An Investigation on the Framework of Dressing Virtual Humans

A thesis submitted for the Doctor of Philosophy

By

Hui Yu

School of Engineering and Design, Brunel University

February 2010

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize the Brunel University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature

I further authorize the Brunel University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Abstract

Realistic human models are widely used in variety of applications. Much research has been carried out on improving realism of virtual humans from various aspects, such as body shapes, hair, and facial expressions and so on. In most occasions, these virtual humans need to wear garments. However, it is time-consuming and tedious to dress a human model using current software packages [Maya2004]. Several methods for dressing virtual humans have been proposed recently [Bourguignon2001, Turquin2004, Turquin2007 and Wang2003B]. The method proposed by Bourguignon et al [Bourguignon2001] can only generate 3D garment contour instead of 3D surface. The method presented by Turquin et al. [Turquin2004, Turquin2007] could generate various kinds of garments from sketches but their garments followed the shape of the body and the side of a garment looked not convincing because of using simple linear interpolation. The method proposed by Wang et al. [Wang2003B] lacked interactivity from users, so users had very limited control on the garment shape.

This thesis proposes a framework for dressing virtual humans to obtain convincing dressing results, which overcomes problems existing in previous papers mentioned above by using non-linear interpolation, level set-based shape modification, feature constraints and so on. Human models used in this thesis are reconstructed from real human body data obtained using a body scanning system. Semantic information is then extracted from human models to assist in generation of 3 dimensional (3D) garments. The proposed framework allows users to dress virtual humans using garment patterns and sketches.

The proposed dressing method is based on semantic virtual humans. A semantic human model is a human body with semantic information represented by certain of structure and body features. The semantic human body is reconstructed from body scanned data from a real human body. After segmenting the human model into six parts some key features are extracted. These key features are used as constraints for garment construction.

Simple 3D garment patterns are generated using the techniques of sweep and offset. To dress a virtual human, users just choose a garment pattern, which is put on the human body at the default position with a default size automatically. Users are allowed to change simple parameters to specify some sizes of a garment by sketching the desired position on the human body.

To enable users to dress virtual humans by their own design styles in an intuitive way, this thesis proposes an approach for garment generation from user-drawn sketches. Users can directly draw sketches around reconstructed human bodies and then generates 3D garments based on user-drawn strokes. Some techniques for generating 3D garments and dressing virtual humans are proposed. The specific focus of the research lies in generation of 3D geometric garments, garment shape modification, local shape modification, garment surface processing and decoration creation.

A sketch-based interface has been developed allowing users to draw garment contour representing the front-view shape of a garment, and the system can generate a 3D geometric garment surface accordingly. To improve realism of a garment surface, this thesis presents three methods as follows. Firstly, the procedure of garment vertices generation takes key body features as constraints. Secondly, an optimisation algorithm is carried out after generation of garment vertices to optimise positions of garment vertices. Finally, some mesh processing schemes are applied to further process the garment surface. Then, an elaborate 3D geometric garment surface can be obtained through this series of processing.

Finally, this thesis proposes some modification and editing methods. The user-drawn sketches are processed into spline curves, which allow users to modify the existing garment shape by dragging the control points into desired positions. This makes it easy for users to obtain a more satisfactory garment shape compared with the existing one. Three decoration tools including a 3D pen, a brush and an embroidery tool, are provided letting users decorate the garment surface by adding some small 3D details such as brand names, symbols and so on. The prototype of the framework is developed using Microsoft Visual Studio C++,OpenGL and GPU programming.

Acknowledgements

I would like to thank all of those who have supported me through my research. In particular, I'd like to express my sincere sense of gratitude to my supervisor, Dr. Shengfeng Qin, for accepting me in the first place. Without whom I would perhaps never have discovered the fascinating field of computer graphics and design. Thank Dr. Qin for his invaluable support, advices, encouragement and guide since I first arrived at Brunel. My grateful thanks are also extended to my second supervisor, Professor David Wright for his invaluable advices and help. I would like to thank Dr. Jinsheng Kang for his suggestions.

This thesis would never have been completed without the support of my family. I would like to thank my wife for her love, support and encouragement in last several years. The birth of my daughter brought me so much sunshine to my life during these busy days of doing the project. Finally, I am deeply grateful to my parents and sisters for their greatest and selfless love and support.

Table of Contents

Chapter 1. Introduction..1

1.1 Overview..1

1.1.1 What Are Virtual Humans?..3

1.1.2 What are Applications of Virtual Humans?..4

1.2 Why Do We Need to Dress Virtual Humans?..7

1.3 Problem statement..9

1.4 Facility and Software..10

1.5 Road Map to the Thesis..11

1.6 Summary..13

Chapter 2. Literature Review..14

2.1 Introduction..14

2.2 Semantic Human Body Modelling..14

2.2.1 Three Dimensional Human Body Modelling..15

2.2.2 Human Body Segmentation..17

2.3 Virtual Garment Design and Human Dressing..19

2.3.1 Standard Methods..20

2.3.2 Sketch-Based Methods..25

2.4 Sketch-Based Techniques..29

2.5 Conclusion..37

Chapter 3. Framework for Dressing Virtual Humans..38

3.1 Motivations..38

3.1.1 Traditional methods for Garment Design..39

3.1.2 Development of Hardware..39

3.1.3 Introduction to the Framework..40

3.1.4 Garment generation from sweep surface..40

3.1.5 Garment generation from offset surface..41

3.1.6 Garment generation from sketching cross sections..41

3.1.7 Garment generation from sketching garment contour..41

Chapter 4. The Sketching Interface for Dressing Virtual Human.. 45
4.1 Introduction..45
4.2 Interface for Garment Pattern Generation...48
4.2.1 Trousers Generation from Sweep and Offset Surface...51
4.2.2 Implementation.. 53
4.3 Interface for Sketching Garment Contour and Processing..57
4.4 Testing Results...63
4.5 Conclusion.. 64
Chapter 5. Semantic Human Body Modelling...65
5.1 Introduction..65
5.2 Data Structure.. 65
5.2.1 Basic Concepts... 65
5.2.2 Topological Entities.. 66
5.3 Human Body Modelling.. 69
5.3.1 Data Pre-processing.. 69
5.3.2 Human Body Segmentation... 70
5.3.3 Point Sampling on Elliptic Cross Sections..80
5.3.4 Point Sampling on Cross Sections in Shoulder Areas...81
5.3.5 Surface Representation...85
5.3.6 Results.. 87
5.4 Body Feature Extraction... 89
5.4.1 Key Feature Extraction.. 90
5.4.2 Reference Skeleton...94
5.4.3 Testing Results... 94
5.4.4 Conclusion.. 95
Chapter 6. Dress Virtual Humans from Sketches...96
6.1 Introduction..96
6.2 Overview of the Dressing Method.. 99
6.3 Garment's Vertices Generation..101

6.3.1 Sketch Recognition 103
6.3.2 Spline Curve Fitting 103
6.3.3 Curve Classification 106
6.3.4 Garment Vertices from Inner Contour Curves 107
6.3.5 Garment Vertices from Outer Contour Curves 107
6.4 Garment Shape Optimisation Processing 108
6.4.1 Normal Calculation of Cross Section 109
6.4.2 Signed Curvature and Curve Energy 110
6.4.3 Shape Processing Based on Level Set Method 111
6.5 Mesh Generation 117
6.6 Garment Mesh Refinement 122
6.6.1 Mesh Smoothing 122
6.6.2 Mesh Subdivision 124
6.7 Shape Modification 126
6.8 Penetration Detection 127
6.9 Decoration Tools 130
6.9.1 Three Dimensional Pen 130
6.9.2 Three Dimensional Brush 131
6.10 Results 133
6.11 Conclusion 137
Chapter 7. Conclusion and Feature Work 138
7.1 Summary of the Thesis 138
7.2 Contributions 140
7.3 Limitations and Future Work 141
References 143
Appendix A Publications 155

List of Figures

Figure 2.1 Human body segmentation... 18
Figure 2.2 Reeb-graph human model representation.. 19
Figure 2.3 Interactive garment design... 20
Figure 2.4 Developing simulation techniques for an interactive clothing system...................... 22
Figure 2.5 Dressing virtual people..23
Figure 2.6 Interaction free dressing virtual humans.. 24
Figure 2.7 Generating unified model for dressed virtual humans...25
Figure 2.8 Feature based 3D garment design through 2D sketches...27
Figure 2.9 Sketching garments for virtual characters.. 28
Figure 2.10 Front and back drawing and garment... 29
Figure 2.11 Teddy: a sketching interface for 3D freeform design...32
Figure 2.12 Towards calligraphic interfaces sketching 3D scenes with gestures....................... 33
Figure 2.13 Sketch-based modeling with few strokes.. 34
Figure 2.14 Modeling developable surface..35
Figure 2.15 Drawing for illustration and annotation in 3D...36
Figure 3.2 The flowchart of dressing from sketches..42
Figure 4.1 Semantic human model generation... 46
Figure 4.2 Example of input data..47
Figure 4.3 Sampling parameter setting... 48
Figure 4.4 Garment generation from offset surface..50
Figure 4.5 Point offset.. 51
Figure 4.6 Trousers from sweep surface...52
Figure 4.7 T-shirt and trousers menu...53
Figure 4.8 Specify garment parameters by sketching...56
Figure 4.9 Main menu and sketch garment dialogue window.. 58
Figure 4.10 Change colour of a garment surface..60
Figure 4.11 Change colour of the garment surface...61

Figure 4.12 Garment modification........62
Figure 4.13 3D decoration........63
Figure 4.14 From sketches to 3D garment surface........63
Figure 5.1 Mesh types........66
Figure 5.2 Hierarchy of these data structure........67
Figure 5.3 Noise removal and hole filling........70
Figure 5.4 Leonardo Da Vinci's Vitruvian Man........71
Figure 5.5 Eight-head-tall-body representation........72
Figure 5.6 Model projection with head feature region........73
Figure 5.7 Four quadrants slice with a centroid........77
Figure 5.8 Crotch slice with two identified feature regions........78
Figure 5.9 Body segmentation........79
Figure 5.10 Point sampling on cross sections........81
Figure 5.11 Calculation of cross section centroids........84
Figure 5.12 Model simplification........85
Figure 5.13 Cross section connection........86
Figure 5.14 Mesh of body parts........87
Figure 5.15 Human models with different LODs........88
Figure 5.16 Semantic human model........89
Figure 5.17 Projection of a human model........90
Figure 5.18 Feature girth calculation........91
Figure 5.19 Human models with key body features........93
Figure 5.20 A skeleton model........94
Figure 5.21 Human models with key body features........95
Figure 6.1 Flowchart of garment construction by sketching profiles........97
Figure 6.2 An example of a sweater generation........100
Figure 6.3 Contour curves and garment vertices........102
Figure 6.4 Spline curve fitting........104
Figure 6.5 Garment curve generation using the level set method........115

Figure 6.6 Garment tension mimic using the level set method....117
Figure 6.7 Triangulation....118
Figure 6.8 Part sewing....120
Figure 6.9 Examples of mesh sewing....121
Figure 6.10 Point position modification....123
Figure 6.11 Loop subdivision....124
Figure 6.12 An example of subdivision....126
Figure 6.13 Examples of garment shape modification....127
Figure 6.14 Examples of penetration detection and response....130
Figure 6.15 Illustration of pen drawing....131
Figure 6.16 Illustration of three dimensional embroidery tool....132
Figure 6.17 Illustration of embroidery tool....133
Figure 6.18 An example of a strap skirt....134
Figure 6.19 Demonstration of using decoration tools....136

List of Tables

Table 1 Comparison of application and requirements for building virtual humans...................... 5
Table 2 Stroke gestures...28
Table 3 Key body feature explanation... 90

Chapter 1. Introduction

1.1 Overview

This thesis proposes a prototype framework for dressing three dimensional virtual humans. Three dimensional virtual humans get increasingly popular in various applications such as computer games, cartoon movies, virtual reality and so on. High demands for fidelity and realism of the virtual characters require the rapid improvement both in hardware and software of computer graphics and animation design techniques. Recently, the development of body scanning systems and motion capture equipments make it possible to produce virtual humans based on real human body data and motions, which highly improve the realism of virtual world. However, it is time consuming for non-professional users to dress virtual humans [Maya2004]. This poses a difficulty for the applications of virtual humans in different areas.

Most previous studies investigated virtual human dressing methods based on the traditional method for dressing garments for real people. The traditional method for dressing virtual humans usually takes two stages as follows:

Stage 1: Tailoring -- create 2D pattern pieces;

Stage 2: Dressing -- put pattern pieces on a human body and sew them together.

Usually, to dress a virtual human, designers must go through the two stages. In the first stage, designers have to obtain the measurements of the virtual human and design pieces according to the whole garment shape. And then in the second stage, designers need to put the pattern pieces one by one and sew them together after properly position them on the body. If designers are not satisfied with the dressing result and want to modify the garment, they have to repeat the same process. This process usually requires some expertise or good understanding of garment design from the designers.

Over last years, noteworthy efforts have been made to improve dressing procedure and effects [Thalmann1997, Fuhrmann03]. The limitation of above mentioned methods is that users are only allowed to choose garment patterns from the database or restricted to some predefined patterns, which prevent users from dressing virtual humans by their own design. This requires for developing more effective and intuitive methods to give users or designers more flexibility in the garment design and dressing process.

This thesis investigates a framework for intuitively dressing virtual humans (Figure 1.1). The target application of the proposed framework is the entertainment industry such as computer games and cartoon movies. The objective of the framework is to fast dress 3D virtual humans. This objective requires both realistic human bodies and an intuitive and an easy-to-use interface. The human models used in this thesis are reconstructed from real human bodies obtained using a body scanning system. A sketch-based interface is developed to allow users to directly draw garment profiles on the front view of human bodies. Garments represented by geometric surfaces are generated and dressed on these human models constrained by key body features.

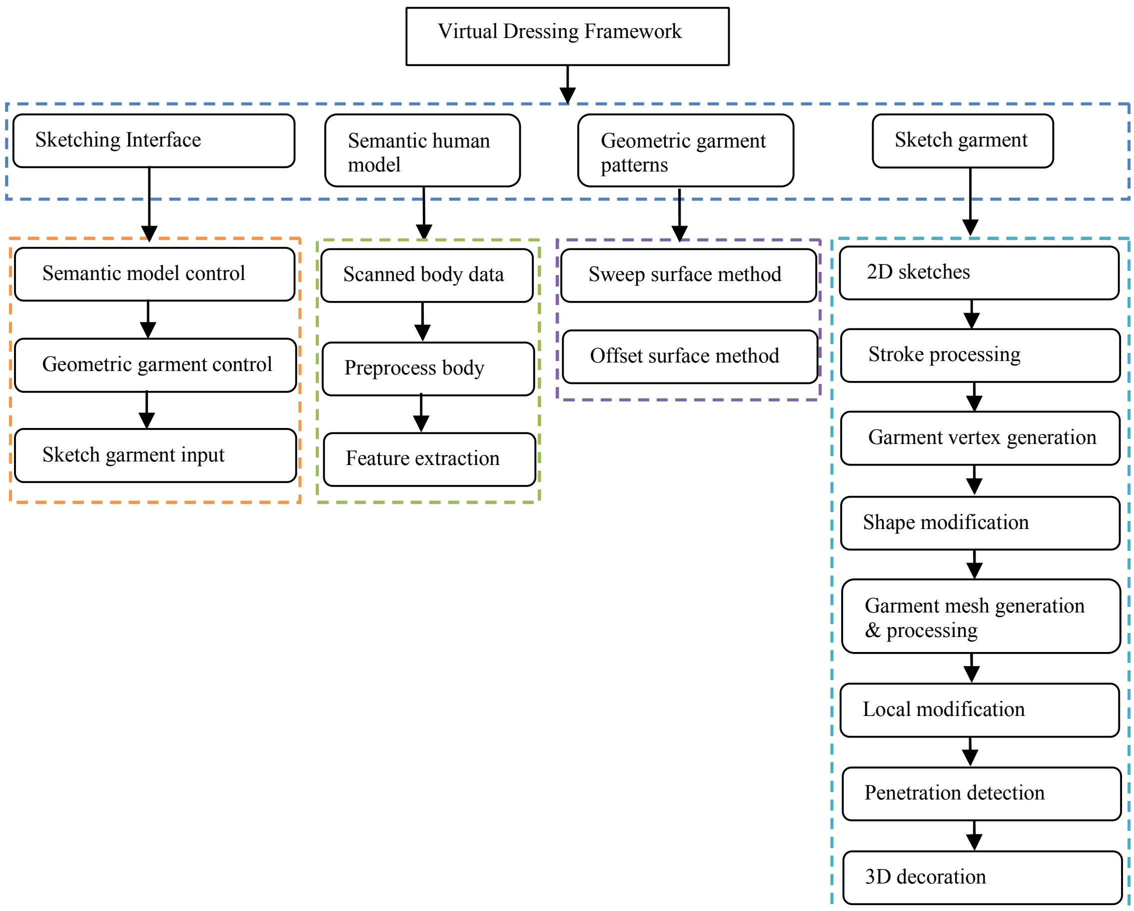


Figure 1.1Framework demonstrationVirtual Humans

1.1.1 What Are Virtual Humans?

Virtual humans are computer models of human being used to substitute real people in virtual worlds. Cavazza and Thalmann [Cavazza1998] classified virtual human characters into the following categories:

Pure avatars or clones: A virtual human figure representation generated and controlled by computer programs. These virtual actors have natural-looking bodies and faces and their animation correlate to actual bodies and faces.

Guided avatars: A virtual human figure interactively controlled by live participants, that are usually portrayed visually as 2D icons or 3D shapes, or full 3D bodies [Badler1993, Stansfield1994 and Robertson1996]. Guided actors do not directly correspond to users' actions.

Autonomous Actors: A virtual human character with intelligence and personality. Autonomous actors are able to have their own behaviour and can perceive the virtual environment such as objects and other virtual characters around her or him through sensors. The autonomous actor may interact with the environment based on the perceived information such as playing tennis [Peters2003, Hill2000].

Interactive-perceptive Actors: they are autonomous avatars with more features such as the ability to communicate interactively with other actors and real people [Emering1998].

1.1.2 What are Applications of Virtual Humans?

The last few years have seen rapidly increasing demands for 3D virtual humans used in various applications including computer games [Rockstar], Engineering analysis and simulation [Demirel2006], virtual reality [Pelechano2008,Thalmann1995], virtual storytelling and entertainment [Göbel 2004, Yu2006], virtual education [Ryokai2003, Parsons2008], cartoon movies [Savant2001] and virtual design [Lämkull2007, Weidlich2009] etc. These applications accelerate the development of virtual humans in terms of human body appearance and motion, especially with the introduction of some advanced devices such as motion capture systems [Animazoo, Vicon]and body scanning systems [Hamamatsu, Textile].

Virtual humans are essential in computer games and cartoon movies. Much effort has been taken to enhance the realism of virtual humans, especially the details of human bodies such as hair and faces. One of the important problems is that these virtual humans need to wear garments.

In addition, the clothing industry will in the future benefit from the existence of dressing systems that allow customers to have their own 3D human models, and use it to virtually try on garments by their own design, whereby customers can see exactly how the clothes will fit on them.

Here is a summary of major applications of virtual humans by Badler [Badler1997A] as follows:

Engineering and simulation: analysis and simulation for virtual prototyping and simulation-based design.

Virtual Conferencing: virtual humans are used as virtual representations of participants in efficient tele-conferencing.

Virtual Environments: agents or avatars live in a virtual place for visualisation, analysis or interaction.

Interaction: Real-time graphical bodies inhabiting virtual worlds.

Monitoring: Acquiring, interpreting, and understanding shape and motion data on human movement, performance, activities, or intent.

Games: animated real-time characters with personality.

Training: Skill development, team coordination and decision-making.

Education: Distance mentoring, interactive, assistance, and personalised instruction.

Military: Battlefield simulation with individual participants, team training and peace-keeping operations.

Design/Maintenance: Design for access, ease of repair, safety, tool clearance and visibility and so on.

Badler thought that virtual fidelity of virtual humans was application dependent [Badler1997A]. He argued that fidelity to human size, capabilities and joint and strength limits were indispensible to some applications such as design evaluation. While in computer games, training and military simulations, temporal fidelity such as real-time behaviour is essential. He characterized the state of virtual human modelling to three aspects:

Visual: Cartoon shape; Functional: Cartoon actions; Cognitive: One-shot animation.

Different capabilities or requirements for building virtual human bodies are needed in terms of various applications shown in Table 1 [Badler1997B].

Table 1 Comparison of application and requirements for building virtual humans

Application	Appearance	Function	Time	Autonomy	Individuality
Cartoons	high	low	high	low	High
Games	high	low	low	medium	Medium
Special Effects	high	low	high	low	Medium
Medical	high	high	medium	medium	Medium

Ergonomics	medium	high	medium	medium	low
Education	medium	low	low	medium	Medium
Tutoring	medium	low	medium	high	Low
Military	medium	medium	low	medium	Low

From the above table, we can see that the requirement for appearance of virtual humans is higher than the rest aspects. Especially, for cartoons and games, fidelity of appearance becomes more important.

1.2 Why Do We Need to Dress Virtual Humans?

In applications, virtual cloth of a virtual human is often substituted by cloth-like texture to avoid presenting visually a naked virtual human body. That is, texture is often directly mapped onto human bodies to represent cloth in virtual environments (Figure 1.2(a)). But it is obvious that a 3D human body with cloth represented by textures is far from realistic. Figure 1.2 (a) shows a virtual human with texture mapping as garments and Figure 1.2 (b) shows a virtual human wearing 3D garments. The main reason is that creation of 3D virtual cloth for virtual human is tedious, expensive and time-consuming using current software packages [Maya2004]. Usually, it needs skilful designers to obtain desirable results.

(a)

(b)

(a) A virtual human with garments represented by textures [Lämkull2007]. (b) A virtual human wearing a 3D garment [Cordier2003].

Figure 1.2Virtual humans with different kinds of cloth representation

With the development of technologies such as body scanning system and motion capture system, we can obtain realistic human body from real human body and motions constructed from the real human motions. However, it is very difficult to obtain pleasing scan data of a human wearing cloth currently. Furthermore, it is impossible to obtain all kinds of virtual garments by scanning real garments. Therefore, high demands for realistic 3D virtual humans require a quick and easy way to dress these human bodies.

The main challenging problem with dressing virtual humans is that garment shapes are not only complex but also diversified, which make it difficult to intuitively design garments according to the body shape and then put it on the body accordingly. Let us take a look at how garments are designed using the traditional method in real life. Designers need to generate garment pieces according to 2D drawing. After these pieces of garment patterns have been completed, they have to be sewn together accordingly. Designers cannot view the final result of a garment until it is put on a human model after sewing up. The traditional pipeline for garment design restricts flexibility of garment style development. The straightforward method for dressing virtual humans is to separate the garment design and dressing procedure [Cordier2002, Vassilev2000].

Human-like realistic virtual human characters are developed in terms of the surface, articulation structures, and motions and so on. This thesis focuses on improving the exterior appearance virtual humans. Applications of the proposed framework mainly target computer games and cartoon movies.

