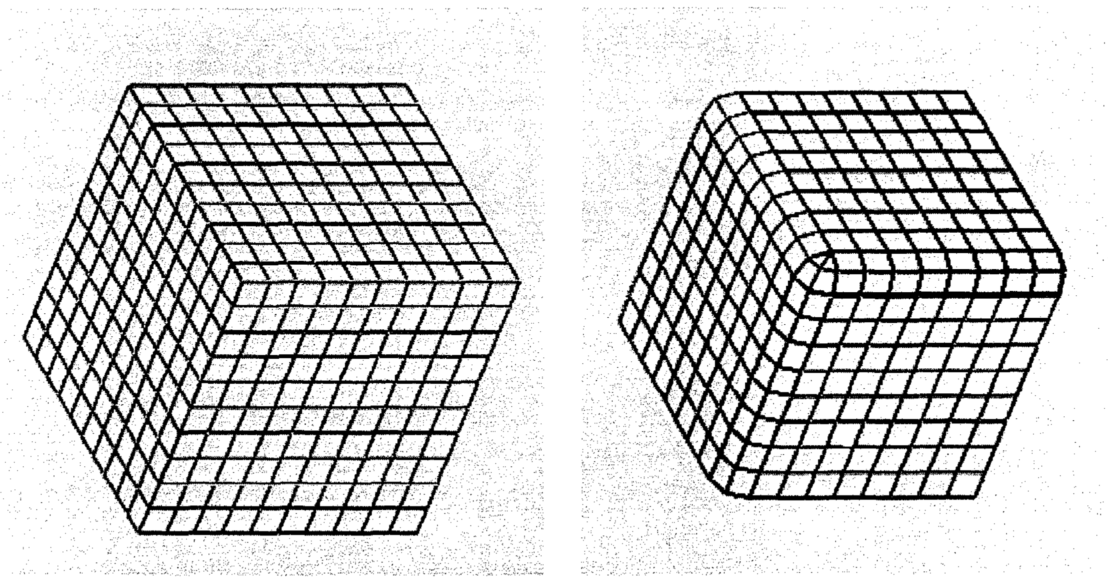


Figure 3.26: Example for the three patch stitching

a) Three connected NURB patches; **b)** C1 smooth connection for three stitched patches

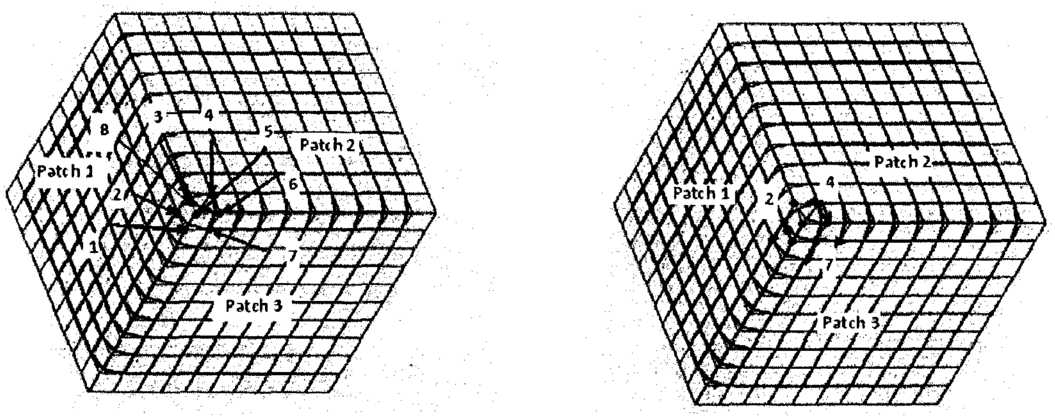


Figure 3.27: The details of the image on figure 3.26- Illustration to motivate mathematical presentation

The left image in figure 3.27 is the construction of the NURB patch used to fill in the hole in three stitched patches case. The control polygon consists of the points of the local NURB patches' colored by the red dots and labed by number(1, 2, 3, 4, 5, 6, 7) (left image). In the right image of figure 3.27 the numbers 2, 4, 7 are the common end points between NURB patch to fill in the hole and other three NURB patches (*Patch 1, Patch 2, Patch 3*). The arrows shown on this image are their first order dirivatives along u direction and v direction.

Denote: point 4 as $p2_{1,1}$; point 2 as $p1_{m-1,1}$; point 7 as $p3_{1,1}$. From of the equation 3.22:

$$\begin{array}{ll}
C_{0,0} = C_{0,1} = C_{0,2} = p2_{1,1} \text{ (number 4)} & C_{1,0} = p2_{0,1} \text{ (number 3)} \\
C_{1,1} = p2_{0,0} \text{ (number 5)} & C_{1,2} = p2_{1,0} \text{ (number 6)} \\
C_{2,0} = p1_{m-1,1} \text{ (number 2)} & C_{2,1} = p1_{m-1,0} \text{ (number 1)} \\
C_{2,2} = p3_{1,1} \text{ (number 7)} & \\
p2_{0,1} - p2_{1,1} = C_{1,0} - C_{0,0} & p2_{1,0} - p2_{1,1} = C_{1,2} - C_{0,0} \\
p1_{m-1,1} - p1_{m,1} = C_{2,0} - C_{1,0} & p1_{m-1,0} - p1_{m-1,1} = C_{2,1} - C_{2,0} \\
p3_{1,1} - p3_{0,1} = C_{2,2} - C_{1,2} & p3_{1,1} - p3_{1,0} = C_{2,2} - C_{2,1}
\end{array}$$

Note : $pi_{m,n}$ is the local NURB patches's points i: patch number, m:row, n: column. Thus, the left sides of the above equations represent the first order derivatives of these local NURB patches' points 4, 2 and 7. $k = 3$ for both u and v direction so that the boundaries between *Patches 1, 2 and 3* and new NURB patch are overlapped . $C_{i,j}$ is the control points of the NURB patch used to fill in the hole. Thus, the right sides of the above equation represent the first order derivatives of the new NURB patch at common end points.

Therefore: this NURB patch share the C1 continuity with the stitched three patches (after the jioning of the fillet NURB patches) at their common boundaries and common end points 2, 4 and 7.

<u>Case 4 :</u> Five stitched NURB patches

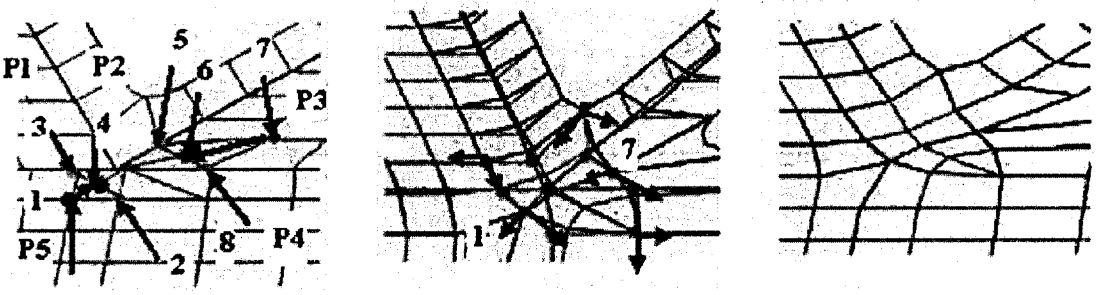


Figure 3.28: Illustration of five stitched NURB patches: **a)** Five stitched patches; **b)** Boundary derivatives (first order) at common end points for five patches' and their fillet patches and two NURB patches used to fill in the holes; **c)** C1 smooth connection effect for five joining patches' case.

In the figure 3.28-a, the point numbers 1 and 7 (colored by dark red dots) are the common end points lie on the *patch 5*(P5) and *patch 3* (P3) and new NURB patch used to fill in the hole. Interpolate the midpoint of point number 3 and point number 2 to obtain point number 4 and let point 3, 2, 4 collinear. And then let the point number 1 take up the position of the point number 4 so that point 3, number 1 and number 2 are collinear. The same for the points number 5, 6, 7 and 8. Interpolating the control points for two NURB surfaces using the equations 3.23 and 3.24, we obtain these two NURB surfaces which have the C1 continuity conditions (boundary being overlapped, first order derivatives are same at their common end points 1 and 7).

Note: let k=3 for both u and v directions to keep the boundaries being overlapped. This NURB patch used to fill in the hole consists of two NURB patches using the same method shown in case 1. Therefore, these two NURB patches are satisfied the C1 continuity with the five NURB patches at their common boundaries.