1.3 Problem statement

The first problem is that: is it possible to combine the procedure of garment design and virtual dressing together to enable designers to view their design and dressing results simultaneously? The answer is definite with help of an intuitive sketching interface. When a user or designer wants to describe a 3D object conceived in their mind, they usually express their thoughts by sketching them on paper, which are mainly 2D projection of the 3D object. The sketching interface can record the projection of a garment shape. Sketching Strokes need to be processed and associated with the human model. This raises the second research problem, how to associate sketching strokes with a human model.

The third problem is about determining the third dimension of a garment according to 2D garment contour strokes and a human model. This is one of the key problems for 3D garment construction. Varied garment shapes add complexity for 3D construction, as users cannot fix the same z-values for various styles of garments. In addition, a garment should be worn by a human model realistically after construction, so some reference positions, which are usually some key body features, should be determined during the procedure of garment construction.

The fourth research problem is to use some geometric methods to process garment surfaces. A garment shape should be optimised after constructing a coarse 3D garment shape from 2D sketches. One the one hand, a garment shape can be optimised to enhance its realism. On the other hand, a garment surface can be improved to obtain a more elaborate mesh representation. In addition, users might want to edit the garment surface or add details to the surface, so a more elaborate mesh surface is necessary.

The last research problem in this thesis is about functions for editing a garment surface or adding details to a garment surface. It is still challenging to draw 3D details directly on a 3D surface, especially on a garment surface with a complex shape.

1.4 Facility and Software

[TC]^2 body scanning system [Textile]

We use [TC]^2 body scanning system (Figure 1.3) to obtain real human body data.

Figure 1.3 [TC]^2 body scanning system

Geomagic Studio software [Geomagic]

We use Geomagic Studio software for pre-processing scanned human body data such as noise removal, hole filling and so on.

Microsoft Visual Studio Compiler, C++ STL

Our framework is built using Microsoft Visual Studio, C++ STL.

OpenGL Graphics Library [OpenGL]

OpenGL graphics library is used for rendering and visualization.

1.5 Research Output

Sketching interface

A sketching interface has been created for dressing virtual humans using Microsoft Visual Studio C++. Users can specify garment parameters and 2D garment contour using 2D sketches.

Method for 3D garment vertex generation

A non-linear interpolation method with feature constraints have been proposed for generating garment vertices from outer curves, which make garment curve look more convincing.

Method for garment shape modification

The level set method is applied to garment curves for shape modification which can prevent the garment shape from exactly following the body shape.

Methods for 3D garment pattern generation

Two methods for garment pattern generation: sweep surface method and offset surface method. Users can change some sizes of garments by sketching on the screen.

Local modification and decoration tools

Users are allowed to locally modify a garment shape due to the property of spline curve. And three decoration tools are developed for detail decoration.

1.6 Road Map to the Thesis

The reminder of the thesis is organised as follows:

Chapter 2 provides a literature review of the related techniques and methods that have been developed in recentyears. Firstly, the evolution of human body reconstruction is reviewed. A brief introduction of human model segmentation and feature extraction is conducted, which is the key problem of a semantic human model. Secondly, related work in virtual dressing and garment design using CAD systems is presented. Finally, the development and related work of sketching interfaces are reviewed.

Chapter 3 presents an overview of the functionalities of the dressing framework. This dressing framework integrated three parts: a sketches interface, semantic human modelling and garment generation methods and dressing techniques. This chapter details the integration of the three aspects. To illustrate the work flow of the system, the main functionalities of the application are outline as a storyboard. Two garment generation and dressing methods are introduced. Firstly, a method for generating tight garments such as tight trousers and T-shirt using offset surface is introduced. Secondly, a sweep surface method for trousers generation is outlined, which is used to generate trousers combining with the offset method. The garments generated by the above mentioned two methods are prepositioned on human models based on human body features. To change the position or some sizes of a garment, users only need to specify them by sketching on the human body.

Chapter 4 introduces the sketching interface of the dressing framework. This chapter first gives an overview of the sketching interface. And then the interface for garment pattern generation is presented, which briefly introduces how to dress human bodies from garment patterns. Finally, the interface for garment generation from sketching contour, which includes sketching garment contour and processing decoration tools and so on.

Chapter 5 describes the procedure of generating a semantic human model which is the reference model for proposed virtual dressing methods. A semantic human model is a human body with extracted semantic information represented by body features, skeleton and so on. The raw data of human models we use is obtained from a $[TC]^2$ body scanner. Once a human model is reconstructed, some key body features which are used as constraints during the procedure of 3D garments generation and virtual dressing. These key body features include key feature points on the body surface and feature curves.

Chapter 6 describes an approach for dressing virtual humans by sketching garment profile strokes. Firstly, a stroke processing method is introduced, which processes sketching strokes into parametric curves. Secondly, garment vertices interpolation algorithms are presented, which are followed by a level set-based optimisation method. Thirdly, methods for mesh construction and optimisation are discussed and implemented to obtain detailed garment mesh. Finally, a penetration detection and response algorithm is developed to avoid mesh surface penetration. Three decoration tools including a 3D pen, a 3D brush and a 3D embroidery tool are demonstrated. Decoration is represented by curves or mesh surfaces. Details of these curves and mesh are introduced accordingly.

Chapter 7 draws conclusions of the thesis and discusses future work.

1.7 Summary

In this chapter, we first address motivation, problem definition and objectives of the research presented in this thesis. And then, we give a brief introduction of the methodology. The organisation of the thesis is outlined after a brief summary of the contribution of the research.

Chapter 2. Literature Review

2.1 Introduction

The problem of dressing virtual humans based on sketches has only drawn attention recently, but related research in virtual human modelling and garment design has been carried out for many years. This chapter reviews research related to virtual human dressing. Our method for dressing virtual humans is based on sketches, which includes three aspects: semantic human models, virtual dressing, and a sketch interface. Virtual dressing actually in this thesis includes two aspects: virtual garment generation and clothing virtual humans. In this thesis, these two aspects are integrated together. After a garment completes, it is put on the human model automatically. Most of previous work completed these two steps separately. In the following subsections, we review research respectively in these areas. We firstly give a general overview of the state of the art of 3D geometric modelling with particular attention to human modelling. And then, we present a review of previous related research on each specific research topic covered in this thesis.

2.2 Semantic Human Body Modelling

The last decade has seen a growing interest in human modelling with a large spectrum of potential applications. Different applications require different semantic information.

The problem of human modelling has been of great interest to many fields such as computer games, biomechanics, ergonomics, anthropology, industrial design, virtual reality and garment design and so on. A variety of human body modelling methods have been developed, which are subject to three major categories: creative, reconstructive, and interpolated [Thalmann2004].

Our human models, reconstructed from body scan data, are 3D geometric models. We mainly review models fall in the reconstructive category.

Our semantic human model construction consists of two stages including human model reconstruction and features extraction. Human bodies we used are 3D geometric human models reconstructed from body scan data. Then the human model is segmented into six parts and built into continuous mesh representation model by part connection. Semantic human features are extracted from reconstructed mesh models.

2.2.1 Three Dimensional Human Body Modelling

A large variety of 3D methods for human modelling have been proposed over last 20 years. These methods can be classified into three categories: anatomical human body modelling, multi-layer body modelling and human body surface modelling. Anatomical human body modelling method try to model virtual human body following the anatomical tissue properties and multi-layer modelling method uses simple geometry to represent human body. In this thesis we use human bodies reconstructed from real human data. Here we just give a brief review of methods for human body modelling.

In the early stage of human modelling, researchers used simple geometry to represent a human model [Badler1979]. The 3D human model included two structures: one was a skeleton of joints and connecting segments, the other was a skin, which was represented by overlapping spheres. Magnenat-Thalmann et al. created a polygonal surface human model using plaster modelling [Thalmann1987].

A multi-layer model consists of several layers such as a skeleton layer usually used to animate the human model, a skin layer represented the surface of the human model and an intermediate layer used to simulate body tissues such as fat, muscles and bones and so on. Chadwick et al [Chadwick1989] presented a method for modelling animated human model based on Free-Form deformation, which was mapped to the underlying skeleton to simulate muscles and fatty tissue.

To improve realism of human models, anatomical human modelling methods were created based on multi-layer techniques. Anatomical methods insist to create individual muscles, tissues and skeleton in terms of actual human model structure. Nedel et al proposed a three layered method for anatomically modelling human body [Nedel1998]. Three steps were involved with each for one layer in generating a human model. The skeleton represented the rigid body conception, the muscle design and deformation based on physical concepts and a geometric surface representing the skin. The disadvantage of this method is that it involves huge amount of computation for generating details of a human body.

Recent advances in three-dimensional scanning technology have enabled creation of high-density point data sets representing the surfaces of real human bodies [Daanen1998, Hamamatsu , Textile]. With the help of this kind of technologies, we can obtain computer-based geometric human models that describe the topology and geometry of a real human body.

Wang et al. [Wang2003A] proposed a feature based approach of constructing human models from laser scan data. That method used horizontal slices to analyse body data. First of all, noisy data are filtered and orientated. And then, key body features were determined, which were used to segment a human model into six parts. Each part of the point cloud human model is approximated by a stack of horizontal polygon curves using a polar decomposition method. Polygonal mesh was constructed by sewing up the neighbouring contour curve data. Through connecting all six parts, a continuous mesh surface of a human model was constructed. Finally, some key body features on a human model including armhole, crotch, neck, chest and belly button were extracted from the mesh model. This method provided a simple idea on reconstructing human model by sewing up body segments. It only extracted some fixed and general key body features for cloth generation.

Allen et al. [Allen2003] proposed a method for building whole-body morphable model based on 3D body scan data by fitting high-resolution template meshes to detailed human scan data with 3D markers. To approximate an object by the template mesh, an objective function was formulated, which was a weighted combination of the following three measures: proximity of transformed vertices to the range data, similarity between neighboring transformations and proximity of sparse markers at corresponding locations on the template and target surface. A set of error functions such as data error, smoothness error and marker error were used to evaluate the quality of the approximation. This method required a set of feature markers to initialize the registration. An assumption for the method was that the pose of the template needed to be similar to the target surface.

2.2.2 Human Body Segmentation

Zhong et al. [Zhong2006] introduced some methods for automatically segmenting and measuring human bodies. Human models obtained using this method were used for 3D garment try-on. Apart from some segmentation algorithms, some fundamental measuring techniques were also introduced. It used two stages to segment a human body. In the first stage, it located target zones on a human body using the proportion of head length to the body height (Figure 2.1). In the second stage, key body points are found within each target zone. Three methods including minimum distance, minimum inclination angle and directional neighbor are proposed to search branching points or triangle for key points positioning. Four steps were conducted to obtain body segmentation in each target zone. Firstly, it used two methods to find the crotch for contacted thigh and non-contacted thigh respectively. Secondly, acromion and armpits were determined to separate arms from the torso. Thirdly, segment legs from the torso using the crotch point. Finally, neck was segmented from the torso using the neck base point, which was identified by using minimum inclination angle algorithm.

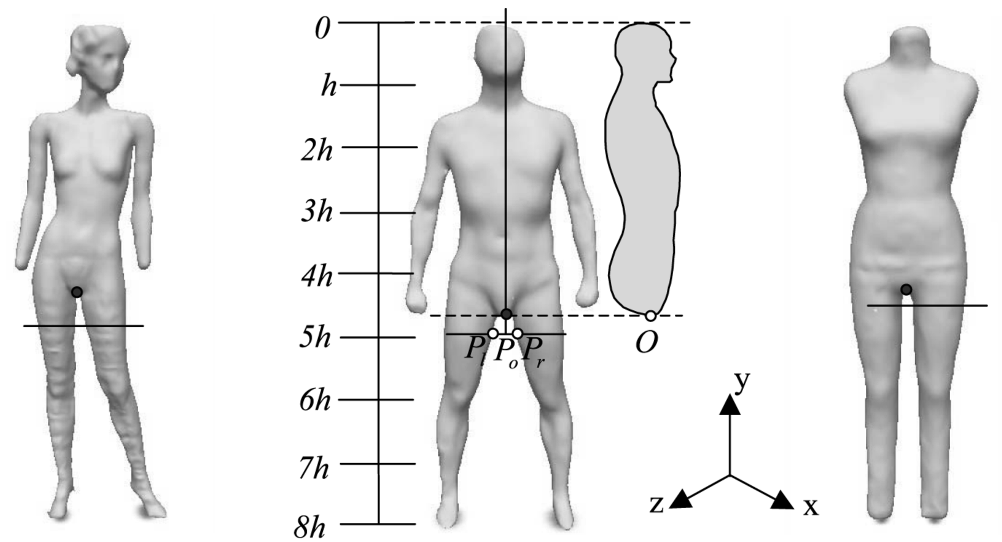


Figure 2.1 Human body segmentation

Xiao et al. [Xiao2003] presented a computational topology framework for segmenting human body (Figure 2.2). This framework built a graphical representation, Reeb-graph, to encode the body scan data and key body points representing key body landmarks such as armpits and the torso. This method could segment human bodies in arbitrary postures without referring the any detailed human heuristics. It segmented a human model into six parts: chest, torso, two arms and two legs.

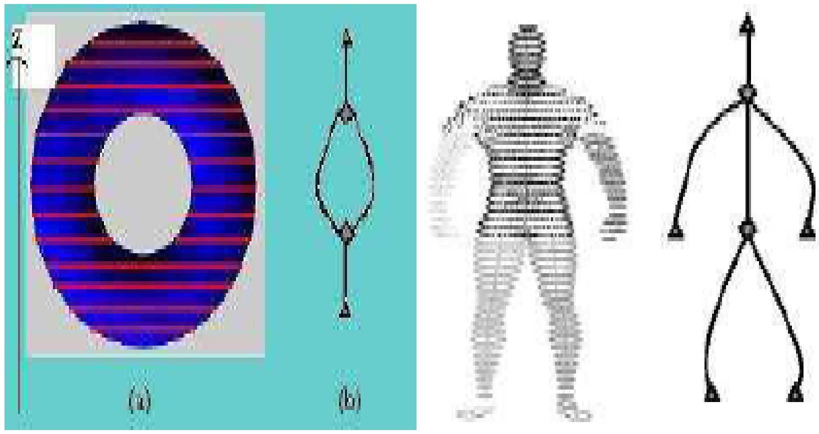


Figure 2.2 Reeb-graph human model representation

2.3 Virtual Garment Design and Human Dressing

Our method introduced in this thesis borrows some ideas from virtual garment design, so we give a brief review of some related methods for tailoring and virtual garment design as well as methods for virtual human dressing. Much research work has been carried out in virtual garment design. Most of current methods emulate the process of garment creation in real life based on 2D patterns as follows:

Garment patterns are designed and cut in 2D space;

Garment patterns are placed around a virtual human model in 3D space and sewn together to obtain complete clothing;

The development of computer technology provides an intuitive way for users to explore 3D spaces through a 2D interface. Sketched based methods were proposed recently. We briefly review the current sketch-based methods for virtual dressing.

2.3.1 Standard Methods

The standard method for garment design is to sew up 2D patterns, the sizes of which are obtained based on measurements of a human model.

Interactive dressing virtual humans

Hinds [Hinds1990] developed a method for interactive garment design using a static trunk of a mannequin's body, which is represented by bicubic B-spline surfaces (Figure 2.3). Garment surface was considered as an assembly of three dimensional surface panels represented by mathematical models. A garment panel was generated as a surface offset from the body by various distances. Garment panels were defined as surfaces according to the shape of the reference body. This method explored a new idea for generating geometric garment panels from surface offset, but did not take consideration of the use of body features.

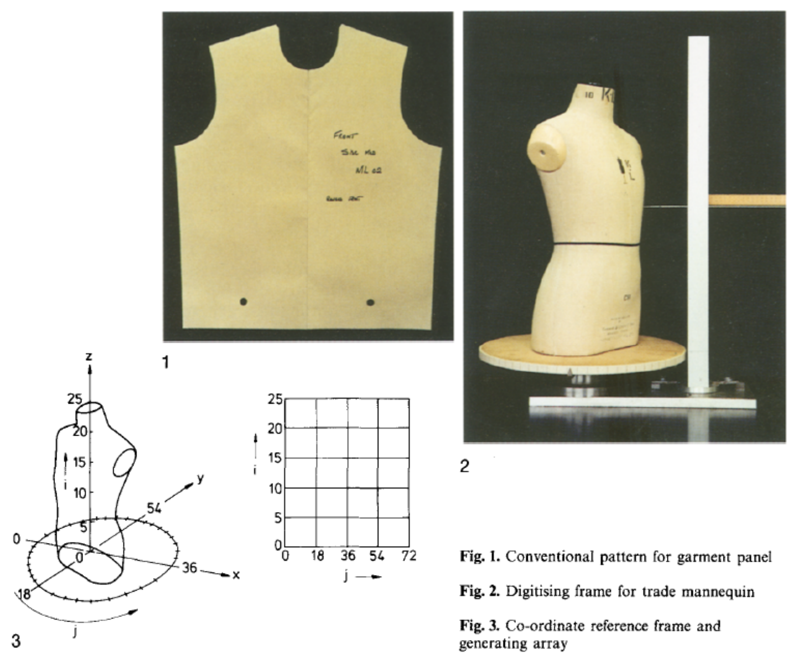


Figure 2.3 Interactive garment design

Fashioniser platform was developed by Volino in MIRALab [Volino1997], which was designed to interactively generate and simulate realistic garments. For each garment, 2D patterns are designed individually. Exact measurements of these 2D patterns are taken and drawn on 2D sketches, which are recreated by Fashioniser platform. Fashioniser platform can generate and simulate various kinds of garments with high realism (Figure 2.4). However, the main problem is that users have to imagine the shapes of the 2D patterns of the garments to be designed, which is difficult for ordinary users. Especially when design some garments with complex shapes, the cuts of garments are usually too complicated. In addition, users need to find the garment size for given measurements of a 3D human model based on individual garment patterns.

(a)

(b)

Figure 2.4 Developing simulation techniques for an interactive clothing system

Vassileve developed a fast technique for dressing virtual humans [Vassilev2000], which was similar to Fashioniser platform (Figure 2.5). Garments were constructed by seaming garment patterns around a 3D human body. Garment patterns were created by drawing packages and exported to the dressing system. Seaming information, such as lines to be seemed together, needed to be specified by users through typing in the file of these garment patterns.

Figure 2.5 Dressing virtual people

Interaction-free dressing

Fuhrmann et al. developed an interaction-free method for geometric pre-positioning of virtual cloth patterns on a 3D human model [Fuhrmann03] (Figure 2.6). It automatically dressed virtual humans with physical-based sewing simulation. The human model from body scan data was segmented six parts for wrapping garment pieces around it. This method sped up dressing procedure by eliminating time for interactive positioning garment patterns. Some feature points are marked manually on the body to help pre-positioning garments. Each pattern was assigned to a body segment after determining the orientation of the pattern. This method dressed virtual humans by providing two kinds of information: seaming information and garment position information. Seaming information including the boundary curves of patterns and connection information of patterns, such as the seams, was stored in standard databases. To move the patterns to the desired positions, it took a general approach independent from the classification of clothing, which was realised through relative positions with respect to the human body. The relative positions can be quite similar for different kinds of clothing, such as that a shirt has the same information with a jacket. It used a classification of segments and assigned the corresponding garment patterns to the body segments. Garment pattern data used in this data was from a clothing database, which prevent it from dressing various styles of garments on virtual humans.

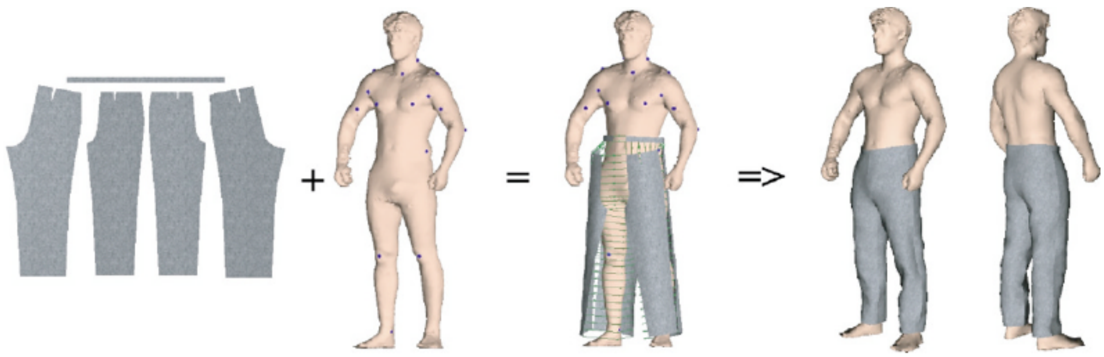


Figure 2.6 Interaction free dressing virtual humans

Kim et al. [Kim2005] proposed two opposite methods for automatic dressing virtual humans: fit the body onto a garment and fit a garment onto the human body (Figure 2.7). These two methods differ from methods introduced above by dressing human models with a complete garment instead of garment pieces. The garment mesh and human model were predefined and exported from some other packages. Garments were dressed on the human model using correspondence between the human model and the garment meshes, which was established using two opposite methods. The first fit-body-onto-garment method fit the body onto a garment by using border constraints and position-to-position constraints. The limitation of the first method was that it can only be used when the garment and the human body mesh have similar topologies. Thus, it was difficult to dress the human model with a skirt using this method. In the second method, the human body model was divided into joint segments according to skinning weights. This fitting supported multiple mappings, which enabled garment vertices to map to several positions on the human body. This advantage allowed dressing skirts on a human body with different topologies. A distance field from the joint segments to the garment surface was constructed, which was optimised by deforming the garment mesh. This method addressed the problem of how to dress a human model. Users dressed a human model using predefined garment mesh generated by a third party software package instead of creating garments by users themselves.

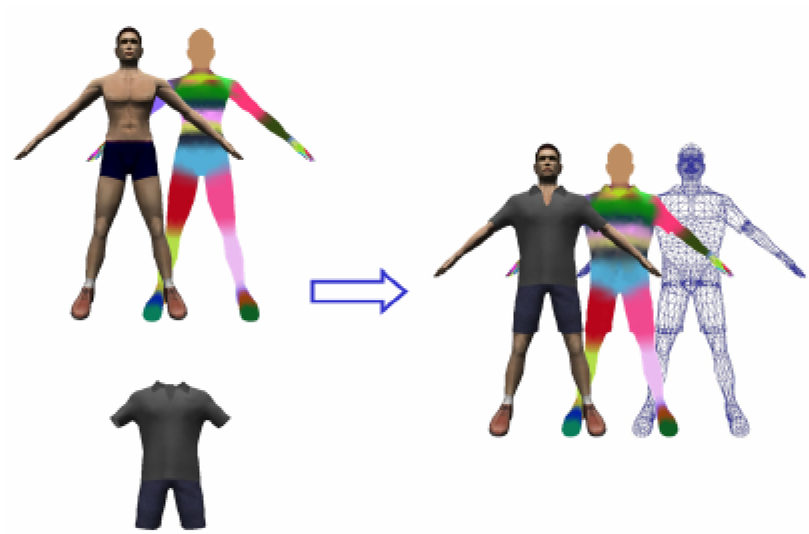
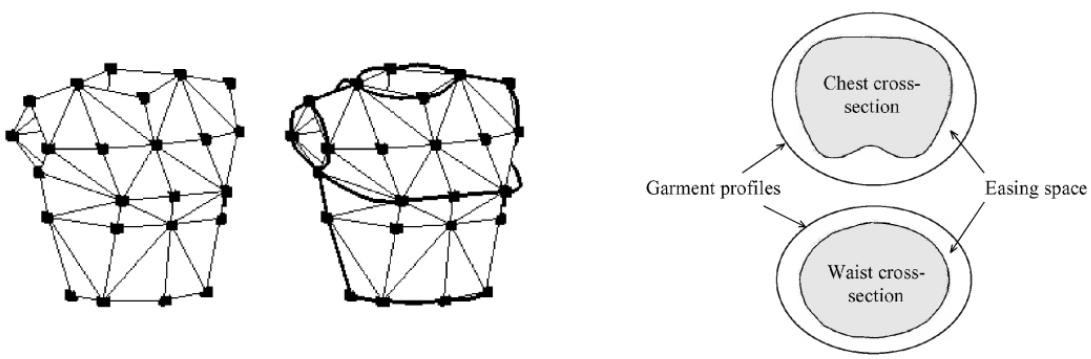


Figure 2.7 Generating unified model for dressed virtual humans

2.3.2 Sketch-Based Methods

Sketch-based dressing virtual humans have been gained attention recently. Only several papers related to sketch-based virtual dressing have been published.

Wang [Wang2003B] developed a 3D garment design system using feature template for 3D garments generation (Figure 2.8). This method could generate a complete garment surface dressed on a human model. The human model was obtained from a body scanning system. Body features were extracted to provide semantic information for garment information. Each semantic feature curve included a sorted set of line segments lying on the mesh surface of a human model. The line segments corresponding to one semantic feature curve have the same feature ID number. And each semantic feature point was an insertion of two semantic feature curves. Thus, line segments were easily determined by feature points. A coarse 3D garment frame was constructed based on the predefined features on human models. The profiles of the 3D garment frame were optimised through 2D sketches by users. These 2D strokes were converted into 3D curves using a projection plane, which approximately passed through the selected edges on the human body. It applied a modified variational subdivision schemes to interpolate the specified profiles. The 3D garment surface was refined with fixed boundary condition. It also applied a collision detection approach to avoid any collision between the garment surfaces and the human body. Finally, it provided a 3D design tool to generate garment patterns directly in the 3D space. However, it only allowed users to specify local shapes of garments profile through 2D sketches, but users have limited flexibility to sketch the overall garment profile.



(a) (b) (c)

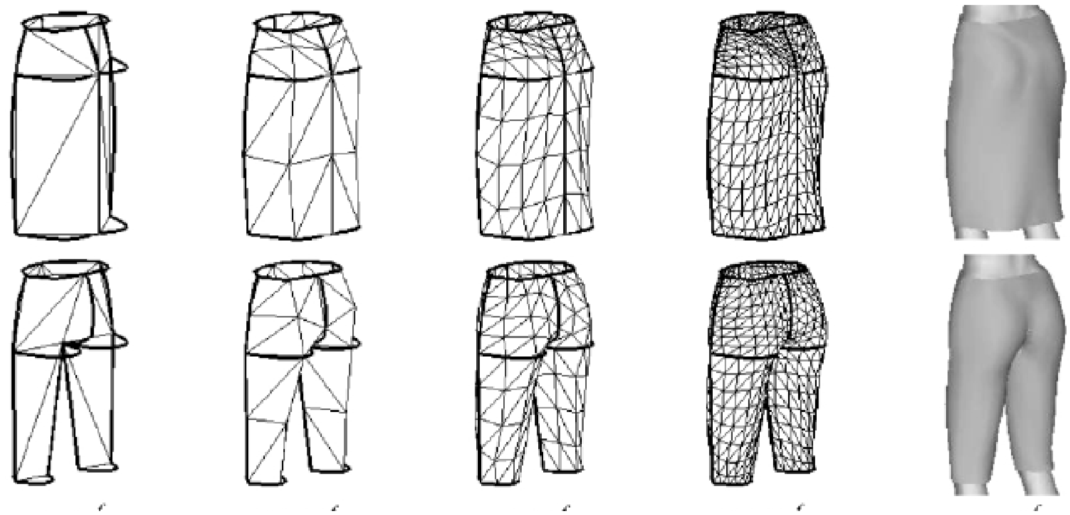


Figure 2.8 Feature based 3D garment design through 2D sketches

Turquin et al. [Turquin2004] proposed a method for interactively generating basic garments for dressing virtual characters (Figure 2.9). This method could create garments dressed on a virtual character from user-drawn strokes. Users drew an outline of the front or back of a garment, and their system could generate 3D garment surfaces around the virtual character. The 3D garment surface and the way the virtual character was wearing were determined once users finished the sketches. The generation of 3D garment surface was based on garment silhouette, its border lines and distance field to the body of the virtual character. This method inferred the variations of the distance between the garment surface and the 3D virtual character by using the distance from the 2D garment silhouette to the character model. The method could generate various kinds of garments based on sketches drawn by users. However it only generated the front or back part of a garment.

Figure 2.9 Sketching garments for virtual characters.

In 2007, Turquin et al. [Turquin2007] improved the above approach for dressing virtual characters with complete geometric garments fully based on sketches (Figure 2.10). Through drawing garment profiles around a 3D human model, that system could generate various kinds of complete 3D garment surface based on the distances from user-drawn strokes to the human model. To remove artefacts on garments, it used Bezier curves in horizontal planes to interpolate the z-values for the garment part between two limbs to mimic cloth tension. It explored an intuitive method to geometrically dress virtual humans. However, it did not provide any algorithms for processing garment shape around the torso part such back, chest and belly, which highly followed the body shape in some degree. In addition, its output was a simple garment mesh which affected the visualization results and further processing, such as fold direction as mentioned in that paper.

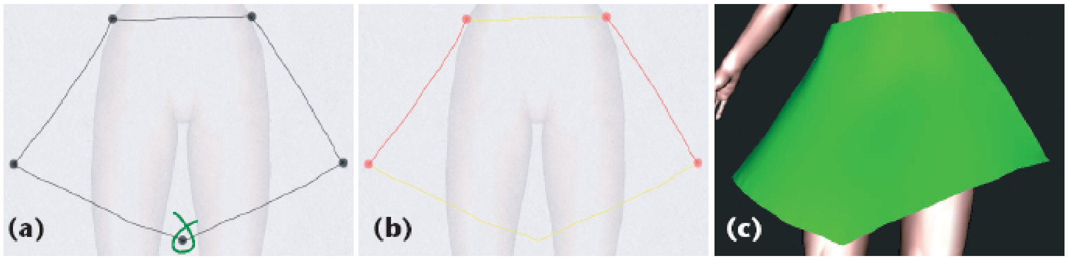


Figure 2.10 Front and back drawing and garment

2.4 Sketch-Based Techniques

Skills required for using most commercial modelling packages, such as 3D Studio Max and Maya, are only accomplished by expert users. In addition, these software or packages require significant time, expertise and artistic talent to generate complex 3D models. Sketch based modelling interfaces are much easier to manipulate and are successfully applied to create 3D geometric models from 2D sketches. Research on sketch-based modelling systems for interactively creating 3D geometric objects from user-drawn sketches has gained increasing attention in recent years.

A variety of sketch-based techniques have been developed for modelling [Naya2002, Olsen2009]. These techniques can be classified into three categories in terms of sketching techniques as follows:

Gesture-based modelling;

Reconstruction-based modelling;

The third approach is a combination of the above mentioned two methods.

The Gesture-based modelling approach uses gestures as implicit commands for constructing solid objects from 2D sketching segments.

Reconstruction-based modelling approach uses algorithms to geometrically reconstruct objects from 2D sketches that represent the 2D projection of the object.

Sketching systems can be roughly classified into three categories in terms of object functions. These three categories are as follows:

Mechanical engineering parts;

Freeform objects;

General and duct-like shape objects.

Naya [Naya2002] and Volino [Volino1997] developed systems for creating 3D solid objects, especially engineering part-like objects, from wire-frame drawings by directly drawing the axonometric view of an object to create a 3D model.

Igarashi et al. [Igarashi1999] developed a system could produce complex freeform 3D objects represented by triangular meshes using a gesture-based sketching interface. Schemidt [Schmidt2005] explored the technique using implicit models for free-form modelling from sketches. Wang [Wang2007] presented a sketch-based method for modelling 3D shapes using a multi-resolution shape representation, which allowed users to edit the shape and structure of the mesh at any level of hierarchy.

The sketch-based modelling systems introduced by Pereira [Pereira2004] and Quicksketch [Eggli1997] could generate some simple 3D object. These systems used extrusion-based techniques, which were especially good at creating duct-like shape objects.

A more recent survey is given by Olsen et al [Olsen2009], which classifies sketching systems into four categories according to the types of modelling operations considered. These four categories are model creation systems, Augmentation, deformation and surface representation.

For a gesture-based sketching interface, sketches are interpreted based on user behaviours [Qin2000, Kim2006, and Cherlin2005]. SKETCH system [Zeleznik1996] was one of the first gesture-based sketching interfaces to explore techniques for geometric modelling. It generated 3D scenes using a purely gestural interface based on simplified line drawings of primitives. All operations were specified within the 3D world through moving computer mouse instead of selecting operations from menus. The SKETCH system inferred intended shapes by recognizing two types of gestural elements: strokes and interactors. Strokes, which were pixel-tracks on the plane perpendicular to the view, were generated by the first mouse button. Strokes gestures were classified into five categories with five different mouse actions. Interactors, made by using the right mouse button, had two kinds of actions. To model a 3D scenes, a sequence of strokes and interactors were processed to perform various modelling functions with a finite-state machine. Gestures might be natural ways for users to create 3D geometry using a sketching interface. However, the difficulty is to design and recognize different gestures for different primitives. SKETCH system was developed for architectural design. Table 1 shows stroke gestures and their mouse action.

Table 2 Stroke gestures

mouse action	**stroke**
click and release	dot
click and drag without delaying	axis-aligned line: line follows axis whose screen projection is most nearly parallel to dragged-out segment
click and drag, then "tearing" motion to "rip" line from axis	non-axis-aligned line
click, pause, draw	freehand curve
click with Shift key pressed, draw	freehand curve drawn on *surface* of objects in scene

Igarashi et al. [Igarashi1999] developed a sketching interface for generating 3D freeform objects such as stuffed animals and rotund objects. It improved the usability of SKETCH [Zeleznik1996] using smaller gestures. Both SKETCH [Zeleznik1996] and Teddy [Igarashi1999] systems explicitly generated each single piece of geometric object starting from a blank canvas (Figure 2.11). In Teddy system, users drew 2D freeform strokes specifying the silhouette of an object and the output of the system was plausible 3D polygonal surfaces. Teddy system inferred users' intent and conducted appropriate editing operations through freeform strokes instead of traditional manipulation techniques such as buttons and menus. As soon as a user completed a single freeform stroke drawing, Teddy system automatically could construct a corresponding 3D shape, which could be modified using various operations. It provided some modelling operations based on the stroke's position and shape as well as the current model. These gestural operation included creation, painting, extrusion, bending, cutting, smoothing and transforming. Teddy allowed users to manipulate the camera using the secondary mouse button based on a virtual trackball model.

Figure 2.11 Teddy: a sketching interface for 3D freeform design

Quicksketch [Eggli1997] was introduced for creating preliminary mechanical CAD models allowing gestural generation of objects with inferred constraints.

Pereira et al. [Pereira2000] proposed a gesture-based intuitive design system, which used contextual information and feedback to free users from remembering detailed modelling gestures (Figure 2.12). This system allowed users to generate 3D models both from a draft by combing drawing and conventional modelling tools and by semantic transformations. It used menus for some basic commands such drawing operations (line, polyline, rectangle and ellipse), modelling functions and some scene visualization functions. To create an object from a single-view perspective, the system used the 3D gesture to indicate the start of modelling. And any following gestures were processed as 3D modelling gestures.

Figure 2.12 Towards calligraphic interfaces sketching 3D scenes with gestures

Cherlin et al. [Cherlin2005] presented a gesture-based interactive modelling system, which could generate freeform 3D objects from a few strokes. Input user-drawn strokes were processed and filtered using Chaikin scheme to allow efficient and robust 3D modelling (Figure 2.13). Objects represented by parametric surfaces were generated by the system in two phases: creation and editing. In the first phase, object surfaces could be generated using techniques for two types of parametric surfaces, rational blending surface and cross sectional blending surface. In the second phase, users could add subtle variations to the object surfaces using a single deformation stroke. The system allowed users to modify the shape of a 3D object using over-sketching cross-sections of the model.

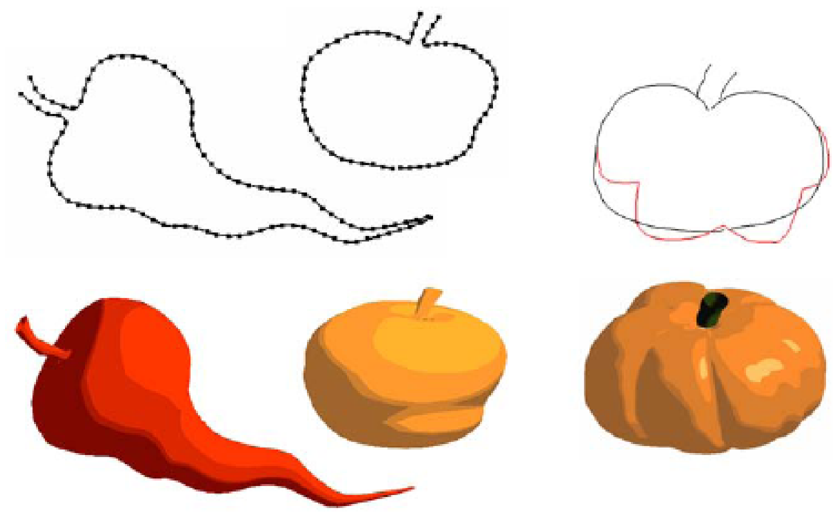


Figure 2.13 Sketch-based modeling with few strokes

For a reconstruction-based modelling system, sketches are automatically interpreted to generate 3D models. Geometric reconstruction method creates 3D geometric objects from 2D sketches, which are taken as the projection of a 3D model [Oh2003, Company2004, Kara2006, and Yang2005]. Generally, there are two main techniques used in this method. One is based on Huffman-Clowes labelling scheme and the other one is based on optimization schemes. This approach allows users to reconstruct 3D model from a single axonometric projection. Some systems have been developed in recent years such as [Lipson1996]. Schweikardt et al. [Schweikardt1998] introduced a sketching interface, Digital Clay, for modelling basic polyhedral models, which used Huffman-Clowes algorithms to derive 3D geometry. Digital Clay allowed users to re-sketch over strokes or to drag the inferred vertices to the right places. It involved more user interaction during the procedure of object construction. Turner et al. [Turner2000] presented a technique for constructing architectural 3D models based on a calligraphic interface.

Rose et al. [Rose2007] developed an algorithm for generating developable surface. Users specify polyline boundary through a sketching interface, that system could generate a smooth discrete developable surface that interpolates this boundary. When drew boundaries over an existing model, that system inferred the depth information from a single sketch. The polyline was set at a frontal distance to the model (Figure 2.14).

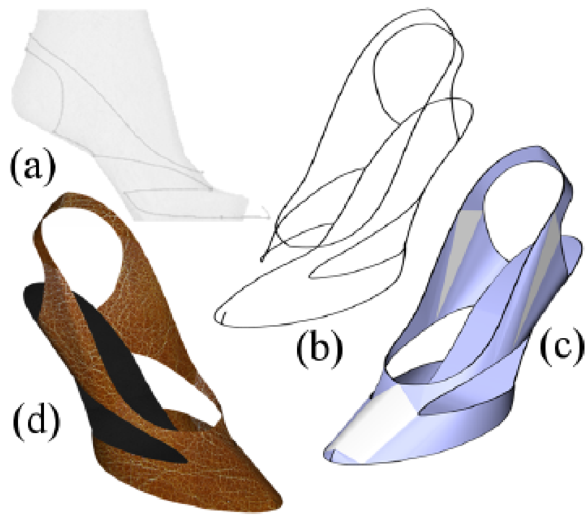


Figure 2.14 Modeling developable surface

Bourguignon et al. [Bourguignon2001] was one of the pioneers who explored sketching methods for garment design based on virtual characters. In their sketching system, users were allowed to draw strokes in 3D space and could view the sketch from arbitrary angles (Figure 2.15). The system directly used user-drawn strokes to interpret the shapes of sketches, which represented the same scene from different viewpoints, rather than attempting to reconstruct a 3D model. A 3D shape was constructed by interpreting strokes as indications of a local surface silhouette or contour based on curvature of initial strokes. When a user drew strokes in empty space, these strokes were projected onto a reference plane, parallel to the camera and containing the world origin. When drew in a space which had already contained objects, the existing entities could be used to position the curves in the space. If an existing object is pointed by the head or the end of a 2D stroke, then the existing object is used to determine a new projection plane. The depth of the point is obtained by a simple picking. And the picked object can be a geometric object or a stroke. This method provides an idea for 3D shape construction by referencing the features of existing objects. However, the system could not produce geometric surface for 3D objects.

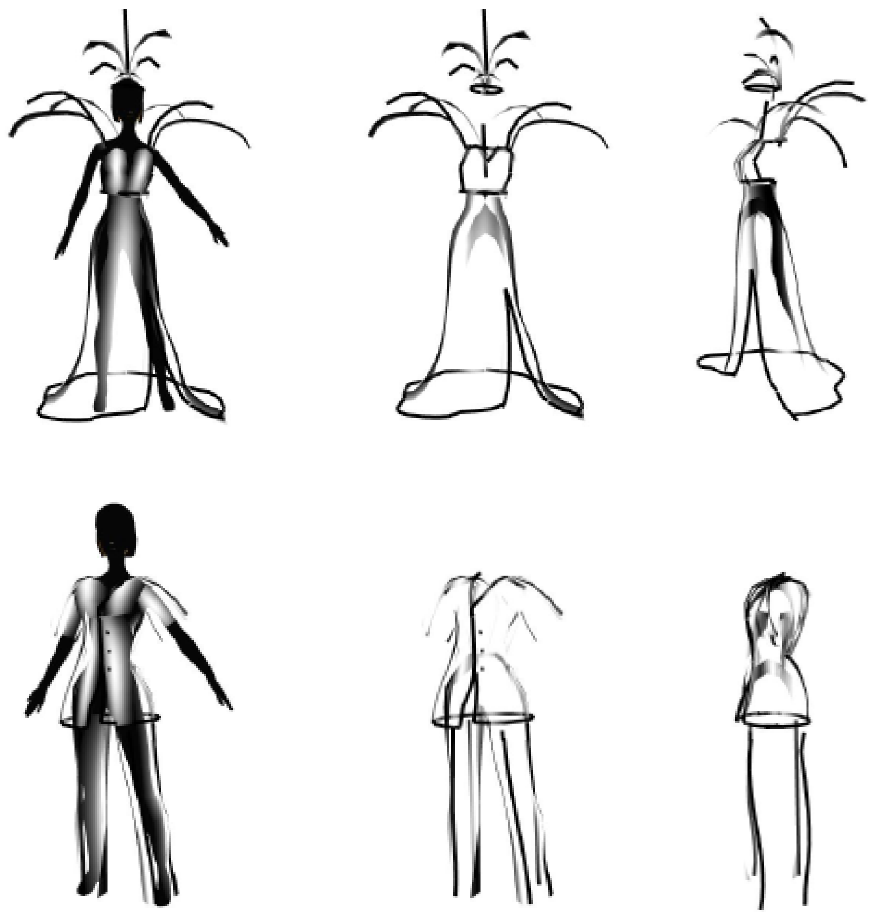


Figure 2.15 Drawing for illustration and annotation in 3D

2.5 Conclusion

In this Chapter we have reviewed the state of art in human modelling, garment design and dressing and sketching interface from a wealth of theory and practice. The method for dressing virtual humans presented in this thesis involves these three research areas. A variety of methods for human modelling have been proposed in the last thirty years. Since human models used in this thesis are obtained using a body scanning system, we have given a brief review mainly on methods for dealing with human body scan data. Virtual dressing problems cannot be separated completely from garment design, so our review on virtual human dressing includes some methods for garment design based on virtual human mannequins. Sketching interface is becoming popular in recent years. We summarized some important and related sketch-based modelling systems.

Chapter 3. Framework for Dressing Virtual Humans

3.1 Motivations

Three dimensional virtual humans are widely used nowadays. There are increasing demands for realistic virtual humans especially in real-looking appearance. A variety of research has been carried out on building realistic virtual human bodies, but little has been done on dressing virtual humans. As in cartoon movies, computer games, engineering simulation etc., realistic virtual humans usually mean at least realistic body shapes and clothes. However, the techniques used in current software package make it difficult for ordinary users to create garments and dress on virtual humans.

The standard method adopted by current software package for designing realistic virtual garments [Maya2004, Syflex2004] is very similar to the technique used in the real world: it starts with the design of all the necessary 2D fabric pieces, which requires specific knowledge and skills, but not necessarily acquired by all computer users. The tedious process of specifying these patterns and their sewing constraints adds more complexity by the need of sorting out adequate values for a set of physical parameters, even when only a static garment shape is needed.

Our research focuses on intuitively generating and dressing virtual humans. We choose sketch-based methods because people still prefer to express their thoughts or imagination by sketching them in the 2D space. It is convenient and natural for people to convey 3D shapes by sketching 2D projections and reconstruct them in their mind based on 2D projections. We are motivated to investigate the virtual dressing problem by increasing demands for realistic virtual humans in a number of application areas in science and technology and the development in related areas. We discuss these aspects in the rest of the section.

3.1.1 Traditional methods for Garment Design

The traditional method is based on garment pattern pieces, which are put on a human model and sewn together. Before these garment pattern pieces are sewn up the designers or tailors cannot know exactly what the garment will look like, and what the effect will be after worn on a human body. With the development of computer technologies, some garment CAD systems have been developed, which allow users to design 2D garment pattern pieces interactively. Designers can modify the 2D pattern pieces if the garment worn on the human body is not satisfactory. But only skilled users or professional tailors have the skills and knowledge to design garment pieces for various styles of garments. Therefore, some new methods need be developed to enable average users to design garment styles and dress on virtual humans directly.

3.1.2 Development of Hardware

With the advent of affordable 3D graphics hardware, sketching directly on a computer seems a practical choice for computer users and designers. Traditionally, when sketching, a user usually uses a pencil and paper. A tablet PC is this kind of tool for sketching directly on computer screen. A tablet PC is a pen-based laptop with a touchscreen and a digital pen or a fingertip instead of a keyboard or mouse. To sketch up an object, a user just needs to draw on the screen with the digital pen or user's finger like drawing on a piece of paper.

The development of technology makes the sketching hardware system easily portable and accessible. It is convenient for users or designers to carry or access such kind of hardware system to carry on their design by sketching. Even the development of computer mouse allows users to sketch on screen of desktop with reasonable precision.

3.1.3 Introduction to the Framework

This thesis proposes a framework for intuitively dressing virtual humans (Figure 1.1). The proposed virtual dressing framework is built based on a sketching interface and body scan data from real human body. As mentioned in the previous section, our research focuses on creating an intuitive approach for fast virtual dressing. Rather than using garment pieces for garment design, we construct a whole complete garment surface from user drawn sketches. A garment surface is represented by a geometric mesh surface. To help gain realism of garments, semantic human models are reconstructed from body scan data before users draw garment contour around the human models. We introduce three methods for dressing virtual humans as follows:

Garment generation from sweep surface;

Garment generation from offset surface;

Garment generation from sketching garment profiles.