For the two NURB patches, due to the arrangement of control points 1 and 7, their first order derivatives at point1 and 7 (common end points) are the same and the boundaries are overlapped. Thus, the boundary conditions for the C1 continuity are held between this NURB surface and the five stitched NURB patches (after the joining of the fillet NURB surfaces).

Up to this point, we discussed a real-time parametric surface (NURB) generation technique, the smooth connection of individually generated surface patches with respect to the C1 continuity rules, exception handling via filling the gaps form during the parametric stitching operations. In the following section, we discuss the thickness issue for the virtual objects.

3.3.3. Discussion on the thickness formation on an arbitrary 3D surface

To simulate the real objects, it is necessary to determine the thickness of the designed objects. As mentioned earlier in the literature review, some CAD/CAM methods such as extrusion can be used to add thickness to the objects. However, for the arbitrary drawing objects, giving an exact thickness to them is a challenge job due to the irregularly forming 3D geometry. Particularly when the scaling operations are applied to generated surface patches, without an efficient method, scaled objects may show errors similar to the images shown in figure 3.29. Correcting such errors may require complex calculations.

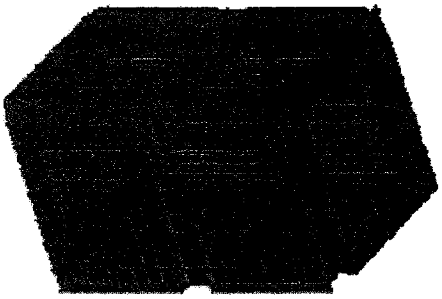


Figure 3.29 Adding thickness to the designed objects by using extrusion method

To avoid the problems illustrated in figure 3.29, we developed an accurate method for our free-form sketching toolkit to add the desired thickness to the 3D shapes which are the parametric surfaces. Proposed thickness formation method can be summarized as:

- Calculate the geometric center for the designed objects.
- Pull the geometric center of the designed objects to the original point set up in our system.
- Given amplification coefficient for x, y and z directions.
- Connect the corresponding boundary points on two images.

By following the above sequence, we obtain the desired thickness. Now, let us further elloborate the each step.

- **Calculate the geometric center for the designed objects**

Given: $\{S_i\}$ $i=\{0,1,\cdots,n\}$ notes as NURB patches Where i: is the patch number, let: $SM_{i,x}, SM_{i,y}, SM_{i,z}$ be the coordinates of the middle point on $\{S_i\}$ where $i=\{0,1,2,\cdots n\}$

We now calculate the geometric center of the designed object with respect to $\{S_i\}$:

We denote $GC_{i,x}, GC_{i,y}, GC_{i,z}$ as the variable of the geometric center of the designed object:

Then:

$$GC_{i,x} = \frac{\sum_{i=0}^{n} SM_{i,x}}{n+1}$$
$$GC_{i,y} = \frac{\sum_{i=0}^{n} SM_{i,y}}{n+1} \qquad 2.25$$
$$GC_{i,z} = \frac{\sum_{i=0}^{n} SM_{i,z}}{n+1}$$

- **Pull the geometric center to the original point set up in our system**

We define the translation matrix ***T*** to the designed objects, and use it to pull the geometric center of the designed objects to the original point set up in our system

$$T = \begin{bmatrix} 1 & 0 & 0 & -GC_{i,x} \\ 0 & 1 & 0 & -GC_{i,y} \\ 0 & 0 & 1 & -GC_{i,z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 3.26$$

Suppose that the given scaling coefficients are t_x, t_y, t_z. Scaling the object according to the coefficients, and connecting the corresponding points along the patch boundaries, we obtain the thickness for the designed object. If the shape of the designed object is not closed (see figures 3.30, 3.31 and 3.32 for illustration), we define the T1 as a transformation matrix to transfer the image of the scaled object to the direction which is opposite to the open face by one thickness unit to guarantee the image boundary along the open side enjoying the same x, y or z value on the edges to avoid the problem shown on the figure 3.30:

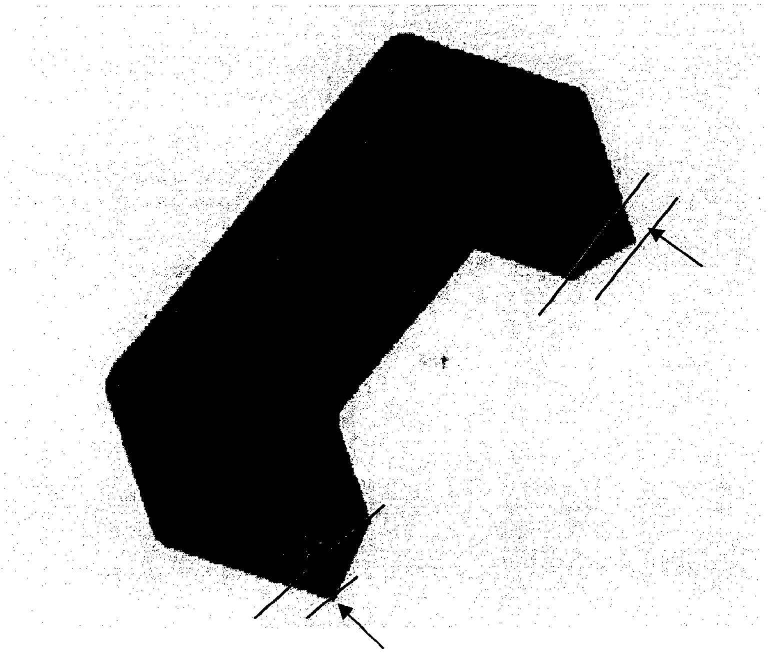


Figure 3.30: Add thickness to the designed objects with open sides without doing certain modifications

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & -t_{1x} \\ 0 & 1 & 0 & -t_{1y} \\ 0 & 0 & 1 & -t_{1z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 3.27$$

The wireframe and the solid image given in the figure 3.30 are illustrated in figures 3.31 and 3.32.

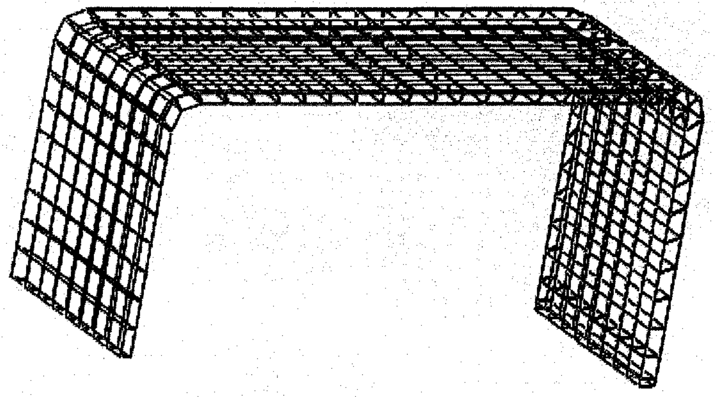


Figure 3.31: Add thickness to the designed objects

Figure 3.32: Solid images for adding the thickness to the designed objects

Compared with this thickness added method, another approach which is available to add the exact thickness information to the designed objects is the direct drawing method, which uses patches to enclose the boundaries to give the thickness and volume representation. It is the easiest and intuitive way; however, it is not suitable for all cases. For example, if we design an object which has the shell-shape just like the car shell, the thickness information of the car-shell is not greater enough to support designers to draw. Or to say, this drawing process will be time-consuming and cumbersome.

3.3.4. Methods used to trim the designed surface

In order to obtain the good effects for the designed objects, sometimes the trim methods are required. In our system, we use three ways to deal with this problem.

- Using method one, we take out the NURB surface local points and using them to reconstruct inside frame parametric curves (NURB spline), and by redrawing we display the desired image segments (suppose for car windows). See figures 3.33a), 3.33b) below.

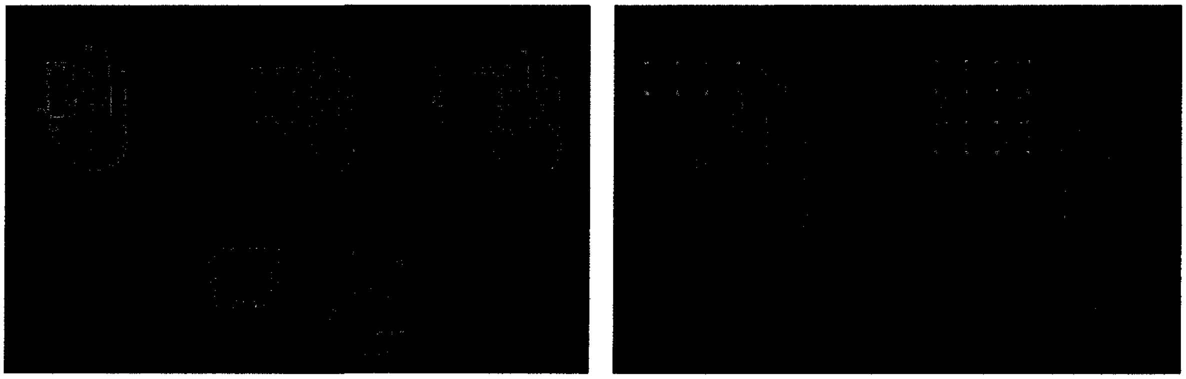


Figure 3.33: a) Left image for trimming the NURB surface generated by multi-point-generation data; **b)** Right image for trimming the NURB surface generated by single-point-generation data.