3.1.4 Garment generation from sweep surface

The first approach uses the merits of sweep surface and the skeleton of a human body to generate trousers. It makes use of a cross section generated from the human model as the base ring for sweeping. The skeleton of a human model extracted from the human model, which is reconstructed from body scan data, is the rail for sweep. This method generates some simple trousers. Normally, this method is used together with the offset surface method introduced in the following subsection.

3.1.5 Garment generation from offset surface

This method generates garments using offset surface from the human body. This method creates garments with the shapes resembles the human body. Some tight garments can be constructed using this method.

3.1.6 Garment generation from sketching garment contour

This approach generates a garment from garment profiles sketched by a user or designer. A human model is placed in the world coordinate system and face towards the user. A user or designer draws garment contour around the human model. Distance field is used during the construction of 3D garment surfaces. To improve realism of garments, some optimisation methods are applied to the complete 3D garment surface.

There are several stages in the garment modelling framework as follows: stroke processing, garment vertices generation and optimisation, garment mesh construction and processing and penetration detection (Figure 3.2). As an important part of 3D design, shape modification feature is provided allowing users to modify garment shape in both 2D and 3D spaces. In addition, the dressing framework also provides three decoration tools allowing users to add 3D details to garment surfaces.

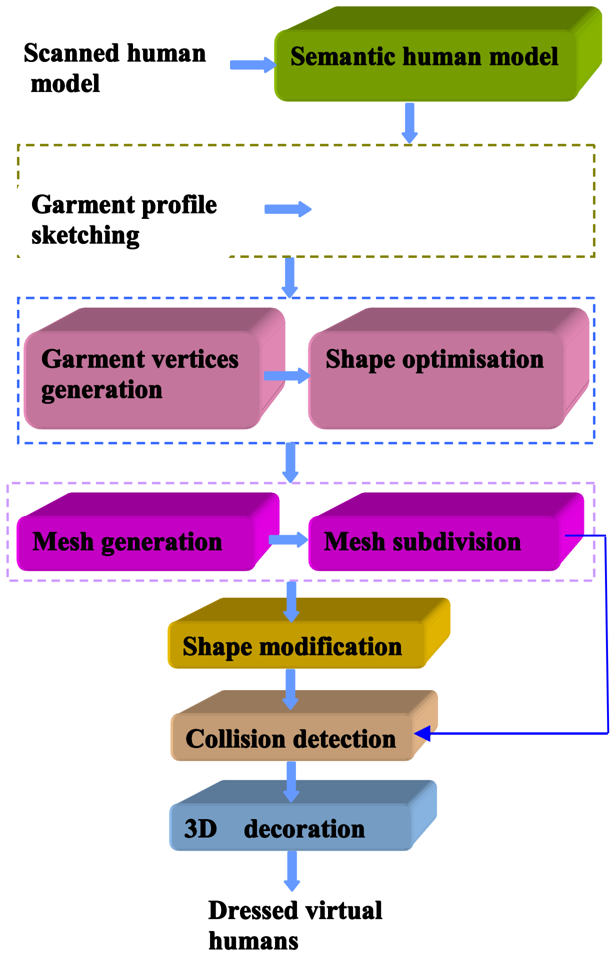


Figure 3.2 The flowchart of dressing from sketches

In the first stage, users draw garment profile strokes around a human model. These sketched strokes are processed into parametric curves using Cardinal spline curve fitting. Raw data of sketched strokes is generated using input devices such as a computer mouse or pen. Due to the intrinsic problem of input devices and variation of moving speeds of drawing pen or mouse, these sketched strokes are usually noisy and irregularly distributed. To solve the above mentioned problems, we fit Cardinal spline curves to the sketched strokes. These curves are classified into two categories including inner curve and outer curve according to whether there is intersection between the human body and the curves. Each curve is recognised and associated with a body part according to their relationships with the semantic human model.

The second stage is to construct 3D garment surface and conduct garment shape optimisation based on a level set method. Because the curves are classified into two categories, we develop two corresponding methods for these two kinds of curves for garment vertices construction accordingly. Garment vertices include two kinds of vertices: silhouette vertices and interior vertices. Each kind of vertices is generated or interpolated using different methods. These garment vertices generated using distance field method represent the rough shape of a garment. To obtain an acceptable garment surface constructed from these vertices, a level set-based optimisation method is developed.

In the third stage, garment surfaces represented by triangular mesh are constructed by sewing up the two adjacent cross sections. There are two steps to construct the whole garment mesh. Step 1: mesh generation for each part of a garment. Step 2: part mesh connection. For part mesh connection, there are two types of sewing: strap-tubular shape and tubular-tubular shape sewing. To gain a more detailed mesh surface, mesh processing schemes are applied subsequently. These mesh processing schemes include mesh smoothing and mesh subdivision.

The fourth stage is to conduct penetration detection and response. Penetration between the human model and garment surfaces happens sometimes during mesh construction or mesh optimisation procedures. A penetration detection and response method is developed based on the distance two different surfaces.

One of the important features of the framework is that it allows users to modify garment shapes in both 2D and 3D dimensional spaces. Through dragging the control points of spline curves which represent the projected shape of a garment, users are able to obtain new shapes of garments.

Finally, three decoration tools are integrated into the framework allowing users to add some details to garment surfaces. These decoration tools include a 3D pen, a 3D brush and a 3D embroidery tool. The 3D embroidery tool is the most complex tool among the three tools. The 3D embroidery tool can generate constrained Delaunay mesh.

To summarise, our dressing framework for dressing virtual humans provides the following features:

Sketching interface;

Semantic human model construction;

Garment vertices construction and garment curve shape modification;

Garment surface processing and improvement surface quality for further editing and visualisation;

Decoration tools.

Chapter 4. The Sketching Interface for Dressing Virtual Human

4.1 Introduction

A sketching interface is designed to process the input user-drawn strokes and visualise the dressing results. It is one of the most convenient and intuitive ways for users to convey 3D shapes on computer. Many sketch-based systems have been developed for various applications such as the Teddy system [Igarashi1999] for easily creating 3D free form models and the interactive sketching system [Bourguignon2002] allowing users to directly sketch in 3D.

The objective of sketching interface is to convey people's thoughts using a natural way. This is a dramatic simplification comparing with the traditional method which requires users to go through a series of operations such as pressing buttons, dragging down menus and so on. Sketching interface allows users to express their imagination using the convenient and natural form of freehand sketching. One of the most advantages of the sketching interface is that it highly improves the accessibility software solutions for various kinds of average users instead of just for some professional and skilled users. For designers, a few strokes are enough to express the main features and characteristics of an object shape. To meet the huge demands various kinds of systems have been developed [Zeleznik1996, Kara2006, Zimmermann2007, and Wang2007].

Therefore we design our framework based on sketches aiming to satisfy users' demand for dressing 3D virtual humans easily. This framework explores two interfaces for dressing virtual humans. One is generating garment patterns for virtual dressing, the other is generating garments from garment contour drawn by users and the garment is then dressed on the human body accordingly.

The framework provides a human modelling unit to generate semantic human models from body scan data (Figure 4.1). And generation of garment patterns and the method for garment generation from garment contour are based on the reconstructed human model .The input data can be Obj file or plain text file. Figure 4.2 shows an example of input data.

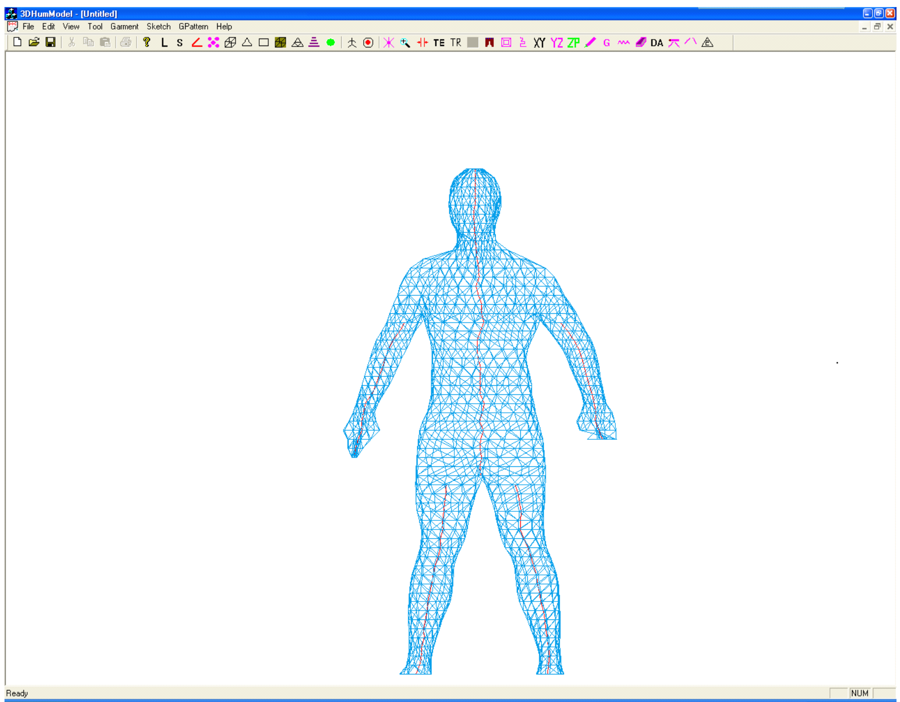
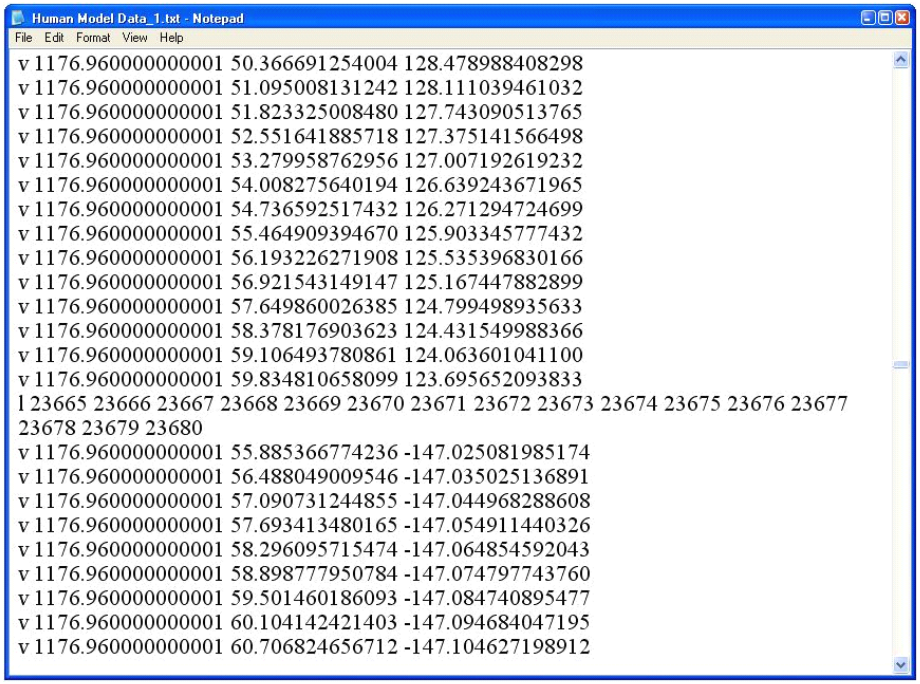


Figure 4.1 Semantic human model generation




```
v 1176.960000000001 50.366691254004 128.478988408298
v 1176.960000000001 51.095008131242 128.111039461032
v 1176.960000000001 51.823325008480 127.743090513765
v 1176.960000000001 52.551641885718 127.375141566498
v 1176.960000000001 53.279958762956 127.007192619232
v 1176.960000000001 54.008275640194 126.639243671965
v 1176.960000000001 54.736592517432 126.271294724699
v 1176.960000000001 55.464909394670 125.903345777432
v 1176.960000000001 56.193226271908 125.535396830166
v 1176.960000000001 56.921543149147 125.167447882899
v 1176.960000000001 57.649860026385 124.799498935633
v 1176.960000000001 58.378176903623 124.431549988366
v 1176.960000000001 59.106493780861 124.063601041100
v 1176.960000000001 59.834810658099 123.695652093833
l 23665 23666 23667 23668 23669 23670 23671 23672 23673 23674 23675 23676 23677
23678 23679 23680
v 1176.960000000001 55.885366774236 -147.025081985174
v 1176.960000000001 56.488049009546 -147.035025136891
v 1176.960000000001 57.090731244855 -147.044968288608
v 1176.960000000001 57.693413480165 -147.054911440326
v 1176.960000000001 58.296095715474 -147.064854592043
v 1176.960000000001 58.898777950784 -147.074797743760
v 1176.960000000001 59.501460186093 -147.084740895477
v 1176.960000000001 60.104142421403 -147.094684047195
v 1176.960000000001 60.706824656712 -147.104627198912
```

Figure 4.2 Example of input data

After loading input data, a point human model is displayed on the main window. To obtain a semantic human model represented by surface, the point human model needs to go through three phases:

Segment the human model into six parts: chest, torso, right arms, left arms, right leg and left leg. Triangulation: each body part is reconstructed through sewing up two neighbouring cross sections. And finally, related body parts are connected by stitching cross sections. Semantic information extraction: body features and skeleton are extracted for garment modelling.

A setting function is designed to provide users with more flexibility to control levels of detail of the human model. Users can set sampling parameters on each segment of the human model to obtain human models with different levels of detail (Figure 4.3). Users can obtain different LOD models by typing in different values of sample angle: Torso sample angle, Right arm angle, Left arm angle, Right leg angle, Left leg angle and chest angle. Please see detail of sample angle in section 5.3.3.

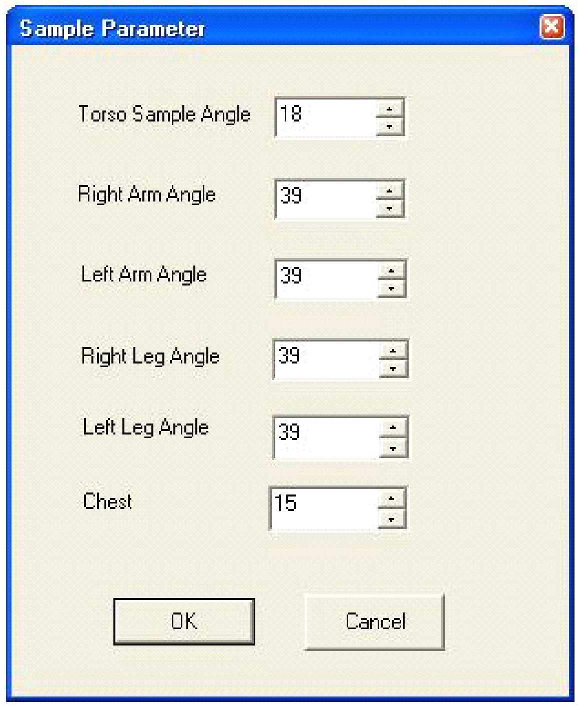


Figure 4.3 Sampling parameter setting

The simplest way for dressing a virtual human is to select garments from the database and automatically put them on human bodies. An interaction-free method for geometric pre-positioning of virtual cloth patterns on 3D figurines was presented in [Fuhrmann03], which was established based on clothing pattern database. In that database, clothing patterns were stored with associated information such as sewing, texture, material properties etc. That system was capable of dressing virtual humans with realistic appearance, but it was not designed to allow users to intuitively change garment parameters to satisfy the individual taste.

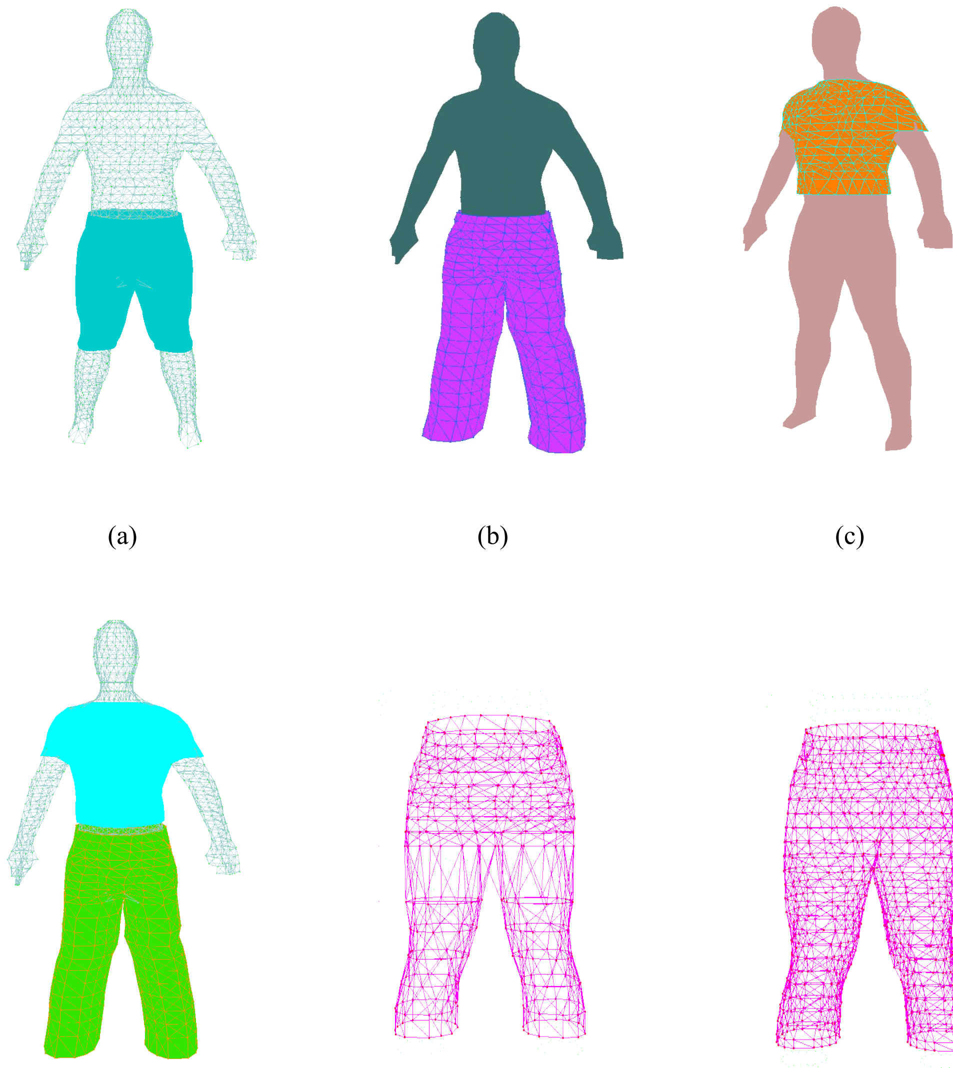
4.2 Interface for Garment Pattern Generation

In this thesis, we introduce two methods for generating garment patterns, which are then stored in a database. A *garment pattern* is a geometric description of an article of garment instead of clothing pieces. All garment patterns have been set with default positions and sizes in terms of key body features of various human bodies. After loading scanned human data, once the semantic human model is reconstructed, all garment patterns are pre-positioned and are ready to be selected by users.

The system can generate garments such as tight trousers and T-shirts by offsetting points on the body surface equally (Figure 4.4 (a) and (c)). Actually, points on each cross section are offset outwards along their radial direction by the distance Δs_i (Figure 4.5) in the local coordinate system. A local coordinate system is defined as follows: the origin is at the centroid of the cross section and x, y and z axis points to the same direction as the global coordinate system. Then the offset point can be calculated as follows:

$$\mathrm{q}\,(x, y, z) = \mathrm{t}^{\mathrm{T}}.\mathrm{p}(x, y, z) \qquad (4\text{-}1)$$

where q (x, y, z) (the outside blue point in Figure 4.5) is a point on the garment surface S and p(x, y, z) (the inside black point in Figure 4.5) is a point on the human body surface Ω. The transformation function t(x, y, z) defines a local coordinate transformation on each point. By default, t$(x, y, z) = [1+ \Delta x_i, 1, 1 + \Delta z_i]$, $\Delta s^2_i = \Delta x^2_i + \Delta z^2_i$. The offset distance Δs_i specified by the user is a scaling factor along both x and z direction. More parameters are provided for the user to create some details. Points on each cross section of a human body surface and garment refer to the reference skeleton. Through interpolating cross sections at any position along the reference skeleton, a user can create various details of the garment. Figure 4.4 (e) shows the coarse trousers mesh model and Figure 4.4 (f) is a refined trousers mesh model. These can be achieved easily through changing the parameter of mesh density level by the user.



(a) (b) (c)

(d) (e) (f)

Figure 4.4 Garment generation from offset surface

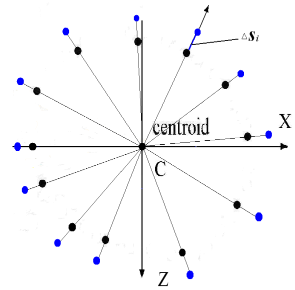


Figure 4.5 Point offset

4.2.1 Trousers Generation from Sweep and Offset Surface

A 3D model can be obtained by sweeping a 2D cross section along a 3D trajectory [Requicha1980]. Many algorithms for creating swept surface or volume and their applications have been introduced [Blackmore1997, Abdel-Malek2000]. In this method, a pair of trousers consists of two parts: trousers legs and hip-waist part. Here, we apply a sweep technique to generate legs of some trousers patterns. We use an offset surface method to create the hip-waist part of a pair of trousers, which is detailed in section 4.2.3.

The 3D surface of a garment can be created by sweeping a 2D template cross section along the reference skeleton. We use a simple but efficient linear sweep method, the sweep trajectory being a straight line [Barthe2003]. Our trajectory of sweep is a reference skeleton, which is linear between the neighboring two centroids. Therefore, this method proceeds in two steps:

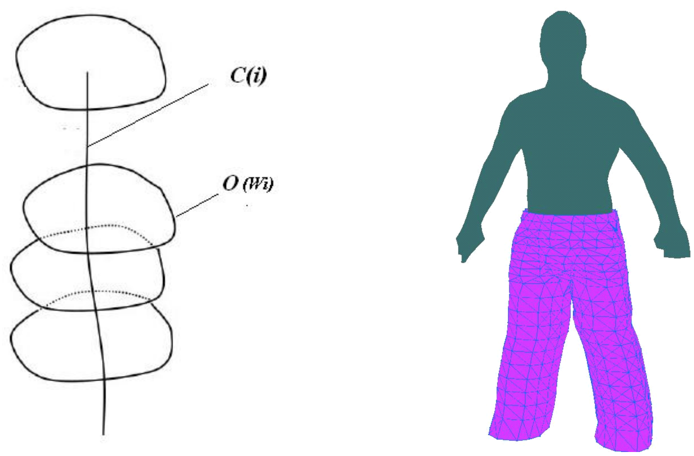
Generation of the template cross section;

Sweep of the template cross section along the reference skeleton.

The radial offset method introduced in the previous section is applied on these two cross sections to create a template cross section with constraints, the offset Δs_i given by the user. After obtaining the template cross section, points on the template cross section are mapped to points on a garment. The swept garment surface $S_w(i, w)$ can be described by the following equation:

$$Sw(i, w) = C(i) + O(w_i) \tag{4-2}$$

where $C(i)=[x_i, y_i, z_i]^T$ is the reference skeleton. The vector $O(w_i)=[x_i*tx_i,\ 0,\ z_i*tz_i]^T$ represents a transformed cross section (Figure 4.6). Variables tx_i and tz_i are scaling factors along x and z directions. Simply, the system takes tx_i= 1+ Δs_i, tz_i= 1+ Δs_i as default. Furthermore, through replacing Δs_i by a more complicated function, we can obtain more complex garment shapes.



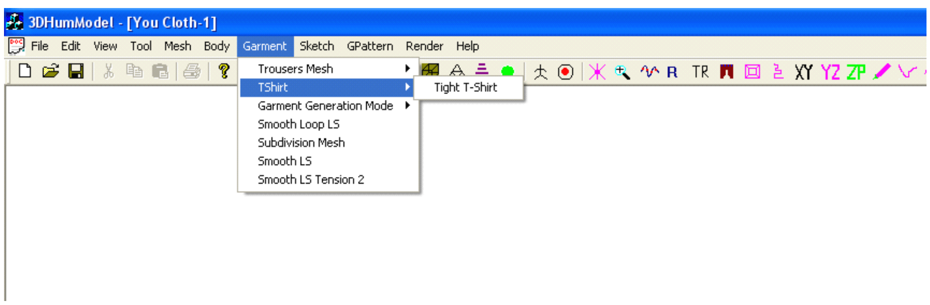
(a)) Sweep surface generation. (b) Trousers generated by sweep and offset surface.

Figure 4.6 Trousers from sweep surface

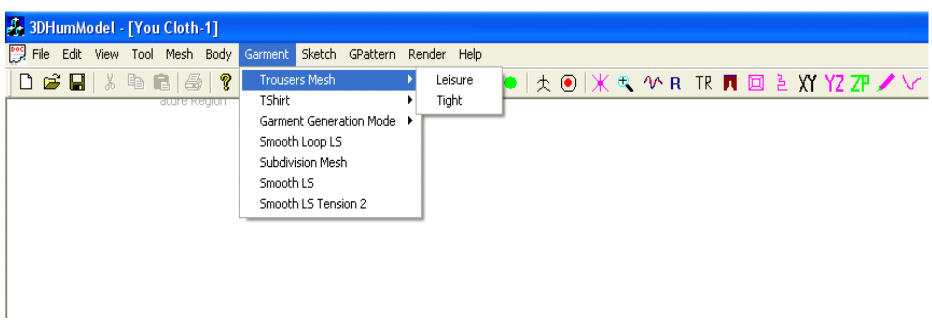
Let us take legs of loose trousers as an example (Figure 4.6 (b)). We choose the first two cross sections below the crotch point on each thigh as base cross sections, which are then converted into two template cross sections by specifying a fixed radial offset constraint Δs_i. These two template cross sections are used to sweep along the reference skeleton of two legs. We specify zero as the low height *h_l*, so these two trousers legs cover the whole legs. It is worth pointing that the upper part of the trousers from the crotch point to the waist is constructed by the radial offset method because the offset surface can match this part well.

4.2.2 Implementation

To dress a human body with a T-shirt, users just click the "Tshirt" item under the menu “Garment" on the main menu (Figure 4.7 (a)). In other words, when users select a garment from the database by clicking on the item “Leisure” or “Tight” under “Garment” and “Trousers Mesh” on the main menu (Figure 4.7 (b)), it will be dressed on the default position of the human body automatically.



(a) T-shirt menu



(b) Trousers menu

Figure 4.7 T-shirt and trousers menu

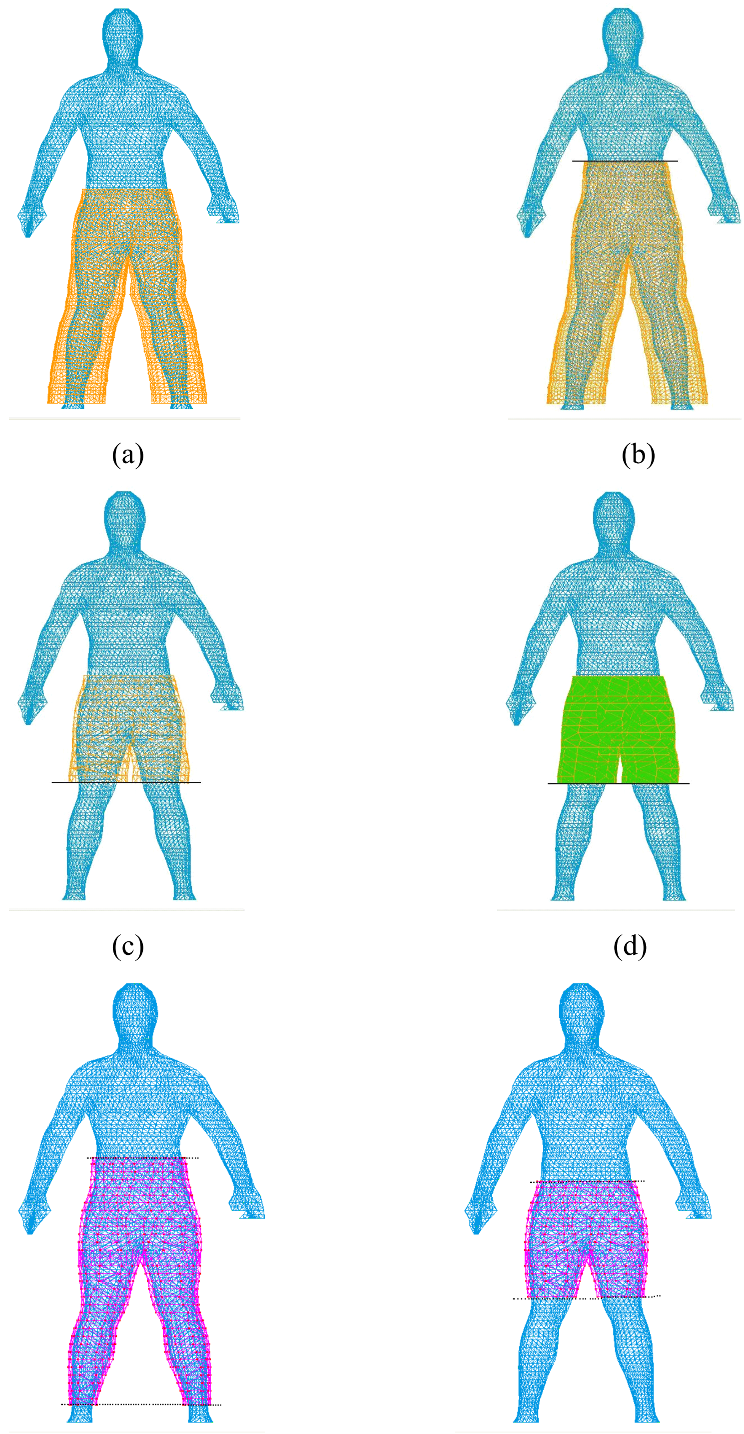
The interface allows users to change some measurements of a garment by sketching directly on the human body. In this thesis, we introduce a simple sketch-based approach for specifying these parameters intuitively. There are two kinds of parameters that a user can specify: Trousers and T-shirt.

Trousers

Users can change the length of a pair of trousers legs by sketching a horizontal stroke on the human body. This stroke specifies the edge positions of trousers legs or waist edge. Figure 4.8 (a) shows a human model wearing a pair of trousers with a default length of waist length and leg length. Figure 4.8 (b) demonstrates the new length of a pair of trousers waist specified by a user sketch. Figure 4.8 (c) illustrates a model wearing a pair of pants, which is obtained by specifying the new length of a pair of trousers legs. Figure 4.8 (d) and (i) are the rendering results of a pair of pants and a pair of trousers. Figure 4.8 (e) and (f) show models wearing tight trousers with different length of trousers legs specified by user sketches. Figure 4.8 (g) is a rendering result of a pair of tight trousers.

T-shirt

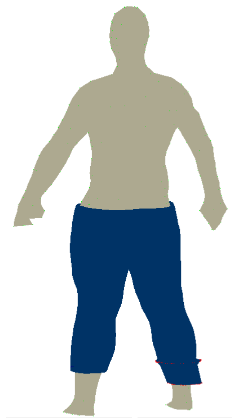
Users can change the length of T-shirt by sketching on the human model to specify the lower edge. Figure 4.8 (h) is a human model wearing a T-shirt.



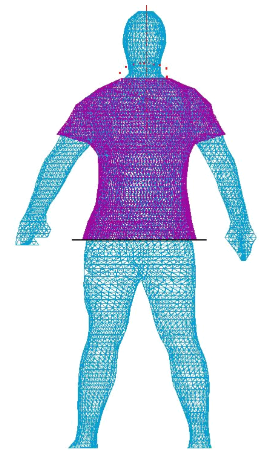
(a) (b)

(c) (d)

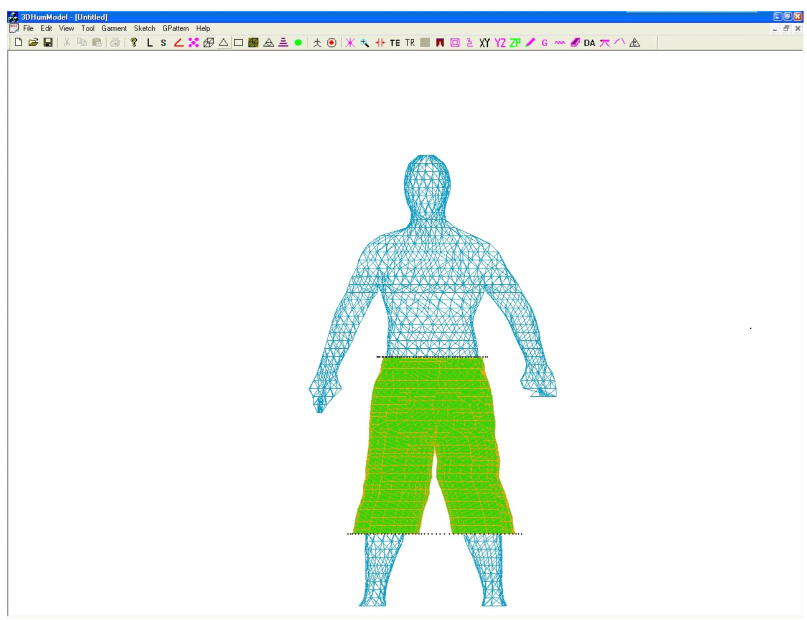
(e) (f)



(g)



(h)

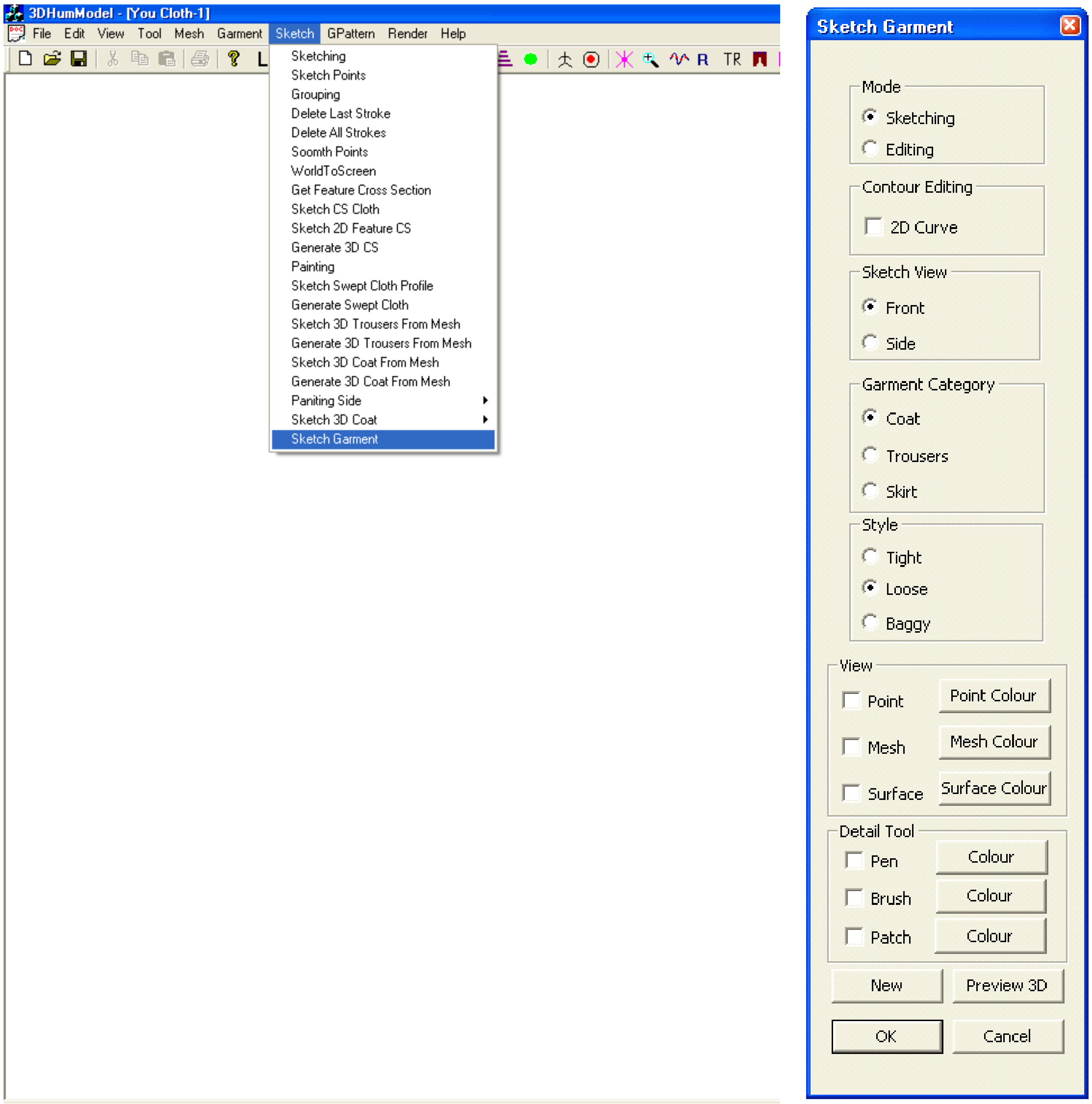


(i)

Figure 4.8 Specify garment parameters by sketching

4.3 Interface for Sketching Garment Contour and Processing

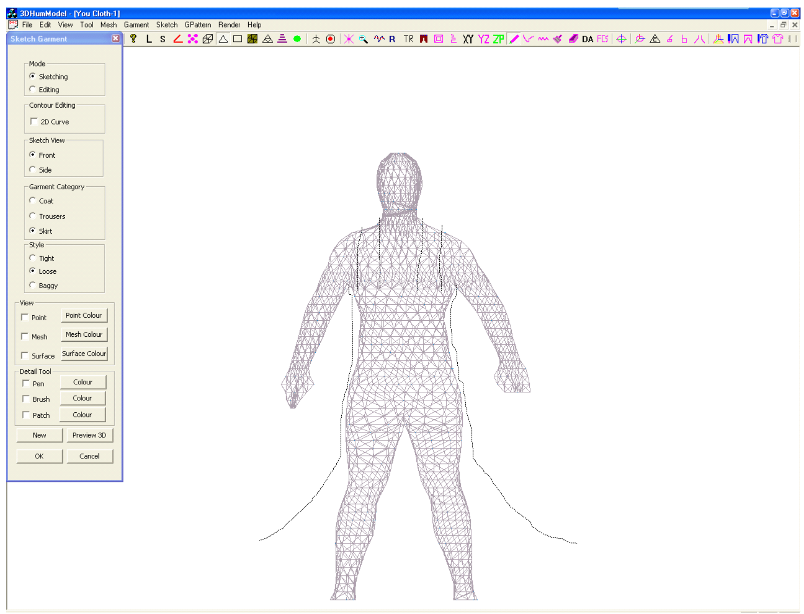
The sketch-based interface allows users to draw a 2D garment profile around a 3D human body. The sketch profile represents the front-view shape of the garment. When a user clicks the item "Sketch Garment" on the main menu, a "Sketch Garment" dialogue window will pop up allowing the user to choose options for sketching garments (Figure 4.9 (a) and (b)).



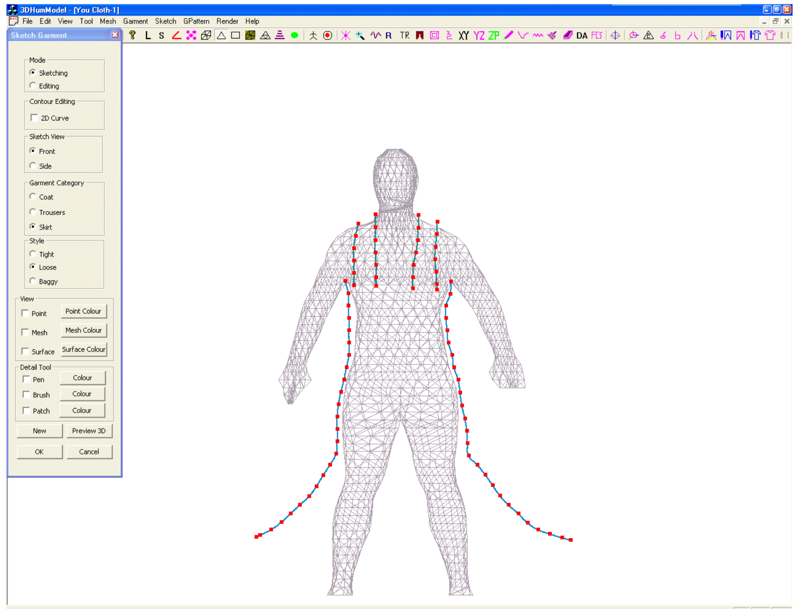
(a) Main menu (b) Dialogue window

Figure 4.9 Main menu and sketch garment dialogue window

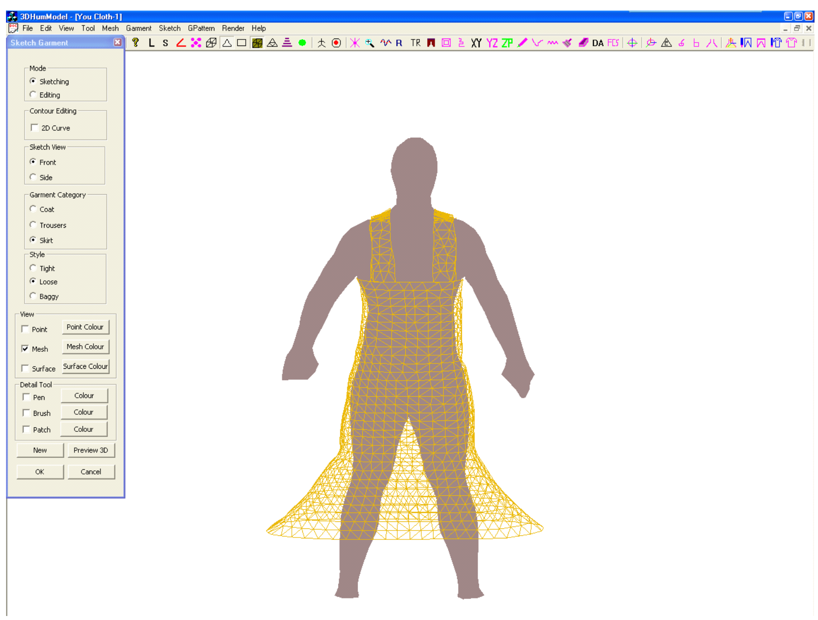
Users can choose to sketch a coat or a skirt by clicking on buttons in the "Garment Category" on the "Sketch Garment" dialogue window. The default choice of the "Garment Category" is the "Coat" item. A user draws strokes of a garment profile on the screen (Figure 4.10 (a)). These strokes are then converted into spline curves with the completion of the stroke. After finishing all sketches around a human model, the profile of a garment is represented by a group of spline curves (Figure 4.10 (b)). When the user clicks the "Preview 3D" button on the "Sketch Garment" dialogue window, an elaborate 3D geometric mesh surface of the garment is generated and dressed on the body (Figure 4.10 (c)).



(a)



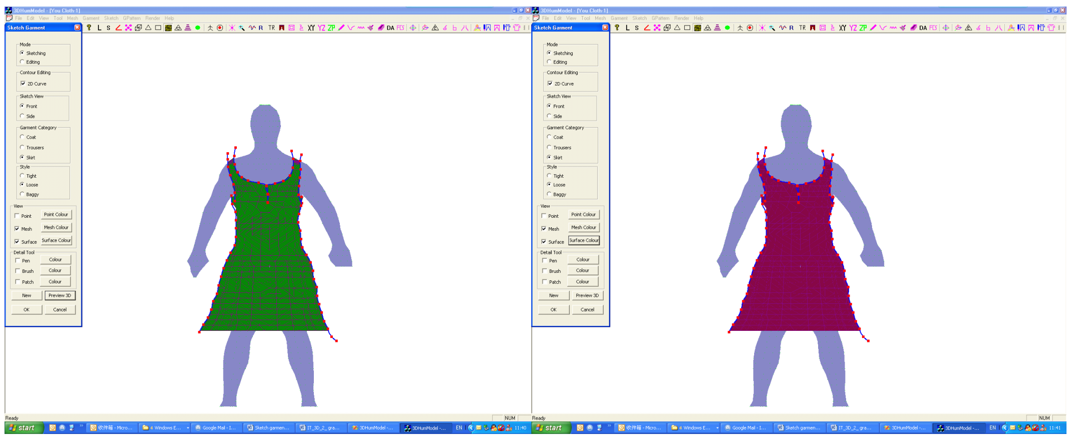
(b)



(c)

Figure 4.10 Change colour of a garment surface

User can view point, mesh and/or surface garment model by selecting the check boxes, which are Point, Mesh and Surface. The default choice for visualization of the garment model is a garment mesh model. Users can change the visualization colour for point, mesh and surface of a garment model through clicking the corresponding colour button on the panel (Figure 4.11).