From the figure 3.33 a), 3.33b), along the inside frame curves, red points are shown; they are the rebuild control points which can be moved by using the tip of the middle index finger. Additionally, the change of the position of the local NURB surface points does not affect the other local points. In other word, if we modify the inside frame curves by moving those red control points, the boundary curves of the NUBRS surface will not suffer affections, thus if the frame shape for the trimmed surface is not satisfied, we can perform the local modification to obtain the desired outline shape on the existed NURB surface. See figure 3.33 c).below.

The left image of figure 3.33 c) shows the modification effects by moving the red control points of the inside frame parametric curves. The change of the position of the inside frame curves' doesn't affect the boundary curves' shape.

The right image of figure 3.33 c) shows that moving the boundary curve control points not only causes the shape change for the boundary curve but also causes the shape change for the inside frame curves.

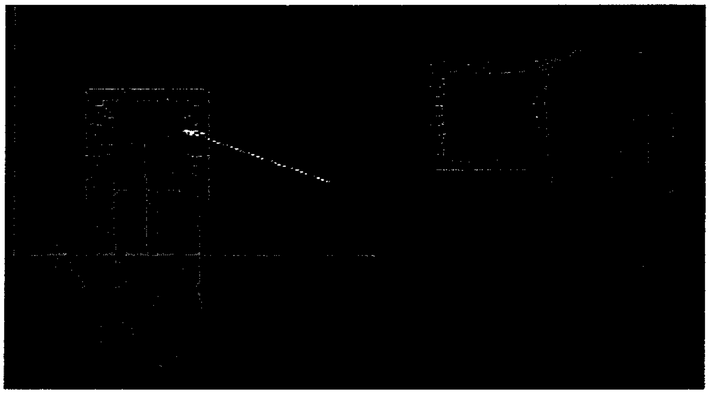


Figure 3.33: c) Modification for trimmed NURB surface

- Another trimming method used in our system is to take out the NURBS surface local points to reconstruct the outline curves (NURB spline) on the designed surface, then by using the background color (suppose dark color) with a certain thickness to hide the undesired images around the shape outline curves, we can display the desired images. See figure 3.34 below:

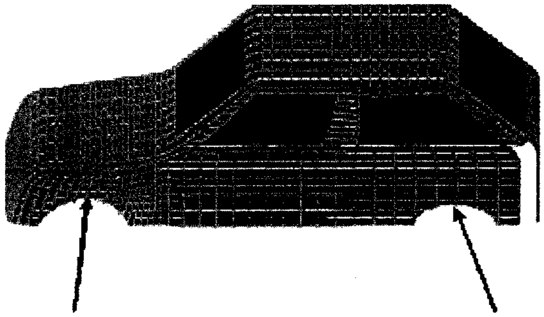


Figure 3.34: The outlines pointed by the black arrows show the image effects by using the second method.

- The third trimming method we use in our system is just to hide the undesired images and keep the desired images to be shown on the screen. See figure 3.35below.

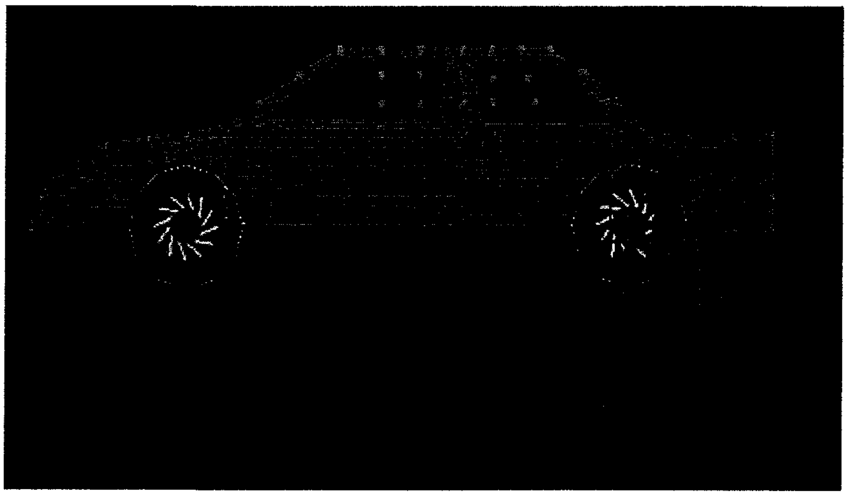


Figure 3.35: Method three for trimming the NURB surface

We analysis these three methods used in our system.

1. Method one [figure 3.33-a, figure 3.33-b] provides a fast way to build the inside shape on existed NURB surface by writing the library functions. And, by moving the control points, the desired inside outline shape can be obtained.

   This method can be introduced in a mathematic way below:

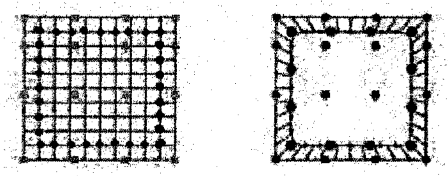


Figure 3.36: Method one for trimming the NURB surface

We denote the data points shown by red, black, blue, and green dots and lied on the NURB surface as $\{d_i\}$ i={0,1,2,....m} in our case i={0....8}. Denote parametric curve data as $\{p(u)\}$ 0<= u <=1; and $\{C_i\}$ i={0,1,....m}. In our case i={0,...3}.

We use lease-square method to fit these data to obtain the NURB parametric curve using the equations 3.6, 3.7, 3.8and follow the steps below:

1. Calculate the u value.
2. Calculate the control points.
3. Calculate the NURB parametric curve points' coordinates.

4. Draw the image.

Similar to the lease-square method we use in NURB surface fitting. We give the general form of it as below:

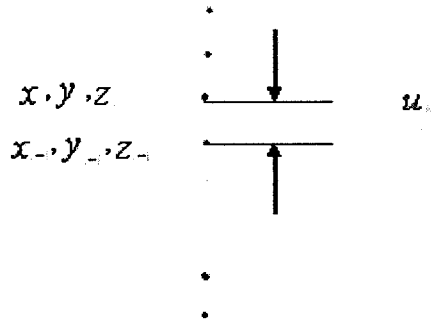


Figure 3.37: Draft for calculating the *u* value

1. **Calculating the u value.**

According to the figure 3.36, we calculate the u value using the equation shown below:

Let L_i is the length between two adjacent points, suppose X_i and X_{i-1}

Then:

$$L_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \qquad 3.28$$

$$u_k = \frac{\sum_{i=0}^{k} L_i}{\sum_{i=0}^{m} L_i} \qquad i = \{0,1,......m\} \tag{3.29}$$

2. **Calculate the control points' coordinates.**

$$\begin{bmatrix} d_0 \\ \vdots \\ d_k \end{bmatrix} = \begin{bmatrix} R_0^{m+1}(u_0) & \cdots & R_m^{m+1}(u_0) \\ \vdots & \vdots & \vdots \\ R_0^{m+1}(u_k) & \cdots & R_m^{m+1}(u_k) \end{bmatrix} \begin{bmatrix} C_0 \\ \vdots \\ C_m \end{bmatrix} \tag{3.3}$$

Denote N:

$$N = \begin{bmatrix} R_0^{m+1}(u_0) & \cdots & R_m^{m+1}(u_0) \\ \vdots & \vdots & \vdots \\ R_0^{m+1}(u_k) & \cdots & R_m^{m+1}(u_k) \end{bmatrix} \tag{3.31}$$

$$N^T d = N^T N C \tag{3.32}$$

3. **Calculate** $p(u)$ **for parametric curve coordinates.**

$$P(u) = \sum_{i=0}^{m} C_i R_{i,k}(u) \tag{3.33}$$

We use equations 3.6, 3.7, 3.8 to determine the value of $R_{i,k}(u)$. By using the above method and utilizing the local NURB surface points, we obtain the parametric curve. And then we connect the new generated NURB parametric curves with the local NURB surface and show the desired image to obtain the trimming effects on the existing NURB surface.