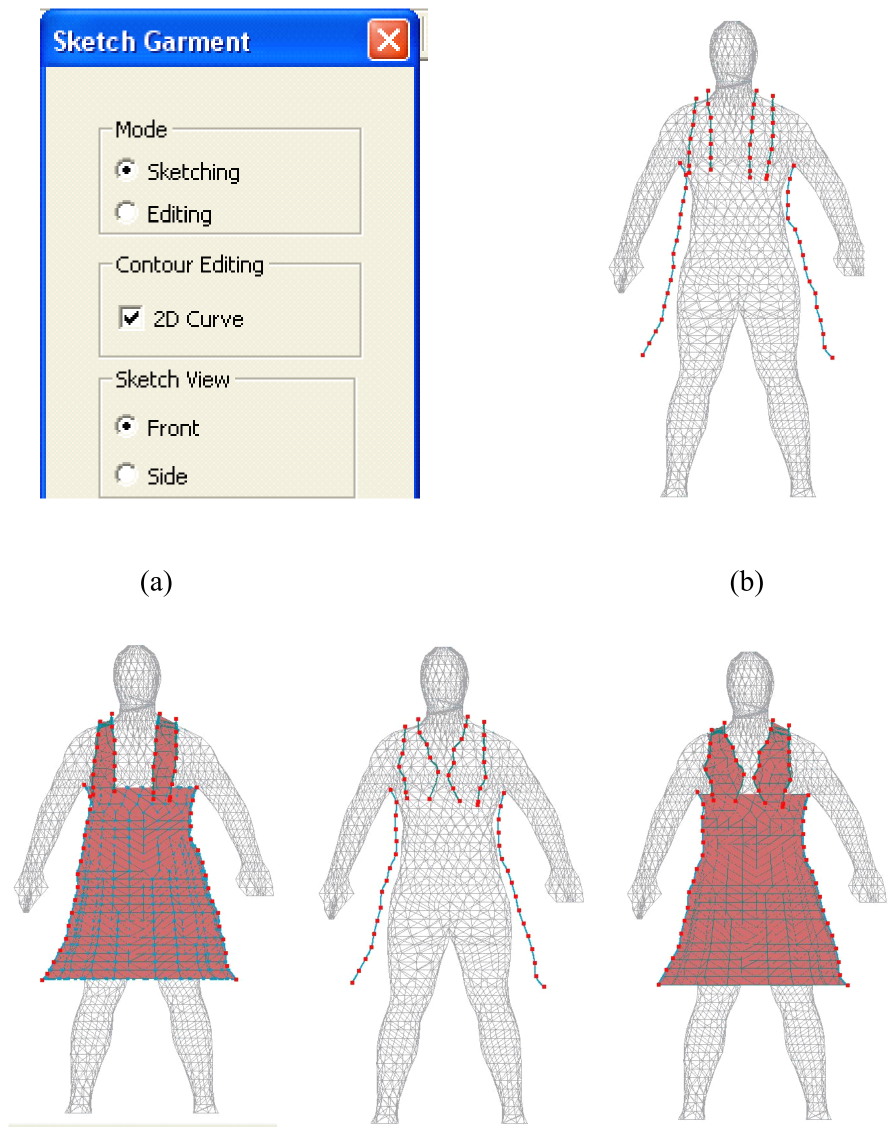
(a) (b)

Figure 4.11 Change colour of the garment surface

The procedure for vertex construction of the garment is constrained by key body features. A level set-based method is developed to optimize the garment shape. To improve visualization and obtain a more detailed garment mesh for further processing, mesh refinement schemes are applied through click items on the main menu.

Users are allowed to edit these spline curves after selecting the “2D Curve” radio button on the “Sketch Garment” dialogue window (Figure 4.12 (a)). And Figure 4.12 (b) and (c) demonstrate a skirt profile around a human model and its surface generated from the skirt profile. Figure 4.12 (d) and (e) show a new garment profile modified using editing tool based on the original profile and its surface.

Three 3D decoration tools are provided, allowing users to add decorations to the garment surface. Through clicking any checkbox under the “Detail Tool” on the dialogue window, users can pick up the corresponding tool to add detail to the garment model (Figure 4.13).



(a) (b)

(c) (d) (e)

Figure 4.12 Garment modification

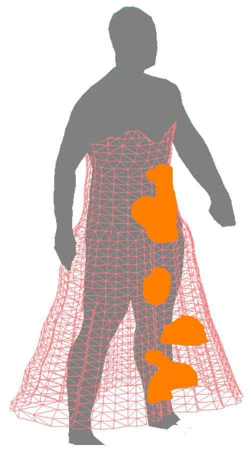
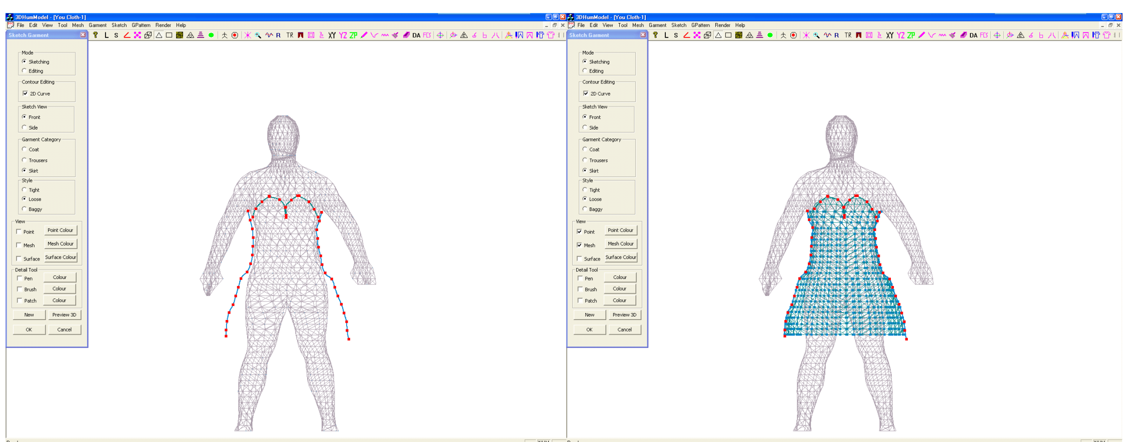


Figure 4.13 3D decoration

4.4 Testing Results

Figure 4.14 shows an example of generation of a pair of evening skirt. Figure 4.14 (a) shows curve generation after the user complete sketches of a skirt contour. and Figure 4.14 (b) demonstrates the 3D skirt mesh generated from the curve of the skirt contour in (a).



(a) (b)

Figure 4.14 From sketches to 3D garment surface

4.5 Conclusion

This Chapter described the interface for generating 3D garments from 2D sketches. This interface generally integrated two functions including a semantic human model generation function and a 3D garment generation function. The semantic human model function takes slice point human model as input and generates mesh models with semantic information. In the 3D garment generation function, several features are created to enhance ease-of-use and effectiveness of 3D garment generation and modification. These features are summarized as follows:

Sketches processing techniques;

Three 3D decoration tools;

Interactive dialogue window for sketching and editing options;

Techniques for interactively determine garment sizes from the database.

In addition, two basic garment generation methods, which are sweep surface method and offset surface method, have been described to demonstrate the ease-of-use of the interface.

Chapter 5. Semantic Human Body Modelling

5.1 Introduction

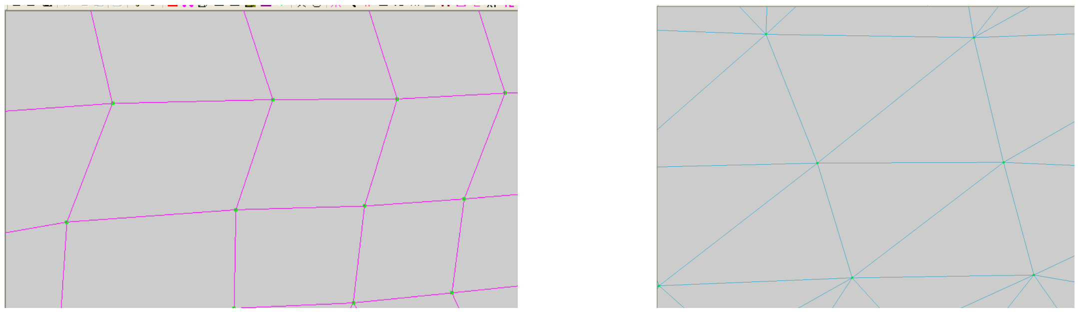
In this chapter, we describe the process of creating semantic human models which is a pre-requisite for our dressing method. The virtual dressing method is based on semantic human models. A semantic human model is a human body with semantic information represented by certain of structure and body features.

The raw data of human models we use is obtained from a $[TC]^2$ body scanner. Once a human model is reconstructed in a certain structure, we extract some key body features which are used as constraints during the procedure of 3D garments generation and virtual dressing. These key body features include the positions of some feature points on the body surface and feature curves.

5.2 Data Structure

5.2.1 Basic Concepts

One important consideration is the representation of human models. There are various kinds of ways to describe an object surface. One of the most common-used representations is the polygon model surface. Polygonal model representation is a set of planar polygons such as triangles or quadrilaterals (Figure 5.1), which are the main representation for 3D objects. Polygonal mesh models can efficiently approximate surfaces with arbitrary complexity.



(a) Rectangular mesh. (b) Triangular mesh

Figure 5.1 Mesh types

5.2.2 Topological Entities

We define our mesh models using the same description in [Hoppe1993]. A mesh M is represented by a set of vertices V defining the shape of the mesh in $\Re^n$ and a simplicial complex K representing the connectivity of the vertices, edges and faces. Where $V = \{v_1, v_2, ..., v_m\}$, $v_i \in \Re^n$. The simplicial complex K has the following elements:

Vertices: $\{1\},\{2\},\{3\}$;

Edges: $\{1,2\},\{2,3\},\{1,3\}$;

Faces : $\{1,2,3\}$.

In this thesis, two kinds of three dimensional objects are modelled: 3D human models and 3D garments. We store all elements of mesh model in the triangular topological entity, which is a structure of tMeshSurface. The topological entity structure should store enough information about surfaces of these geometric models for mesh processing and visualisation. A topological entity is defined by its connectivity instead of position information in its space. In this thesis, a geometric model includes several types of topological entities as follows: vertex, edge, triangular face and surface. A data hierarchy is constructed for holding all information of a geometric model. The higher level entity is defined based on its data structure by its connectivity with the lower entities. Figure 5.2 demonstrates the hierarchy of these data structure.

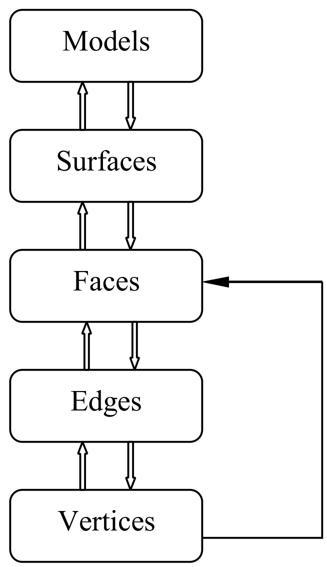


Figure 5.2 Hierarchy of these data structure

Here are some data structures used in this thesis for the construction of entities as follows:

```
Vector3
{
Struct {float x, y, z} ;
…
};
```

```
tNormal
{
  Bool flag; // status flag of normal
  Vector3 norm;
};
tTriangleEdge
{
Int index;// which edge
Bool flag;
Int *vertIndex[2];// vertex index
Vector3 edgeVert[2]; //two vertices
};
tTriangleFace
{
   Int index; // the index number of the face
   Bool  flag; // status flag of the face
   Vector3  vert[3]; // three vertices
   tNormal; // normal of the face
   tTriangleEdge *triEdge[3];
};
tMeshSurface
{
    Int index;
```

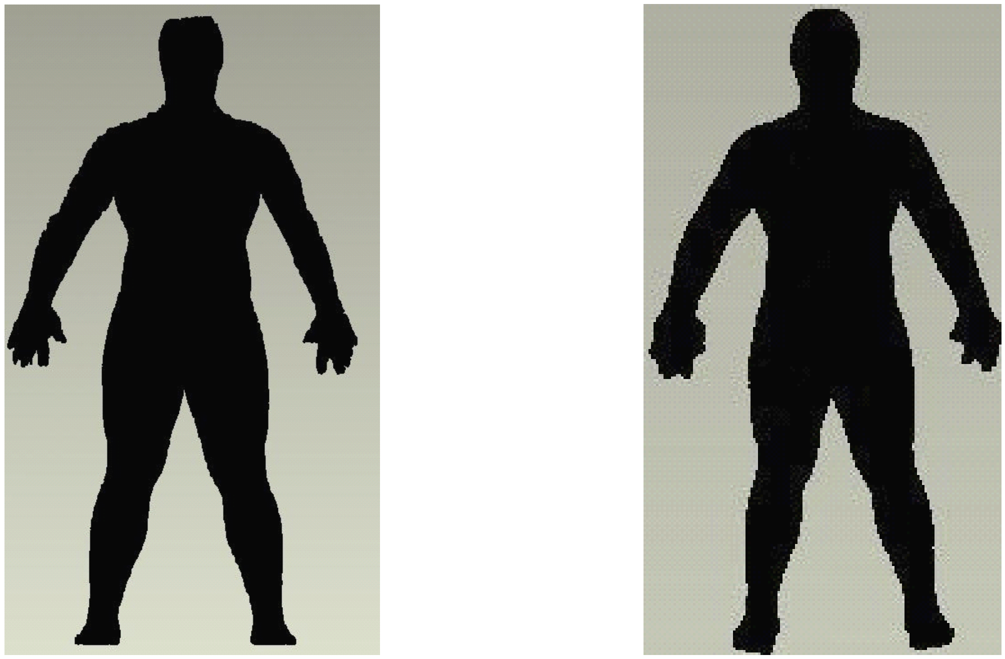
```
    Int flag; //
    Int numFace;// face number
    Vector <tTriangleFace>  triFaces; //triangle face vector
    Vector <tTriangleEdge> triEdges; // triangle edge vector
};
```

5.3 Human Body Modelling

5.3.1 Data Pre-processing

The human body data we used is obtained using [TC]^2 body scanning system [Textile]. The body scanner contains twelve scanning heads, each of which consists of a camera. The scanning heads are oriented about 90^0 apart with three at each angle in order to capture the full body shape. The scanning volume under this configuration is roughly 1.2m in diameter and 2 in height. When the scanning process starts, the subject being scanned should remain still. The whole scanning process takes only a few seconds.

Scanned human body data is normally prone to containing noises, holes etc. due to insufficient scans going through every detail of the human body or occlusions. A lot of methods for hole filling have been proposed [Allen2002, Seo2003, Kolja2002]. To make it simple, we use Geomegic Software [Geomagic] to remove noises and fill holes of the scanned data. Figure 5.3 (a) shows an image of the original human body scan data from a body scanning system. And Figure 5.3 (b) shows the image of the human model after pre-processing, in which holes such the big one on head have been filled.



(a) Original data (b) Hole filling

Figure 5.3 Noise removal and hole filling

5.3.2 Human Body Segmentation

The human model represented by *M_h* is paced in the world space facing towards z-axis. The human model is segmented into six parts: chest, torso, right arm, left arm, right leg and left leg.

General speaking, the key of the segmentation procedure is to find out some feature points. Two steps are applied to this procedure. First of all, a small feature region is determined, which contains the key features for segmentation. And then, special searching algorithm is adopted to locate these feature points. We identify feature regions according to these feature points.

The measurement of a human body by proportions goes back to Da Vinci, who was a Renaissance artist and created Viturvian Man, a drawing, based on proportions. Leonardo Da

Vinci's famous Vitruvian Man (Figure 5.4) demonstrates standard proportions of a human body. In this thesis, we use eight-head-tall-body division to measure a human body [Ratner2003]. Figure 5.5 (a) shows a demonstration of eight-head-tall-body division. And Michelangelo's David stature matches the eight-head-tall-body division (Figure 5.5 (b)).

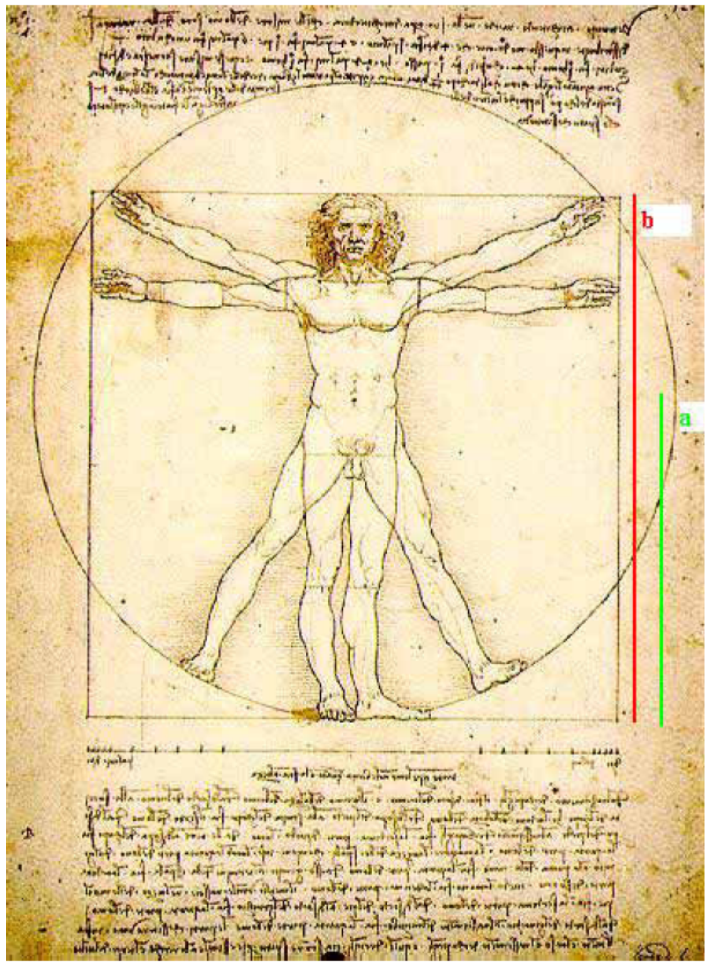
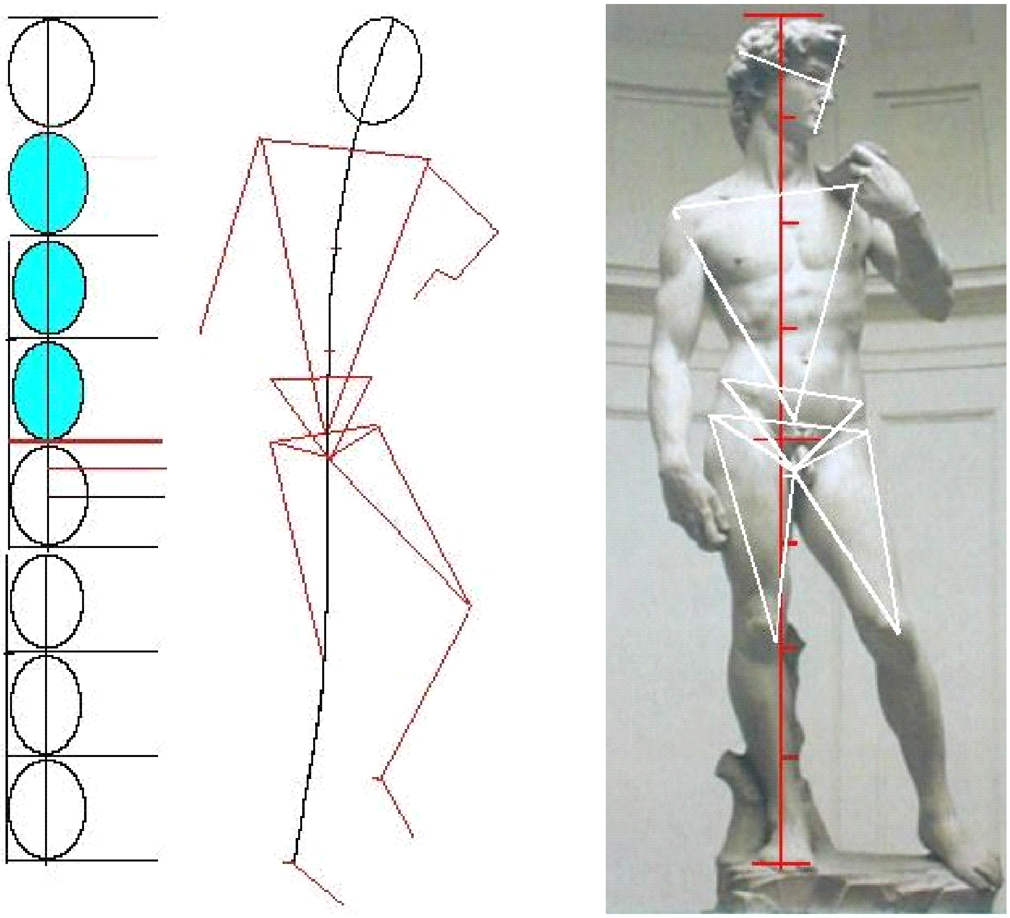


Figure 5.4 Leonardo Da Vinci's Vitruvian Man



(a) A demonstration of eight-head-tall-body division. (b) Michelangelo's David stature

Figure 5.5 Eight-head-tall-body representation

1) Head segmentation

Assume that M represents the human body and the height of the human model is H. The total number of points of the human model is N. We use right-handed coordinate system: $+X$ to the right (horizontal), $+Y$ up (vertical), $+Z$ out of the screen. And then the human model is oriented to face $+Z$ direction. Then the feature region for the segmentation of head is R_h , which is the region between the two red lines shown in Figure 5.6:

$$R_h = \{P_j \mid 6.7/8H < Y(P_j) < 7.2/8H,\ j=0,\ 1,\ 2,\ \dots,\ N_h\},\qquad (5\text{-}1)$$

Where $P_j \in M$ and j is index of model points. And N_h is the total number of points in the feature region of the head. When the feature region is determined the next step is to find out the minimum distance between end points on each cross section within the feature region. This procedure is implemented in two- dimensional plane.

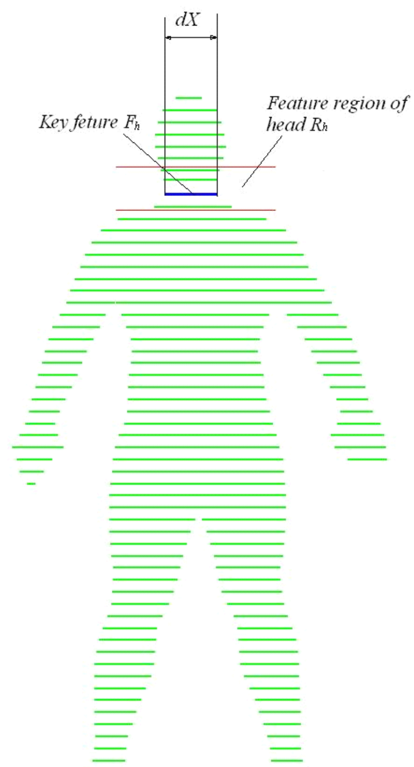


Figure 5.6 Model projection with head feature region

Step1: project the human model onto *XY* plane. We then get the contour of the body with only X and Y coordinates.

Step 2: computer the minimum distance min(dX) (Figure 5.6)between end points along X axis on each projected line which represented a cross section within the feature region:

$$min(dX) = min(||max|X_j|-min|X_k|||), \tag{5-2}$$

Where $max|X_j|$ and $min|X_k|$ are respectively the maximum and minimum X coordinate of the feature points $P_j(X_j, Z_j)$ and P_k (X_k, Z_k) on each horizontal line. Then points on the body with $Y=Y_i$ form the key feature F_h (Figure 5.6). therefore , all points on the body above key feature F_h are classified as head.

2) Arms segmentation

The segmentation of arms is more difficult than the head because of its more complicated topology. These cross sections in the region above armpits and below the height 6.1/8H are defined as complicated topology. The rest cross sections except these complicated cross sections are simple elliptic cross sections. Li [Li1997] proposed to segment the human body using cross sectional data through recognizing the feature of concavity between the arm and trunk. The principal of the method presented in this paper is to find out the four feature points of arm pits and then the separation line goes through these four feature points.

The first still goes to determine the feature region R_a:

$$R_a = \{P_i \mid 5.4/8H<Y_i<6.1/8H, i = 0, 1, 2, \ldots N_a\}, \tag{5-3}$$

Where P_i $(X_i,Y_i,Z_i)$is points on the human model and N_a is the total number of points in the feature region of arms.

Then four steps are adopted to find four feature points .

Step 1: compute the centroid $C_j(X_c, Y_c, Z_c)$ of each cross section within this feature region.

$$\begin{cases} X_c = \dfrac{\sum_{i=1}^{N} X_i}{N}, \\ Y_c = \dfrac{\sum_{i=1}^{N} Y_i}{N}, \\ Z_c = \dfrac{\sum_{i=1}^{N} Z_i}{N} \end{cases} \tag{5-4}$$

Where $P_i(X_i, Y_i, Z_i)$ are points on cross section j. And N is the total point number of the cross section.

Step 2: Compute the width W of each cross section within this feature region along x-axis.

$$W = \left| |X_{e1}| - |X_{e2}| \right|, \tag{5-5}$$

Where $P_{e1}(X_{e1}, Y_{e1}, Z_{e1})$ and $P_{e2}(X_{e2}, Y_{e2}, Z_{e2})$ are two extreme points on the cross section with the maximum and minimum X coordinate respectively along X-axis.

Step 3: compute candidate feature points within the feature region.

We process the data by using sign function $f(\Delta Z_i)$ in four quadrants (Figure 5.7) on each cross section.

Quadrant 1:

$Q_1 = \{P_{i1} \mid (X_c+0.25W) < X_{i1} < (X_c+X_{e1}-r)\}$, Quadrant 2:

$Q_2 = \{P_{i2} \mid (X_c\text{-}|X_{e2}|)|+r)<X_{i2}< (X_c\text{-}0.25W)\}$,

Quadrant 3:

$Q_3 = \{P_{i3} \mid (X_c\text{-}|X_{e2})|+r)<X(P_i)< (X_c\text{-}0.25W)\}$,

Quadrant 4:

$Q_4 = \{P_{i4} \mid (X_c+0.25W) <X_{i4})<(X_c+X_{i4}\text{-}r)\}$,

Where $P_{i1}(X_{i1},Y_{i1}, Z_{i1})$, $P_{i2}(X_{i2},Y_{i2}, Z_{i2})$, $P_{i3}(X_{i3},Y_{i3}, Z_{i3})$ and $P_{i4}(X_{i4},Y_{i4}, Z_{i4})$ are points on a cross section of the human body. Points $P_{e1}(X_{e1},Y_{e1},Z_{e1})$ and P_{e2} (X_{e2},Y_{e2},Z_{e2}) are two extreme points of the cross section (Figure 5.7). And r is a small value obtained through experiments, usually ,0<r<50 mm (Figure 5.7).

$f(\Delta Z_i) = 1,\ \Delta Z_i >0,$

$f(\Delta Z_i) = -1,\ \Delta Z_i <0,$

$\Delta Z_i = Z_i - Z_{i-1},\ (i = 1,2,...M),$

Where point $P_i(X_i,Y_i,Z_i)$ and $P_{i-1}(X_{i-1},Y_{i-1},Z_{i-1})$ are two consecutive points on the cross sections with the total point number of M.

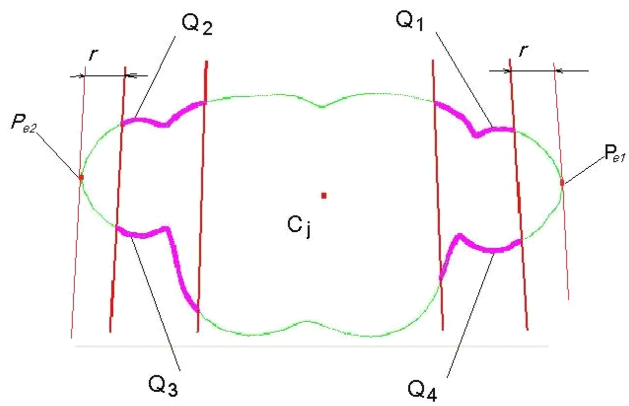


Figure 5.7 Four quadrants slice with a centroid

If the value of $f(\Delta Z_i)$ changes from 1 to -1 or from -1 to 1 at point P_i, then point P_i is labelled as candidate feature points.

Step 4: compute $min(Z_k)$ and all candidate feature points in each quadrant.

$$min(Z_k) = min(||Z|_i - |Z_c||), (k=1, 2, 3, 4) \quad (5\text{-}6)$$

So point P_k (X_k, Y_k, Z_k) are the four feature points on cross section j (Figure 5.7).

If there is no feature being found on one cross section, the procedure will proceed upwards and repeat the same steps on the next cross section. The first cross section that feature points are found on it will give the height for segment arms.

3) Legs segmentation

Legs are identified after finding out the crotch. The feature region of crotch is defined as:

$$R_c = \{P_i \mid 3/8H < Y_i < 4/8H, i = 0, 1, 2, \ldots m\}, \quad (5\text{-}7)$$

Then the data is processed step by step upwards starting from the bottom cross section.

Step 1: find the centroid $C(X_c, Y_c, Z_c)$ using equation 5-4 described above.

Step 2: determine the feature points with similar algorithm as step 3 in arm segmentation.

Two constraints X_l and X_r are provided for reducing the feature region on the cross section. So the two new feature regions (Figure 5.8) are defined as:

$$\begin{cases} R_{c1} = \{P_i \mid X_l<X_i<X_r,\ 0.45H<Y_i<0.52H,\ Z_i>Z_c\ i = 0, 1, 2, \ldots m\}, \\ R_{c2} = \{P_i \mid X_l<X_i<X_r,\ 0.45H<Y_i<0.52H,\ Z_i<Z_c\ i = 0, 1, 2, \ldots m\}, \\ X_l = X_c - t, \\ X_r = X_c + t, \end{cases} \tag{5-8}$$

Where, $P_i(X_i,Y_i,Z_i)$ are points on the cross section and t is a small value, usually 20mm<t<50mm. Point $C_j(X_c,Y_c,Z_c)$ is the centroid of the cross section, and j means the *jth* cross section of the human model.

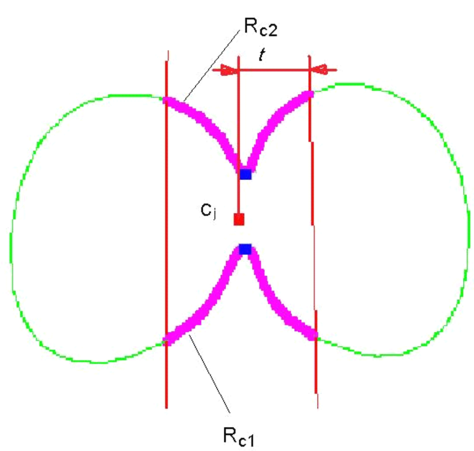


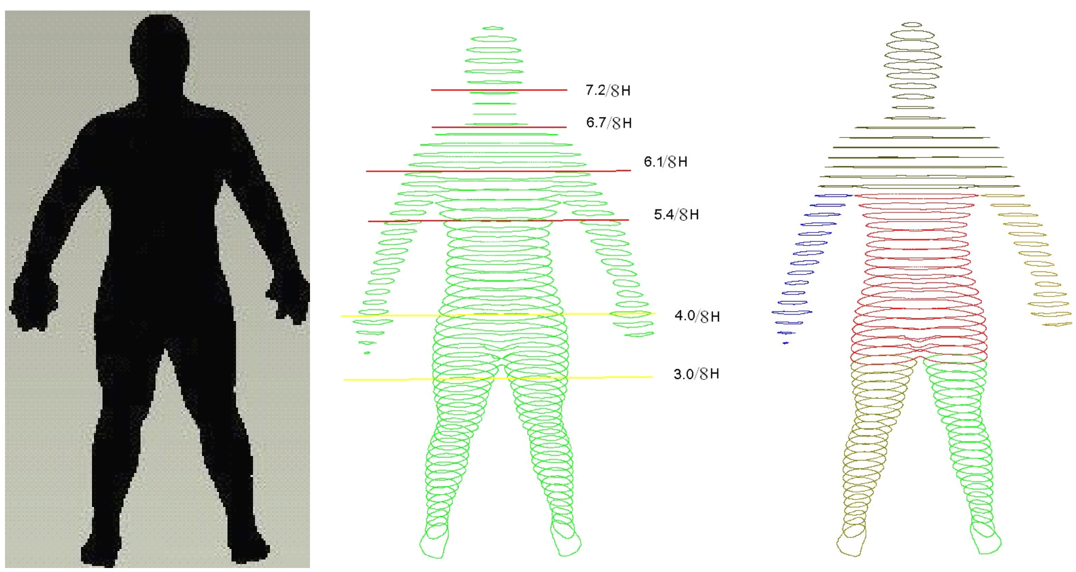
Figure 5.8 Crotch slice with two identified feature regions

Step 3: search candidate feature points.

Candidate feature points are determined in those two new feature regions by applying the sign function related in step 3 of arm segmentation.

Step 4: Feature points extraction

Two feature points (blue points in Figure 5.8) are obtained by repeating respectively the same method described in step 4 of arm segmentation in those two new feature regions. Once finding out these two feature points, it is easy for us to get leg data automatically (Figure 5.9 (c)). Figure 5.9 (a) shows a point cloud human model. Figure 5.9 (b) demonstrates feature regions and (c) is a segmentation result illustrating different parts in different colours.



(a) The original point cloud with 2,001,379 points. (b) The slice model. (c) The result of segmentation.

Figure 5.9 Body segmentation

5.3.3 Point Sampling on Elliptic Cross Sections

After segmentation, an approximation (a new simplified model) of the point-based human body model is created by generating cross sections for each body part. This simplification process is controlled by a set of parameters regarding the LOD application.

We classify cross sections of human body into two categories according to the complexity of shape. The shape of cross sections on limbs and the torso except those including shoulder area can be considered as an irregular ellipse with a simple topology. And cross sections including shoulder area are more complex with dramatic curvature changes above arm pits. Therefore different algorithms are applied on different topological cross sections.

For cross sections with simple elliptic shape, a three-step algorithm is applied.

Step 1: *calculate centroid C.*

Assuming a cross section consisting of n raw data points, its centriod can be computed by using equation 5-4 presented above. A centroid C (Xc, Yc, Zc) is established on each cross section, which is the origin of the local coordinate system and then it is used to divide the cross section into sectors.

Step 2: *create sectors and sample points.*

Each cross section is divided into a number of sectors with the angle φ, represented by areas between two black lines, which determine a number of the samplings. The yellow point is in the area (between the black line and green line) with tolerance τ. Constraints (angle φ and τ) among points on arms, legs and lower torso are found out during point sampling (Figure 5.10). Different constraints lead to different sampling density. This algorithm starts with a random point i and searches for a point j in anticlockwise direction which gives an angle close to φ. This is controlled by a small tolerance τ, which meet the following requirement:

$$\min \left\| (1 - \frac{A}{\varphi} \right\| < \tau \qquad (5\text{-}9)$$

Where A is an angle among the start point, the centriod and the candidate point. If there are two points on both sides of the black line, the point in clockwise direction is prior to the other one.

The identified point will be labelled as a new boundary point. For it, the division process will be repeated until examining all points.

For cross sections with shoulders, four feature points are found out first and marked as boundary points (See subsection C for details). Then the divide process will start with these feature points.

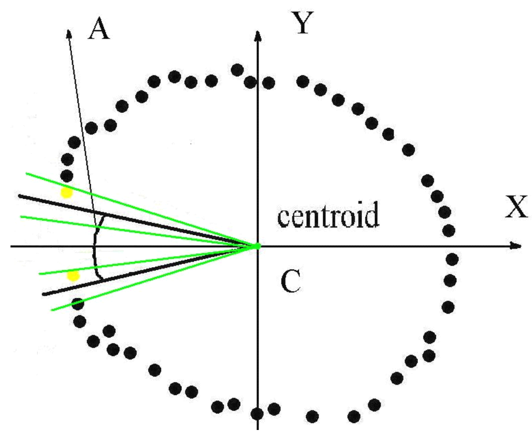


Figure 5.10 Point sampling on cross sections

5.3.4 Point Sampling on Cross Sections in Shoulder Areas

For cross sections with more complex topology, we apply a four-step method.

Step 1: calculate the centroid C_2.

We use the same method described in subsection B to get the centroid C_2.

Step 2: find out four feature points.

After find out the centroid C_2, the cross section is divided into four quadrants with C_2 as the origin. There will be only one feature point in each quadrant. We use sign function to search all candidate feature points, which are called turning points. Then through calculating the minimum value of z coordinate of turning points, we can obtain the feature point of each quadrant.

$$\begin{cases} f(\Delta z_i) = 1, & \Delta z_i > 0 \\ f(\Delta z_i) = -1, & \Delta z_i < 0 \end{cases} \tag{5-10}$$

Where, $\Delta z_i = z_i - z_{i-1}$, $(i = 1,2,...M)$, M is the number of point in this quadrant, and P_i (x_i, y_i, z_i) are points on the cross section. If the value of $f(\Delta z_i)$ changes from 1 to -1 or from -1 to 1 at point P_i, then point P_i is the turning point. Among all the turning points that we get after go through all points in one quadrant on this cross section, the point with minimum z coordinate is the feature point in this quadrant. That is:

$$P_f = \min(P_j), (j = 1,2,...N) \tag{5-11}$$

Where P_f (Figure 5.11)is the feature point and N is the number of turning point in one quadrant. Through repeat of this method we can obtain four feature points, which are then marked as boundary point to divide the cross section into three sub-parts with relatively simple topology.

Step 3: calculate centroid C_1 and C_3.

In each sub-part divided by four feature points, we can get a centroid (C_1 and C_3 in Figure 5.11) with the above equation 5-4.

Then three local coordinate systems are established with the completion of the three controids. Points are sampled within each sub-part, which definitely improve the accuracy of samplings.

Four feature points (P_{f1}, P_{f2}, P_{f3}, P_{f4}, green points in Figure 5.11) are found out by searching the sharp angle formed by three neighbouring points P_{i-1}, P_i and P_{i+1} with following rules. Points are sampled by setting up three local coordinate system based on three centroids (C_1, C_2 and C_3 in Figure 5.11) in areas consisting of shoulders in order to improve the accuracy of samplings.

Step 4: create sectors and sample points.

This step, which is carried out in each of the sub-parts, is the same with the method described in step 2 of section 5.3.3.

Through repeat of this simplification with simple changing φ and τ, the number of samplings is iteratively reduced, and sequences of approximations to the original point cloud human model are generated. Figure 5.12 shows the original cloud point human model and sampled model produced with different values of φ.

Users are capable of controlling the sampling resolution through a couple of parameters including the angle of the sector φ, and the tolerance τ. And through the use of parameters as constraints, the order and structure are preserved during the procedure of sampling, which can improve the geometric surface at low levels of detail.

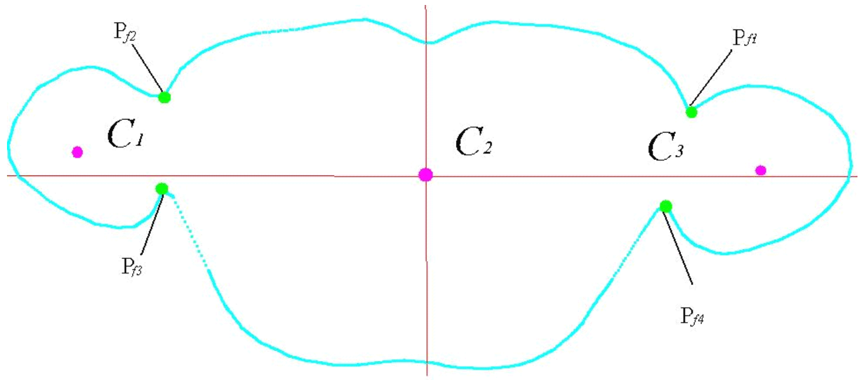
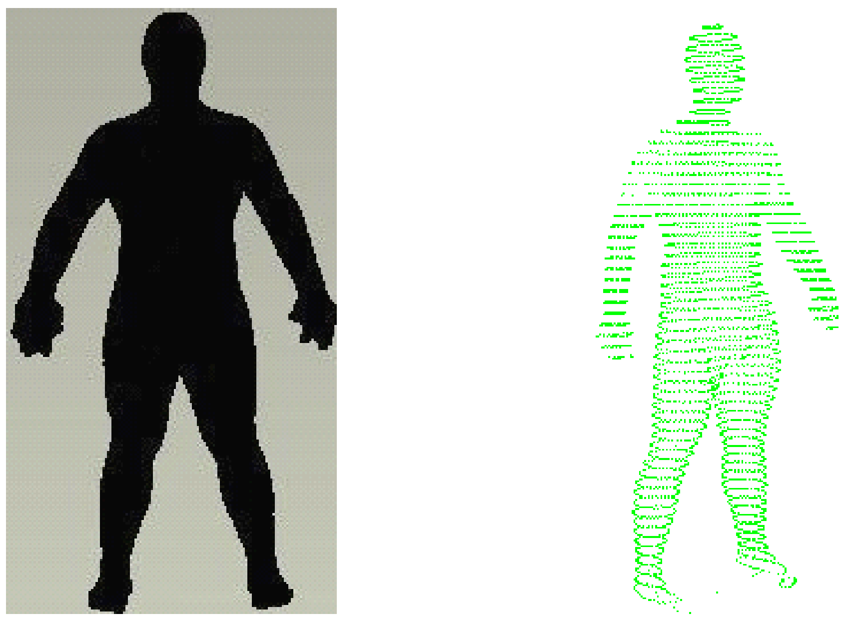
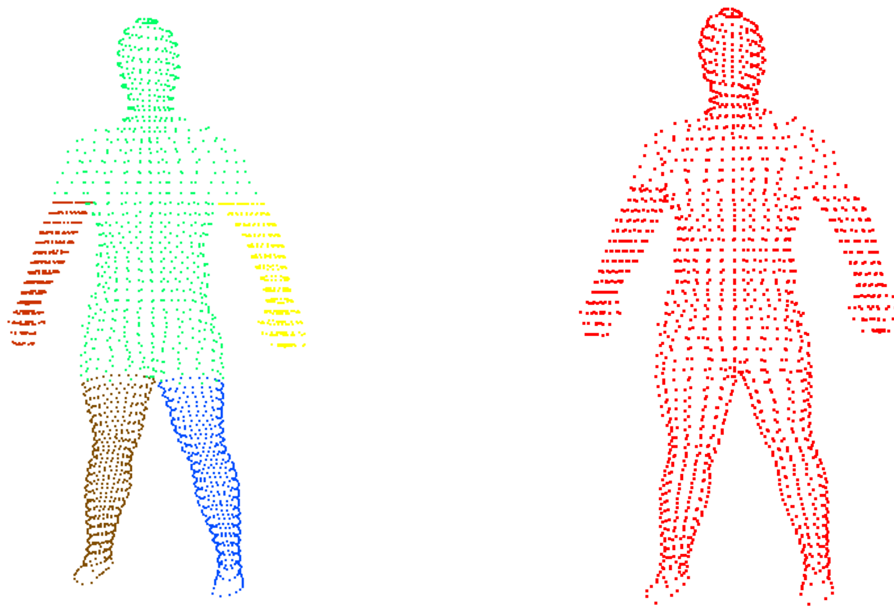


Figure 5.11 Calculation of cross section centroids



(a) Original data cloud model: 2,001,379 points (b) Simplified: 3886 points.



(c) Simplified: 2592 points. (d) Simplified: 1296 points.

Figure 5.12 Model simplification

5.3.5 Surface Representation

Two steps are used to construct a human model surface as follows:

Step 1: Body part surface construction.

After segmentation and point sampling, each segment contains a stack of cross sections. Each cross section consists of a series of sampled points. The correspondence meshing algorithm [Meyers1992] is applied to each part. And then two related parts are sewing up through connecting the adjacent cross sections. Figure 5.13 (a) illustrates the sewing algorithm and a mesh construction results are shown in Figure 5.14 (b)-(f). The data structure used for the constructed mesh is described in Chapter four.

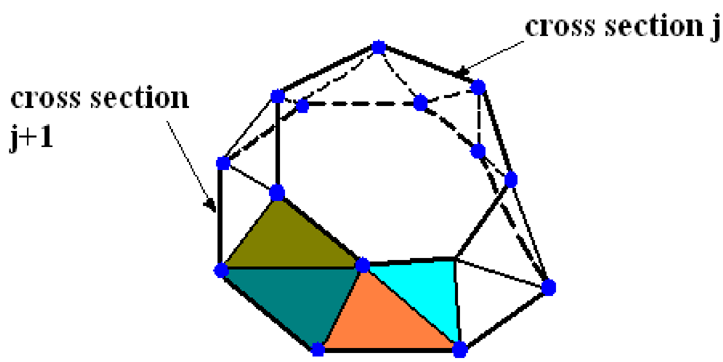
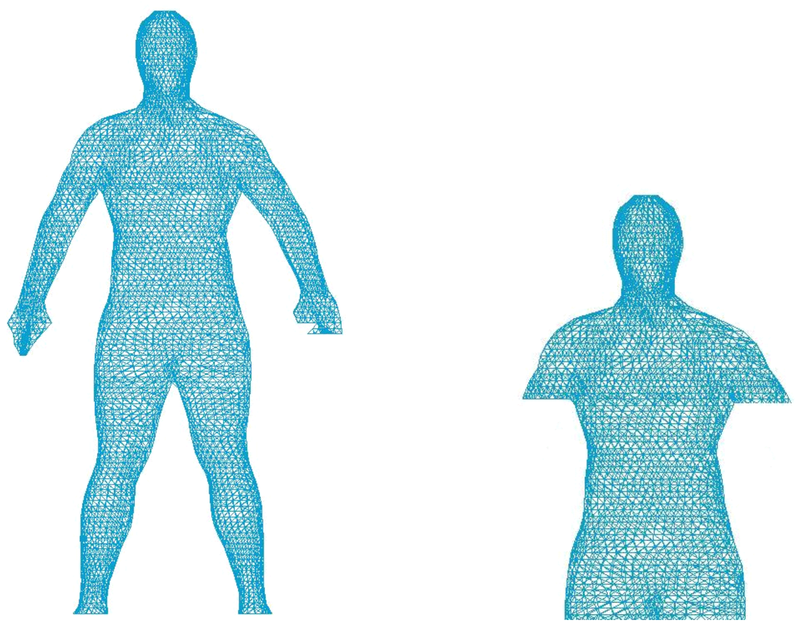
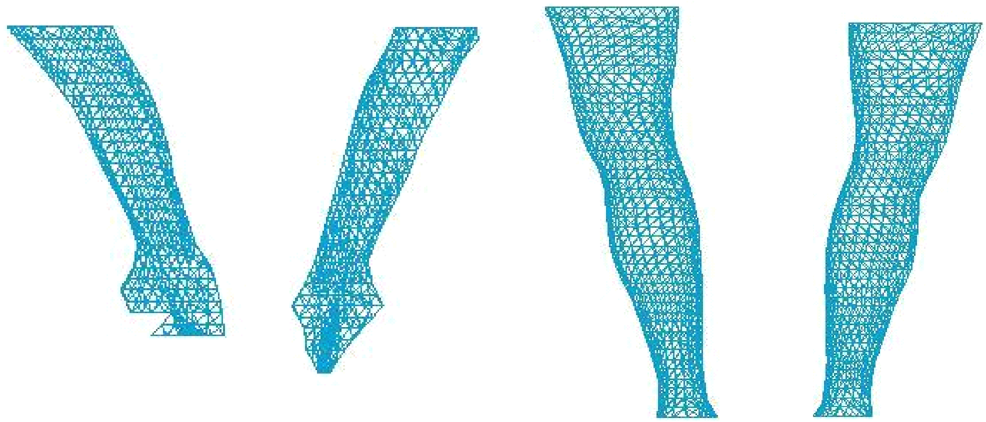