It is worth to note, on the right image shown on figure 3.36. We build up a fillet NURB curve to achieve the C1 smooth transition between two adjacent parametric curves, the method we use in this part is same as that of building fillet NURB surface for two stitched NURB patches. It can be treated as the special case of it. Here, we don't re-address it again.

2. Method two [figure 3.34] is the most intuitive method. By using hand to select local surface points as data and using lease square method to reconstruct parametric NURBS curve as outline as well as using color to hide the undesired image, we obtain the visible trimming effects on the designed objects. The mathematic way for method two is the same as that of method one. Only different is that after building the parametric outline curves on the trimmed NURB surface, we need to use extrusion method to construct an object with a certain thickness and color it with the background color so that we can hide the undesired image and show the desired part on the screen.

3. Method three [figure 3.35] is suitable for the regular shape trimming purpose. Using this method, we apply the background color with certain thickness to hide the undesired image surrounded by the outline shape generated from parametric equations of the regular shape objects.

Comparing three trimming methods with the methods by using algorithms to remove the extra data from the trimmed surface, if we only peruse the visible image effects in conceptual design stage, three methods we use in our system can help us to achieve this purpose quickly and easily.

3.3.5. Dimension modification capability for the designed objects

To the designed objects, we use glScalef function to modify the proportions of their images.

See figure 3.38a), 3.38b) below:

Images' comparisons before dimension modification and after dimension modification

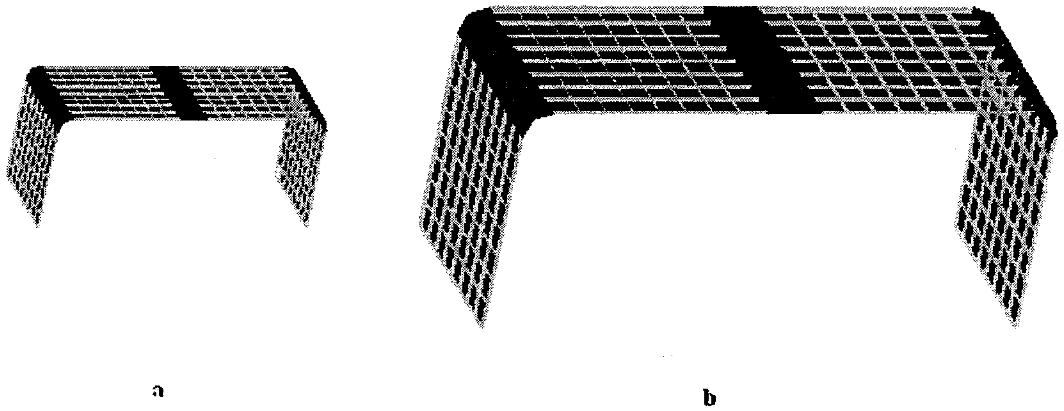


Figure 3.38: **a)** Left image is the original drawing; **b)** Right image is the modified drawing

By now, in the above sections, we have discussed three main steps which are used to fulfill free-form conceptual design.

1. Real-time parametric surface (NURB).
2. C1 smooth connection.
3. Adding thickness to the arbitrary drawn object.

In our free-form conceptual design system, the index finger of the right hand is chosen as drawing tool and used to draw points, NURB spline, and NURB surface and used to perform the surface shape modification by moving the control points or data points of the patches in real-time. It is worth to note, in our system, the expression for the objects' shape depends on the drawing process not depending on the modification process; modification process is only an assistant process with respect to the whole design process. The connection for different NURB splines or surfaces can be easily done by directly drawing the boundary data along the existed patches' boundaries so that C1 smooth connection methods can be applied directly after drawing. In the third step, we discuss an approximate method to add thickness to the designed object.

Additionally, in the research of our method, throughout we believe that hand is the more flexible and natural expression tool for drawing purpose. With the use of 5 DT data-gloves and 6-degree trackers, those natural expression features can be simulated and can be used to replace keyboard or mouse to conduct computer to perform the control commands by giving the definitions for the hand gestures in program. For example, we calibrate the curvature of the right finger and do a counter with the initial value as zero in our program, if the curvature of this finger is greater than suppose 20 degree, computer performs the drawing command so that we draw a point image at the desired position and add one number to the counter; if the curvature is smaller than 20 degree, we stop drawing. It is operated just like by pressing to draw. This increases the intuitive sense to the design process, and makes the design process very close to the real world work.

We compare several methods which use the different design tools or different parametric representations methods through the following images.

Drawing methods comparison for the generation of parametric curves:

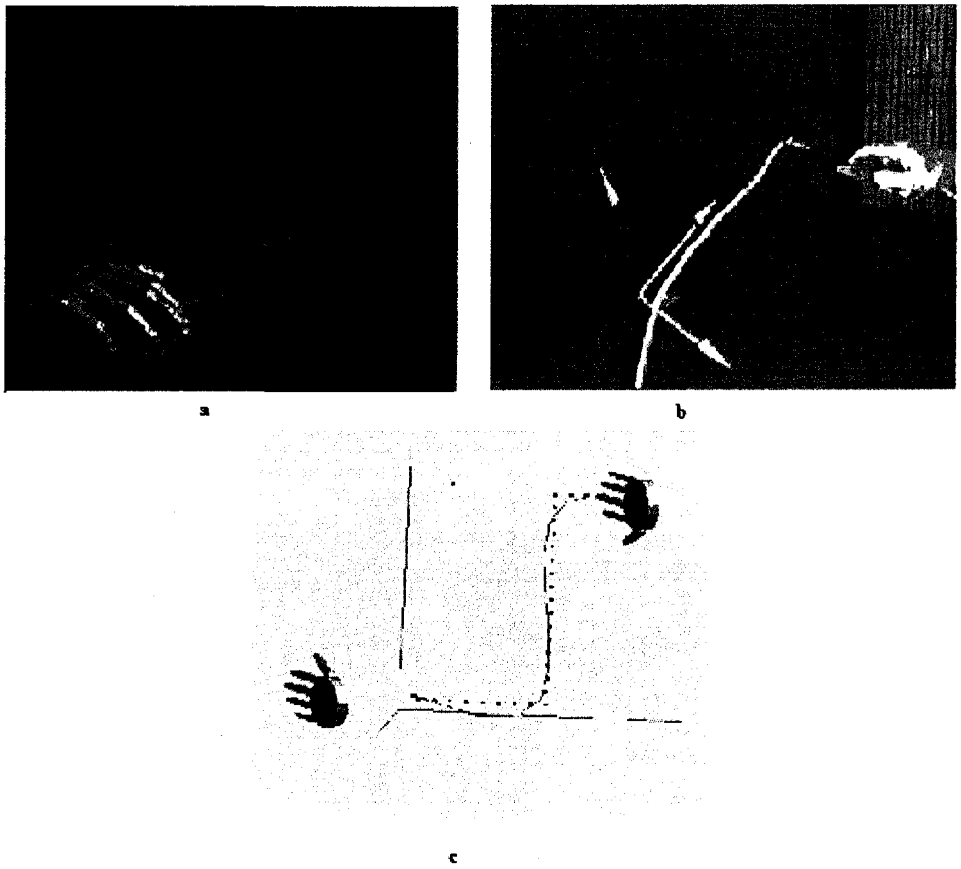


Figure 3.39: Comparison of three basic parametric curves' generation methods used in free-from conceptual design

a) Three steps parametric curve generates display (Akgunduz A. and Yu. H. 2005). **b)** B-spline curve generating (Wesche, G. and Seidel, H-P (2001)). **c)** Bezier curve generated by lease-square method

Drawing methods comparison for the generation of the parametric surface:

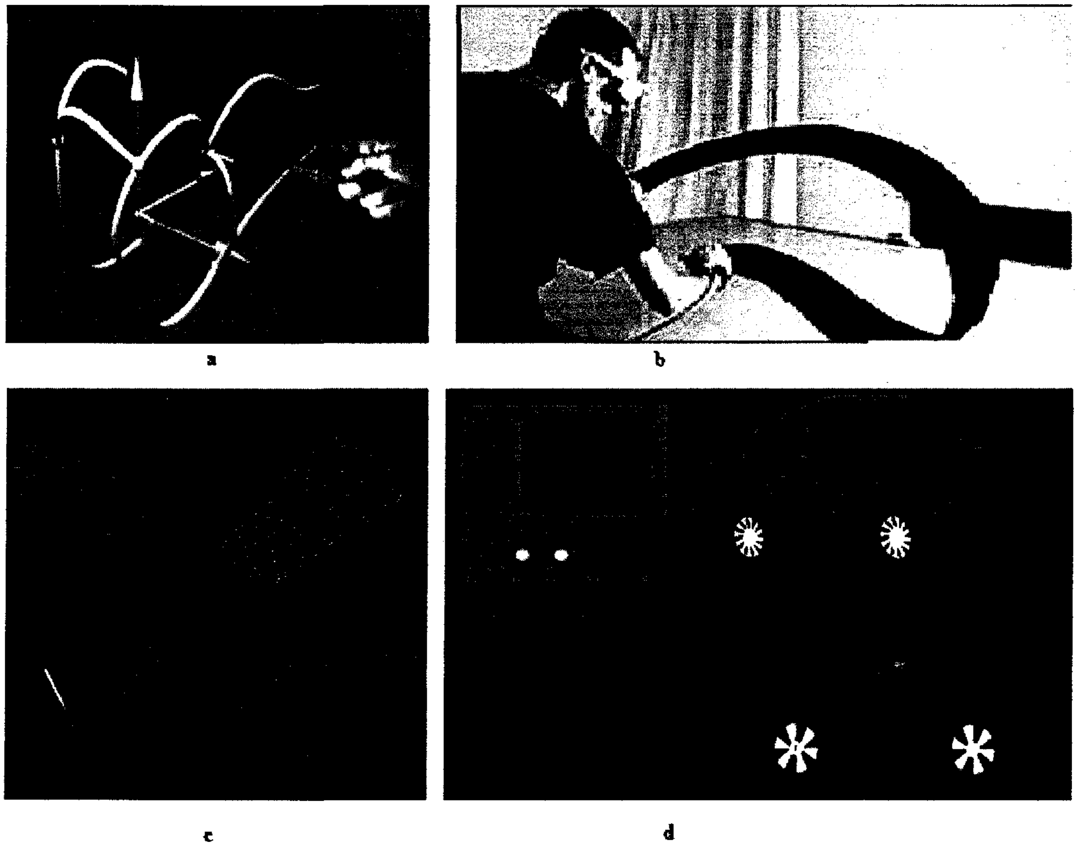


Figure 3.40: Evolution of the proposed free-form sketching technique:

a) Parametric surface generated from the edited skeleton built by parametric curves (Wesche, 2001)
b) Direct polygon mesh surface (Schkolne, 2006)
c) Three-steps method to generate parametric surface (Akgunduz , 2005)
d) Direct NURBS parametric surface patches with C1 continuity boundaries

In this chapter, we discuss three main steps used in our system to perform the free-form conceptual design using mathematic expressions and explanations. In the following chapter, we will display several drawings with some problems to show the developing process of the above methods.

Chapter 4

Advanced problems in free-form sketching And Examples

In chapter 3, we introduced the free-form sketching methods used in our system. In this chapter, we use a series of figures to demonstrate the development of these methods and provide four examples to demonstrate the results.

4.1. Exception Handling: Advanced problems in free-form sketching

In this section, we use the following figures to demonstrate the improvement of the methods.

When this project started, the initial goal was to develop a sketchmg technique that enables designers to express their thoughts in 3D space. Knowing that working with parametric surfaces is the most flexible technique in computer graphics, we develop an algorithm to create surface patches from the data acquired from the user hands. Although the results were as we expected, several new challenges were identified.

When we added (stitched) the surface patches together with an attempt to create finished product designs, we realized there is a transition problem between surface patches as shown in figure 4.1. Although a car would have round corners, our method was generating sharp edges. Another issue was removing the unwanted material from patches. For example, a car body shown in figure 4.1 also requires windows. From the designer's point of view, we realized that removing the window shape from the finished car body is a lot easier and intuitive than designing all the parts of car without the window using several surface patches. Although cutting the window from the car body is easier and intuitive for the designer, mathematical modeling it in a real-time sketching model was an extremely challenging task.

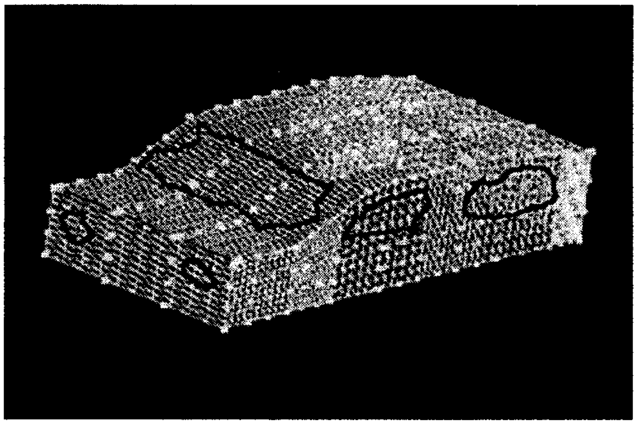


Figure 4.1: First car model

The car model shown in Figure 4.1 only consists of several stitched NUBR patches and shows in the wireframe mode. This car model is lack of details such as: windows, doors, lights, and smooth transition between the stitched patches.

In the following car model (figure 4.2), we applied our removing and adding operations. As seen in the figure, we now are able to deal with the removing of unwanted materials from the parametric surfaces in any convex shape. However, the figure still posses smooth connection problems.

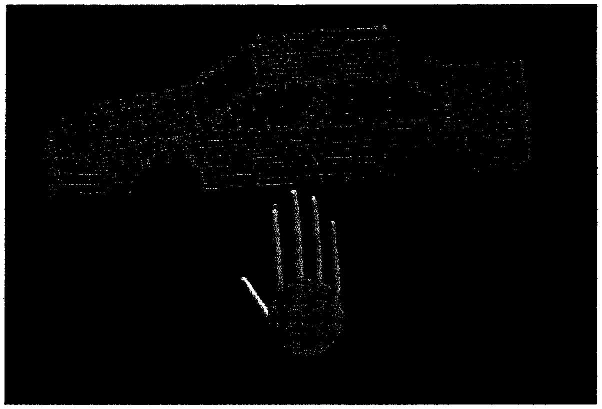


Figure 4.2: Model includes the material removing properties

Following two figures in figure 4.3 pose the connectivity problems between NURB patches. These figures generated using the 5DT data gloves are still not satisfactory for the professional design.

Figure 4.3: Car models with C1 Connectivity problems

Both of the car designs given in figure 4.4: display the necessity for trimming on existing NURB surfaces. See the region pointed by the arrows for illustration. In Figure 4.4, the blue color areas show the C1 transition connection which consist of the fillet NURB patches and NURB patches to fill in the holes.

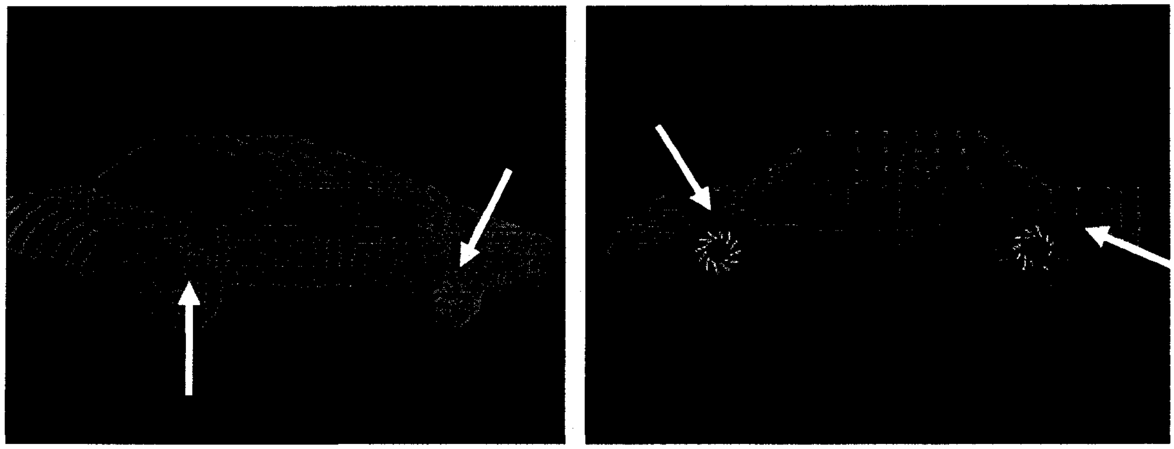


Figure 4.4: Car designs, their C1 Continuity problem is resolved still require trimming for improvement

In figure 4.5, we display finished images in both wireframe and solid views.

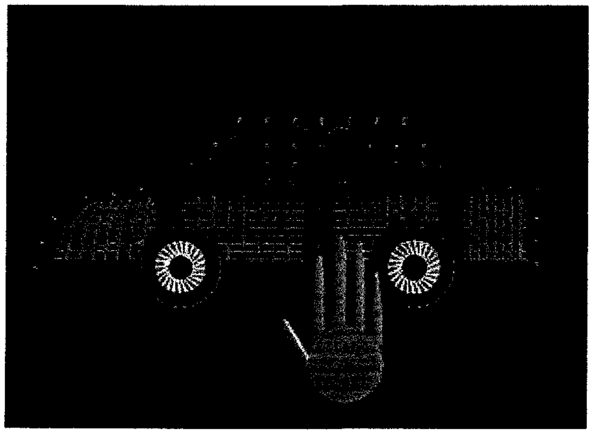


Figure 4.5: Wireframe Image of the finished model

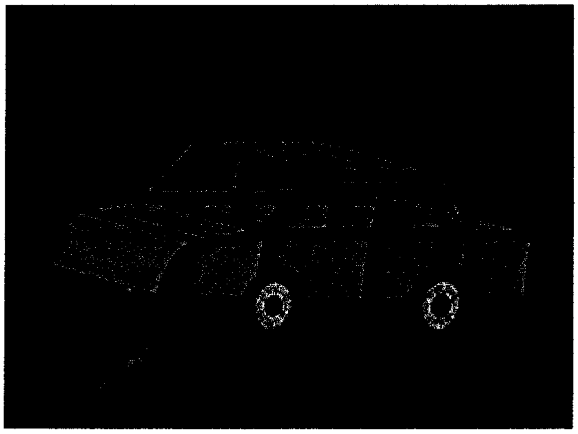


Figure 4.6: Solid image of the finished model

In the figure 4.7, we display another finished car image in both wireframe and the solid views. The design given in figure 4.7 includes surface thickness as well.

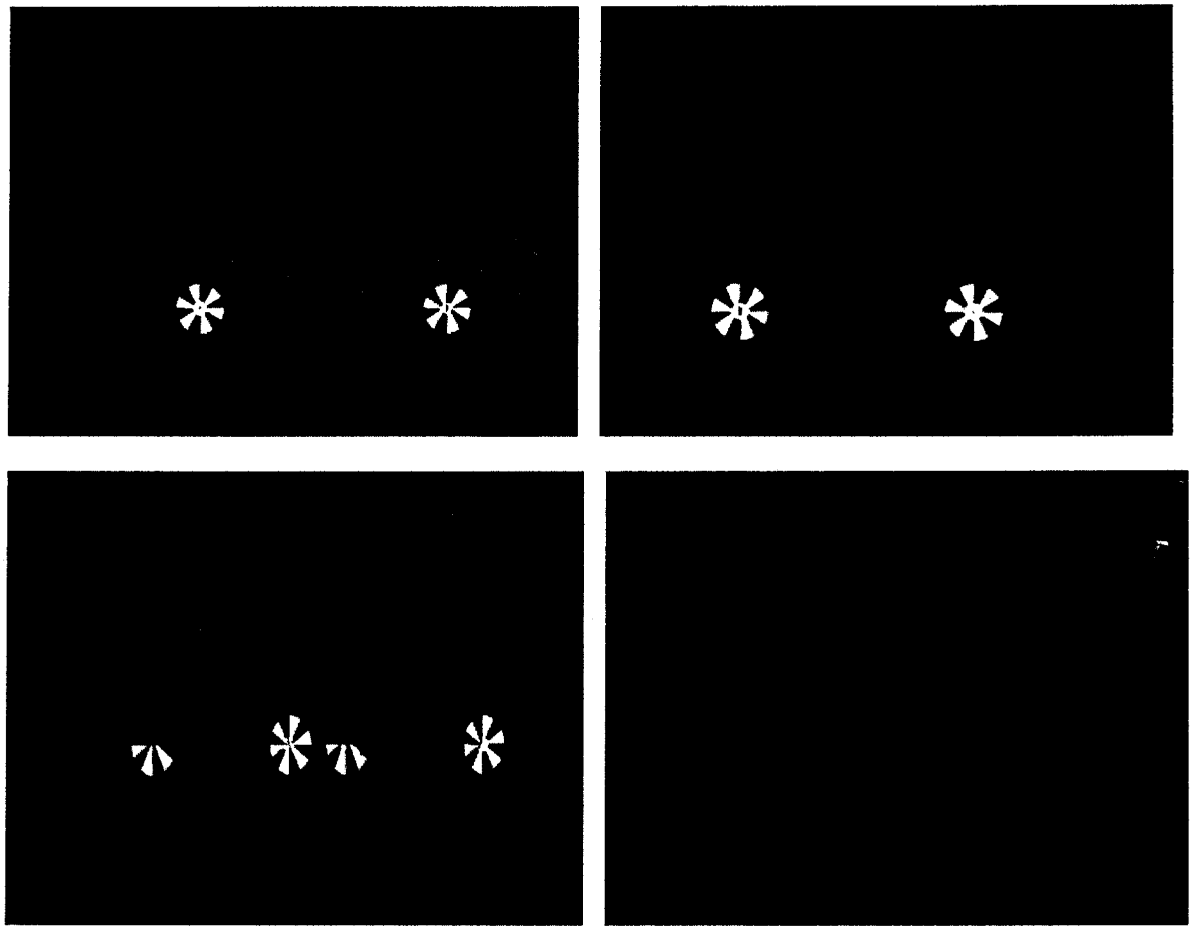


Figure 4.7: Car model with surface thickness in both wireframe and solid views

4.2. Examples

In this part we demonstrate the capabilities of our free-from sketching technique in various examples. These include a car model, table lamp model, microwave oven model and a weighting room model.

4.2.1. Car model

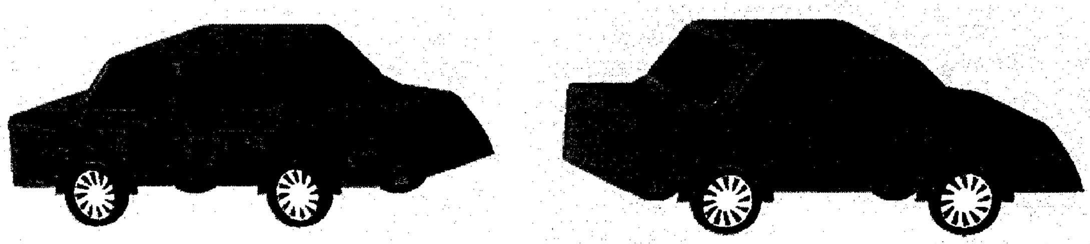


Figure 4.8: Finished car model in black and white format

4.2.2. Table lamp

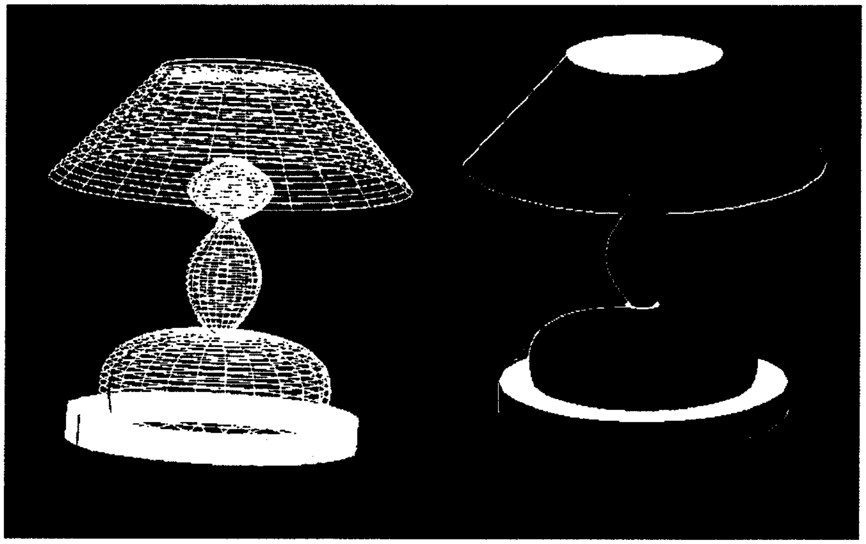


Figure 4.9: A table lamp in wire and solid format

4.2.3. Microwave oven

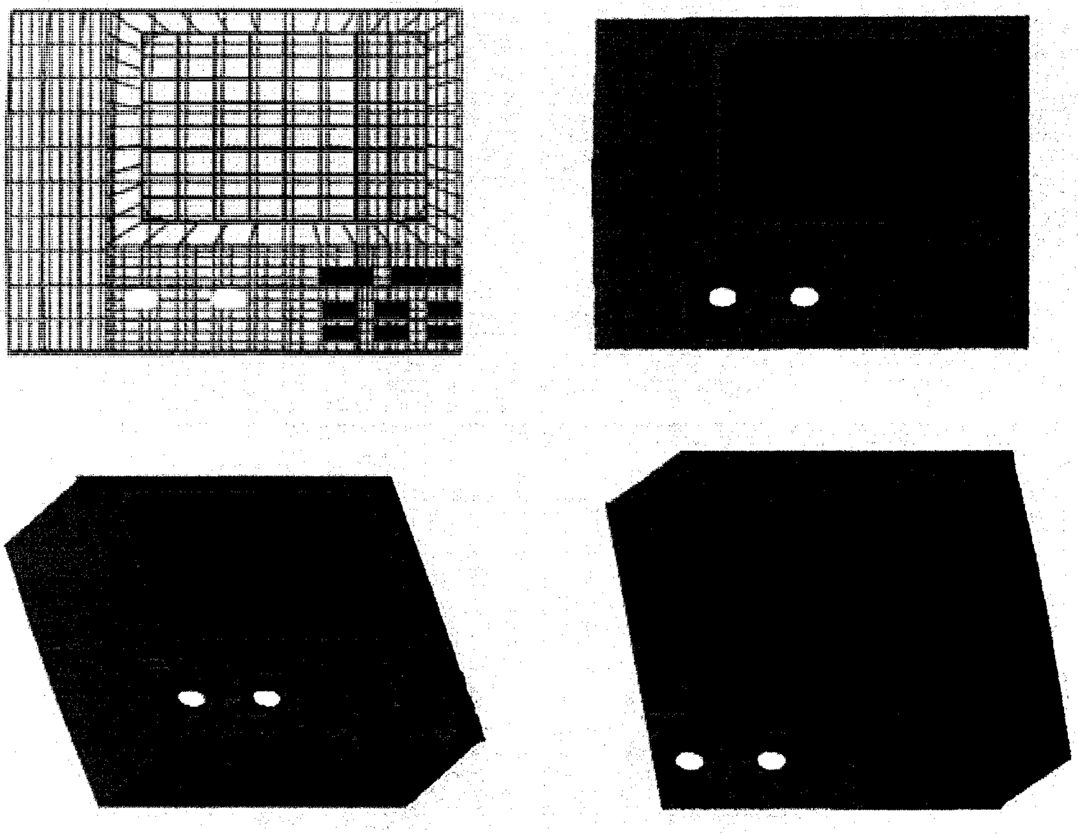


Figure 4.10: Microwave oven design in various views

4.2.4. Chair and table

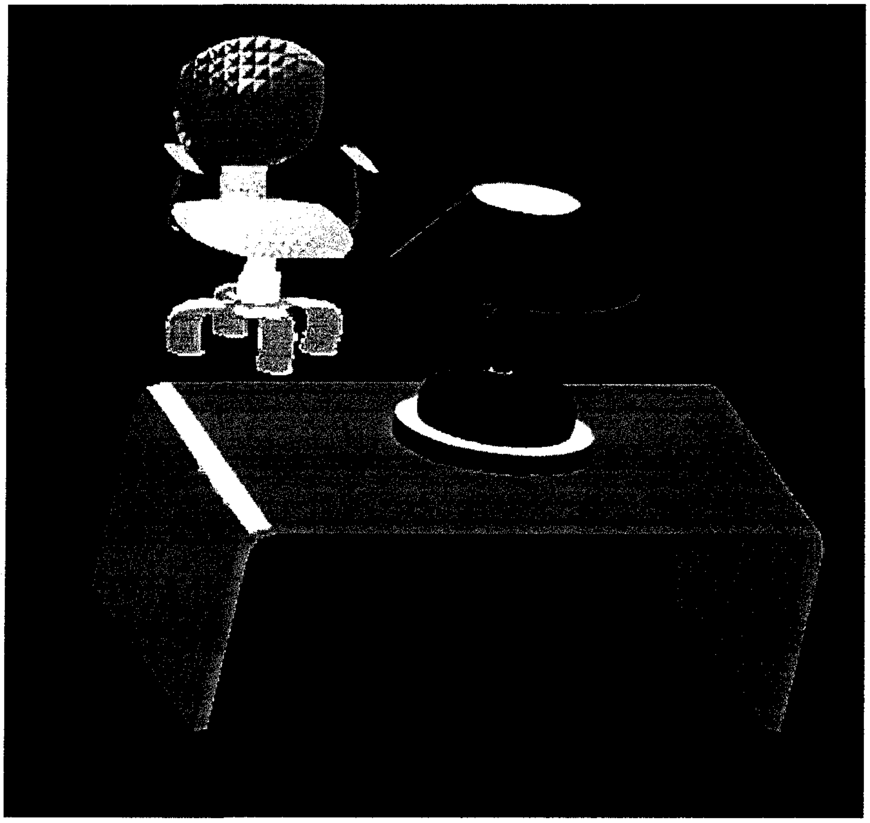


Figure 4.11: Chair and table

4.2.5. A chair and drawing sequence

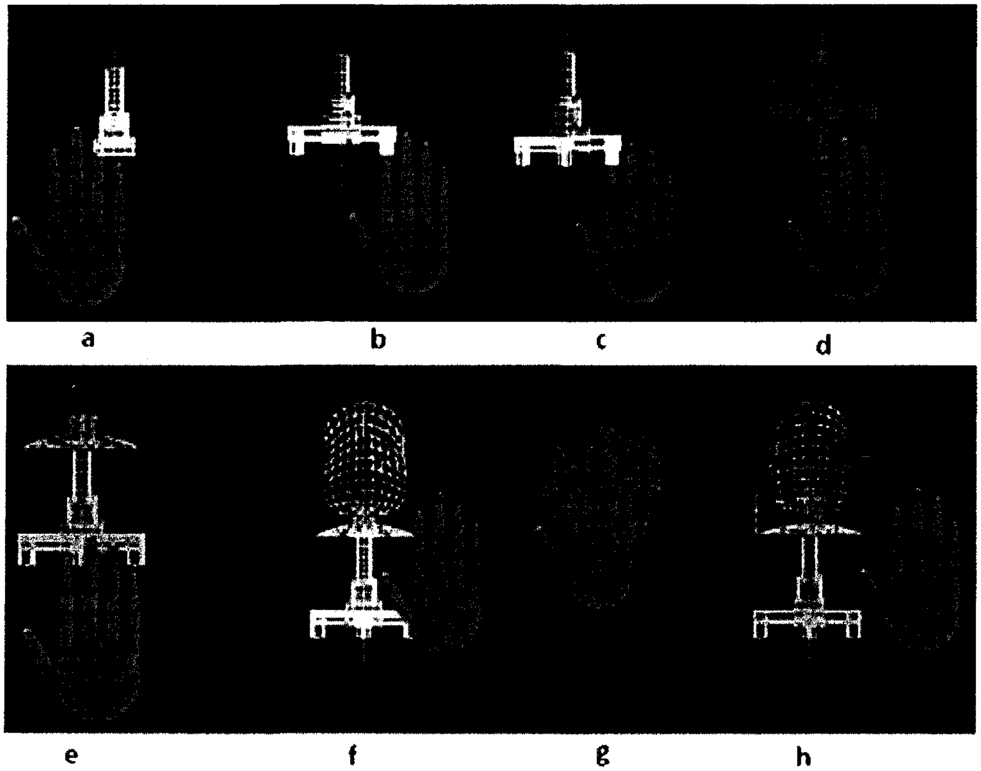


Figure 4.12: a) Designing the chair shaft; **b)** The chair legs' design; **c)** Rotate 90 degree around Y axis to draw the another two legs; **d)** Rotate 90 degree counter clockwise around X axis to draw the pad; **e)** Rotate around X axis 90degree clockwise to prepare to draw the back pad; **f)** Finish the back pad drawing; **g)** Rotate around X axis 90 degree counter clockwise to draw railings; **h)** A completed chair image in wire model .

4.2.6. Solid images of the chair

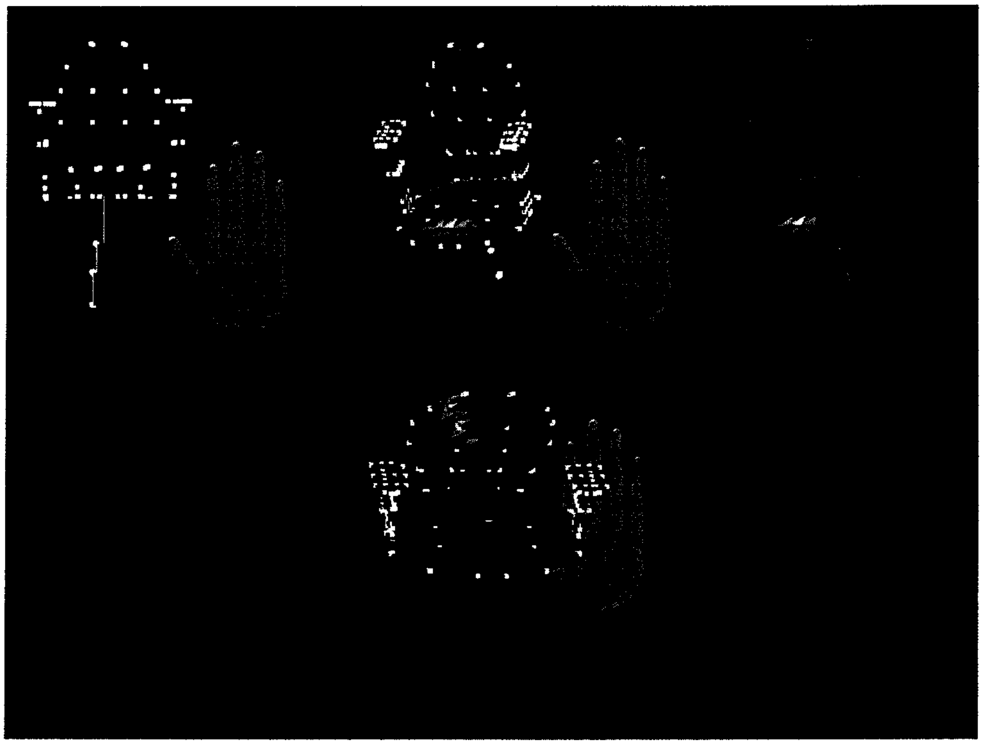


Figure 4.13: Solid images of the chair

Chapter 5

Conclusion and Future works

5.1. Conclusion

This thesis introduced a toolkit that enables designers to perform conceptual design in an intuitive naturalistic environment in 3D space by utilizing the VR technology. In the proposed system, the designers (the users of the VR system) can transform their creative ideas into visible virtual 3D objects and through direct interaction with virtual 3D objects, perform real-time evaluation and correction of the designed model. As part of this thesis we introduced four contributions which are the complementary functions to achieve the desired goal mentioned above.

1. Constructing parametric NURB splines and NURB surfaces in real-time from users' hand gestures. The data can be captured from a single-source (tip of a finger) or multi-source (shape of the finger)
2. Stitching the surface patches and ensuring the C1 continuity along the boundaries between the stitched NURB patches
3. Adding thickness to the arbitrary drawn 3D objects
4. Trimming the existing NURB surfaces

In order to give an accurate portray of the shape of the designed objects, we use data points (or data point cloud) to describe the geometric features of the designed objects. Then we use least-square method (a reverse engineering method to fit data points) to obtain the desired parametric NURB spline or NURB surface that best represents the data captured from users' hand gestures. Finally, we showed how to stitch these NURB patches and NURB splines to obtain visible virtual 3D objects in an accurate and effective way. The connection of the NURB patches requires smooth connection. Thus, we introduced a smooth connection method based on the C1 continuity. The stitching of surface patches is handled in two steps:

1. Fillet NURB surface construction between two joining NURB surfaces
2. NURB surface construction to fill in the holes in more joining NURB patches' case

The developed stitching algorithm help maintain the boundary conditions of the C1 smooth connection along their common boundaries and common end points when connecting surface patches. We obtain the virtual 3D objects with the smooth surface through this method. However, the objects in real world are not only surface patches. They have material properties and thickness information specified by volume. In order to implement such realism in our simulation we further enhanced the toolkit. The proposed free-from sketching technique is the first attempt to give surface thickness to the designed object, which is a parametric surface.

In conclusion, we use VR technology and above mentioned properties systematically to create a design platform where designers/artists can draw a complete 3D object with certain thickness and smooth parametric NURB surface in real-time without the use of assembly process.

In the future work section, we identify several goals that we further like to investigate.

5.2. *Future Work*

We introduced a toolkit to perform free-form conceptual design using VR technology in virtual 3D space. As seen in the examples provided in the chapter 4, the proposed toolkit enables users to sketch their imaginations in a 3D space in details and accurately. The experimental and research work in this area however further revealed shortcomings of the current prototype.

- Creating the immersive environment: The work can be tested in a fully immersive VR environment.
- Drawing the image with 1:1 scale in the immersive environment
- Managing the generated data
- Performing properties test and animation for the designed objects

BIBLIOGRAPHY

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

Akgunduz A and Yu H.: 2005, "Free Form Sketching in 3D Using Uniform Spherical Reference Points", HCI International 2005, July in Las Vegas, USA

Andrew T. Duchnowski: 2002, "Eye Tracking Methodology: Theory and Practice".

Akgunduz A. and Zetu D: 2005, "Shape Recovery and Viewpoint Planning for Reverse Engineering", the International Journal of Advanced Manufacturing Technology, 26(11-12):1370-1378.

Beeker, E.: 1986, "Smoothing of shapes designed with free-form surfaces", computer-Aided Design Volume18, issue 4(May) Pages: 224-232.

Barsky, B.A. and DeRose, T.D.:1990; "Geometric Continuity of Parametric Curves: Constructions of Geometrically Continuous Splines". University of California at Berkeley, University of Washington.

Corbo, P., Germani, M., Mandorli, F.: 2004, Aesthetic and functional analysis for product model validation in reverse engineering applications. Computer Aided Design .36(1):65–74

Cutler,L. Frohlich,B. Hanrahan, P.: 1997, "Two-Handed Direct Manipulation on The Responsive Workbench", Symposium on Interactive 3D Graphics.

Dani, T. H. Gadh, R.: 1997, "Creation of Concept Shape Designs via a virtual Reality Interface", Computer-Aided-Design, Vol. 29.

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

Farin, G.E., Hansford, D.: 2000; "The Essentials of CAGD". Page: 166, 182.

Farin, G.E.:2001; "Curves and Surfaces for CAGD" Page: 330.

Huang, H.B. and Qian, X.P.: "Dynamic B-spline Surface Reconstruction: Closing the Sensing-and-Modeling Loop in Reverse Engineering". Mechanical Materials and Aerospace Engineering, Illinois Institute of Technology, Chicago. IL 60616.

Krause, F.-L. ,Gobel, M., Wesche, G. Biahmou, T.: 2004; "A Three-stage Conceptual Design Process Using Virtual Environments".

Lin, H.W. Bao, H. J.: 2007, "A Practical Adaptive Mesh Fitting Method in Reserves Engineering".

Liu, X. C., Gagh, R. "VirtualSketcher: Skeleton Based 3D Direct Conceptual Shape Design In A Virtual Reality Environment" (unpublished).

Pilnikova, J. Placek, J. Sofranko, J. 1998, Continuity of Bezier Patches Bratislava Slovakia. CESCG.

Pollefeys, M., Koch R., Vergauwen, M., Van Gool, L.: 2000, Automated reconstruction of 3D scenes from sequences of images. ISPRS JJPhotogramm Remote Scenes 55:251–267

Schkolne, S.: 2006, "Making digital shape by hand" International Conference on Computer Graphics and Interactive techniques, Pages: 84-93.

Schkolne, S., Pruett, M. , Schroder, P.: 2001, "Surface drawing: creating organic 3D shapes with the hand and tangible tools", Proceeding of the SIGCHI conference on Human factors in computing systems.

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

Sachs, E; Robert, A; Stoops, D: 1991, "3-Draw a tool for designing 3D shapes". Computer Graphics and Applications, IEEE volume11, issue 6, Nov page(s):18-26.

Tovey, M. Owen, J.: 2000, “Sketching and Direct CAD Modeling in Automotive Design”, 2000 Elsevier Science Ltd, Pages 569-588.

Wesche, G. Droske, M.: 2000, “Conceptual *free*-form styling on the responsive workbench”Virtual Reality Software and Technology, Proceeding of the ACM symposium on virtual reality software and technology, Session: Augmented Reality /3D modelingPage: 83-91.

Wesche, M. Seidel, H. P.: 2001,”FreeDrawer: a free-form sketching system on the responsive workbench”.

Xu, S., Fok, S, C. Tor, S. B.: 2000, “Rapid Creation and Direct Manipulation of Free-form Curves Using Data Glove”, in Proceedings of EDA, Orlando, Florida, USA. PP: 954~959.

Zeid, I.:2005; “Mastering CAD/CAM”, Page: 207~216,325~343,463~498.