Figure 5.13 Cross section connection



(a) Human model.   (b)   Torso



(c) Left arm (d) Right arm (e) Left leg (f) Right leg

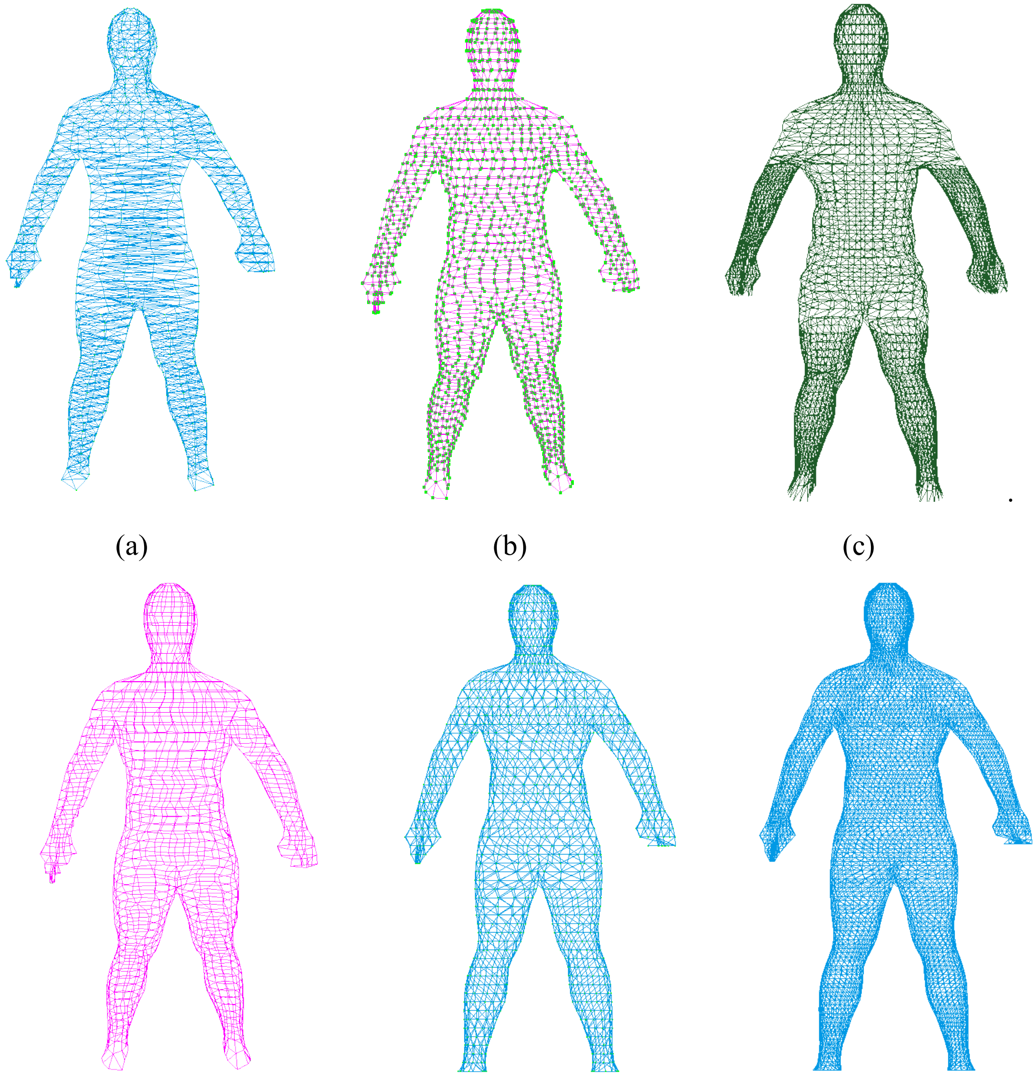
Figure 5.14 Mesh of body parts

Step 2: Part sewing.

A complete human model represented by triangular mesh is constructed by sewing up two adjacent segments (Figure 5.14 (a)).

5.3.6 Results

Figure 5.15 shows some results of human models with different levels of detail.



(a) (b) (c)

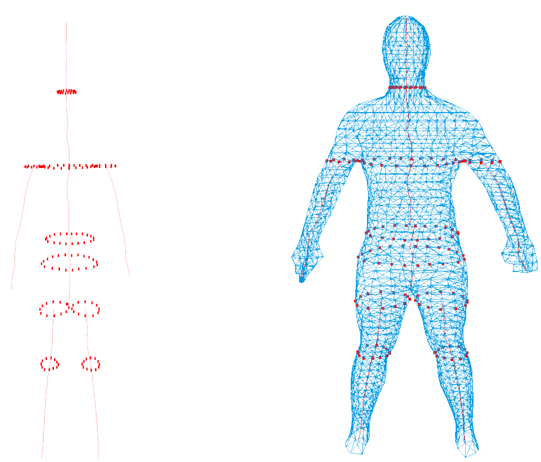
(d) (e) (f)

Figure 5.15 Human models with different LODs

5.4 Body Feature Extraction

The raw body data captured by a 3D scanning system cannot be used directly because little semantic information can be gained by the raw point cloud. The raw body data needs to be processed and converted into a human model with useful features. After cleaning and slicing the scanned data using Geomagic [Geomagic], we segment the human model into six parts: chest, torso, left arm, right arm, left leg and right leg (see section 5.3.2). Then, an algorithm is used to extract the features and a reference skeleton of the human body.

The Tailor system [Moccozet2004] can identify various features such as sharp protrusions, wells and dips. The centroids corresponding to some regions are located by circles, which globally correspond to the anatomical joints of the body. In our method, the extraction of reference skeleton and key features is based on the target and application instead of only in regions of anatomical joints of the human body. With the aim of dressing the human model, we create some key features and one reference skeleton to constrain the dressing process. A semantic mesh model with key features and a reference skeleton is shown in Figure 5.16 (b).



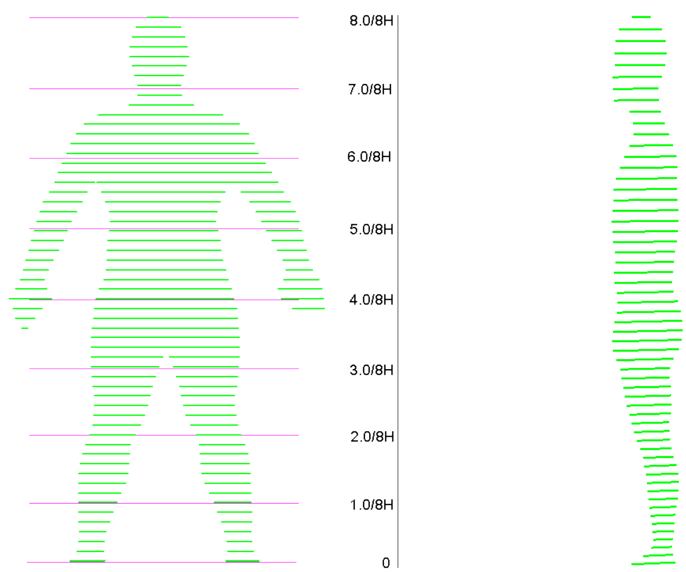
(c) Features & reference skeleton. (d) Semantic mesh model.

Figure 5.16 Semantic human model.

5.4.1 Key Feature Extraction

We extract features in two steps: 1. Definition of feature regions. 2. Calculation of the extreme length/point in each feature region.

Due to differences in body shape among various people, especially different ethnicities, it is impossible to locate key body features with an absolute proportion that applies to every human body. We locate body feature regions through some pre-defined parameters described by the proportion of head to body height. We adopt the idea of describing the body height as eight-heads tall (Figure 5.17 (a) and (b)). Therefore, we first define feature regions trying to give a flexible description of features based on the literature [Ratner2003]. This speeds up the processing and has proven very efficient, because feature regions reduce the number of candidate points of features. Then, we extract features within feature regions previously defined. More features can be calculated or generated for different details.



(a) Projection on the XOY plane. (b) Projection on the YOZ plane.

Figure 5.17 Projection of a human model.

In the following section, we describe how to extract various features in corresponding feature regions for the purpose of garment generation.

Generally speaking, each feature is obtained by calculating the extreme point or extreme length of feature girth in a given feature region. Feature girth is defined as the minimum or maximum circumference of cross section in the feature region. The feature girth is the sum of arc length formed by a set of points on a cross section, each of which is on a horizontal plane parallel to the XOY plane (Figure 5.18). The length of a feature girth is defined as the circumference L*i* of the cross section *j*:

$$L_j = \sum_{i=1}^{n} (l_i * (\pi / 180) * \theta_i) \quad l_i = (r_i + r_{(i+1)}) / 2 \quad r_i = \| p_i - c_j \| \qquad (5\text{-}12)$$

Where θ_i is the angle between the neighbouring points on the cross section j and c_j is the centroid of the cross section. The symbol p_i represents a 3D coordinate of a point on the cross section. Alternatively, the Euclidean distance of each two neighbouring points on a cross section can be used instead of arc length. This makes sense when points on a cross section are dense and very close to each other. However, when a coarse model is generated, the error becomes significant when using Euclidean distance between two neighbouring points.

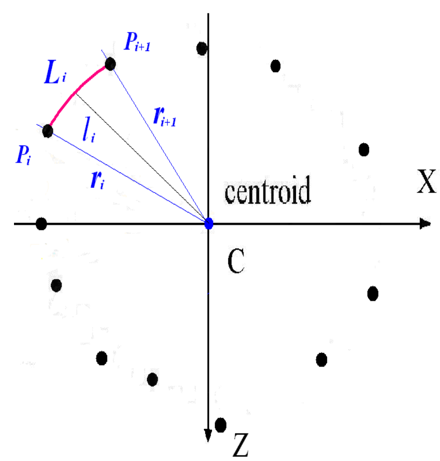


Figure 5.18 Feature girth calculation

We give definitions of some concepts used in feature extraction before describing the algorithm. The height of the human model is represented by *H*. Feature regions of neck, chest, waist, wrist, hip, crotch and ankle are represented by reg_n, reg_c, reg_wa, reg_wr, reg_h, reg_cr and reg_ak respectively. Neck girth and waist girth are the minimum length of cross-sections in their corresponding regions. Chest girth and Hip girth are defined as the maximum length of cross sections in their corresponding regions. Table 1 shows features, feature regions, constraint types and their denotations of an example human model. Nipple is defined as the point with the maximum z value in the chest region. Key body feature locations change slightly on different human models with different levels of detail. Figure 5.19 shows an example of body features. Smaller red points represent features of neck, waist, wrists and ankles. And the larger red and green points denote the features of nipples, scapulas and the hip. Key features might not have exactly symmetric values on left and right parts of the body because the left and right parts of the body may not be exactly symmetric.

Table 3 Key body feature explanation

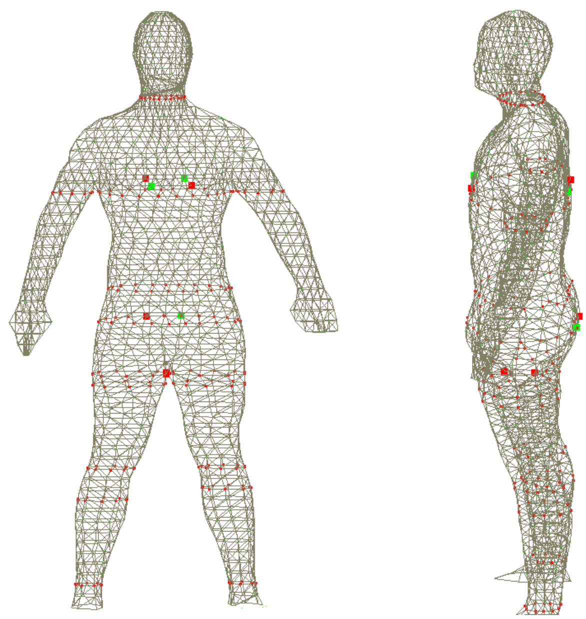
Key Feature	**Feature region symbol**	**Constraint type**	**Denotation**
Neck	reg_n	boundary	b_neck
Nipple	reg_c	Front hanging points	h_nipplel, h_nippler
Scapula	reg_c	Back hanging points	h_scapulal,h_scapular
Waist	reg_wa	Interior boundary	b_waist
Wrist	reg_wr	boundary	b_wristl, b_wristr
Hip	reg_h	Back hanging points	h_hipl, h_hipr
Crotch	reg_cr	Boundary points	b_crotch
Ankle	reg_a	Boundary	b_anklel, b_ankler

Here are some explanations of concepts used in the above table:

Boundary: A boundary constraint marks the boundary for 3D construction. If a stroke (outer contour curve) is beyond the boundary, the segment of the stroke beyond the boundary references the boundary for 3D construction.

Hanging Point: It represents the maximum or minimum value along z-axis of a body segment where a garment touches the human body.

Interior boundary: It marks an interface for dramatic change of the body shape. Some pre-defined thresholds need to change accordingly, such as curvature and curve energy.



(a) Front view of a human mesh model with key body features. (b) Side view of the mesh model with key body features.

Figure 5.19 Human models with key body features

5.4.2 Reference Skeleton

A reference skeleton is defined by connecting centroids of each cross section (Fig. 3 c). Thus each segment between two neighbouring centroids (C_i and C_{i+1}) is a straight line. We can calculate the unit vector of each straight segment easily by the following function:

$$u = (C_i - C_{i+1})/\| C_i - C_{i+1}\|. \qquad (5\text{-}13)$$

With the help of u, each point at any height on the reference skeleton can be obtained by $C(i) = (x_i, y_i, z_i)$. Points on each cross section can be indexed by a reference coordinate system with the origin at the centroid. Figure 5.20 shows a skeleton model extracted from a human model.

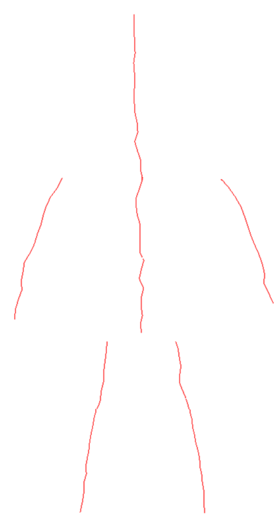
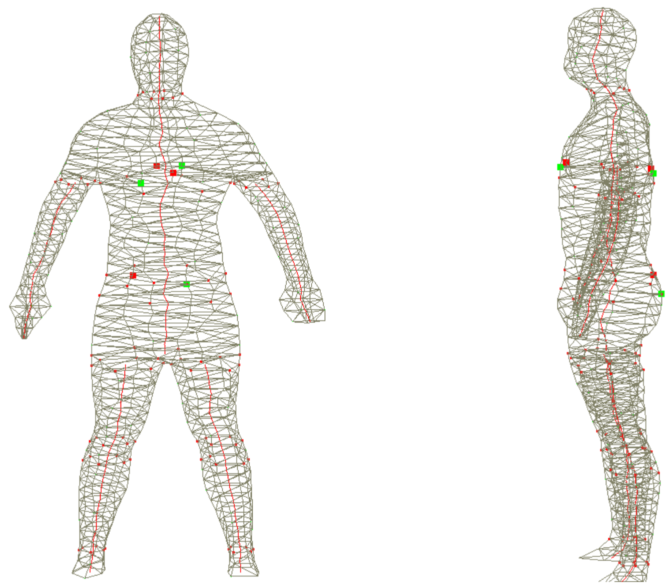


Figure 5.20 A skeleton model

5.4.3 Testing Results

Figure 5.21 shows a mesh model with feature points and a skeleton.



(a) A front view model with key body features being extracted from a lower LOD model. (b) Side view of a model with lower LOD with key body features.

Figure 5.21 Human models with key body features

5.4.4 Conclusion

This chapter presented how to construct semantic human models from body scan data. We segment a human body data into six parts. Each part is reconstructed individually and represented by a mesh surface. A human model is reconstructed by sewing up all the six parts. A skeleton and key body features are extracted based on the application of the human model, which is for garment generation. In chapter 3, we have demonstrated how to use a semantic human model to generate some simple garments. In chapter 6, we will show how to create garments from sketches around a semantic human model.

Chapter 6. Dress Virtual Humans from Sketches

6.1 Introduction

This chapter focuses on an approach for dressing virtual humans by sketching a garment contour around a 3D semantic human model. The input is user-drawn sketches, which represents the front-view shape contour of a 3D garment and the output is an elaborate 3D geometric garment surface dressed on a virtual human body. The procedure of the 3D garment construction is constrained by key body features of the semantic human model, which have been identified and defined in the previous chapter.

The framework presented in this thesis shares similar goals with the two methods proposed by Wang et al [Wang2003B] and Turquin et al [Turquin2007], but there are some important differences. We use human body features as constraints during generation of 3D garment vertices. To prevent garments from exactly following the body shape after inferring garment vertices based on distances to the body in 3D space, we use a level set-based method to change the curve shape of garments. Garment surface is refined using mesh smoothing and subdivision schemes, which improves the visualization effect and gives flexibility for further processing such as providing 3D decoration tools. A flowchart of the general procedure is shown in Figure 6.1.

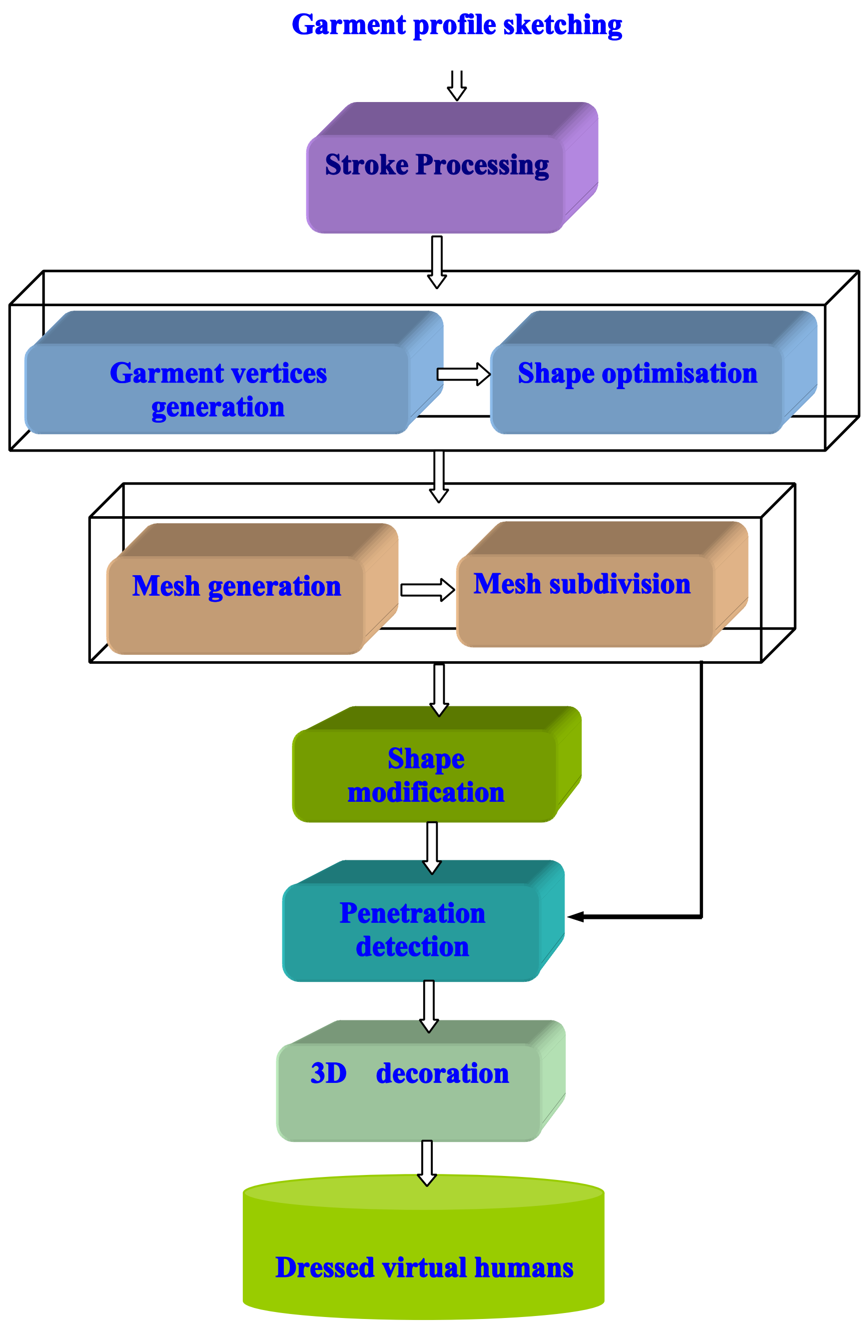


Figure 6.1 Flowchart of garment construction by sketching profiles

We design our system based on sketch interface aiming to satisfy people's demand for dressing 3D virtual humans easily. The system takes user-drawn strokes as input and generates an elaborate 3D geometric garment surface dressed on a virtual human body.

To use the system, users draw 2D garment profile strokes on the screen, and the system automatically processes these strokes and generates elaborate garment surfaces dressed on a virtual human. Important properties of the proposed method are as following:

Semantic human model: Semantic human model is a reconstructed human model from real human scan with some key body features, which are used as constraints during garment generation to ensure garments, are generated and hung on the human body realistically.

Sketch interface: Users draw strokes on 2D screen and these strokes are processed into spline curves, which represent the front view shape of a garment. Through dragging control points into new positions, the garment shape can be modified accordingly.

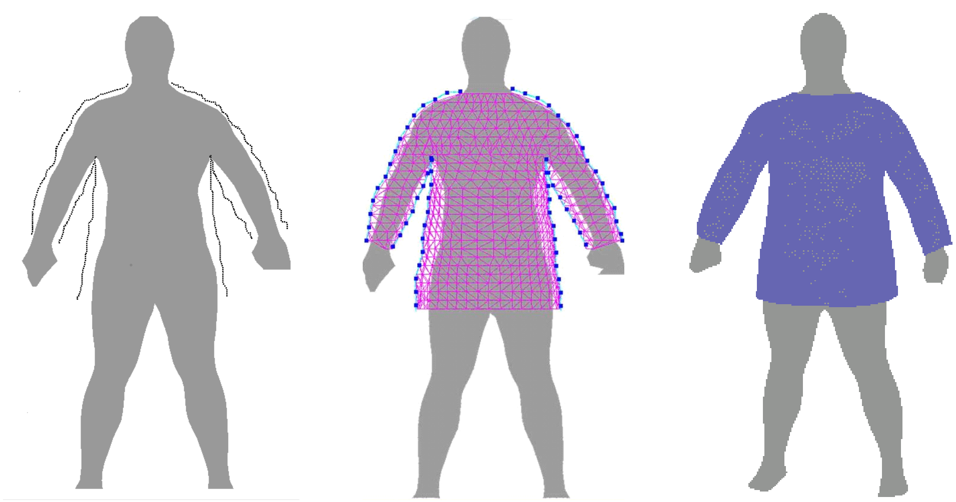
Garment surface generation using distance field and level set-based method: Interior garment vertices are inferred non-linearly based on the distance from outer curves to the human body with key body features as constraints. A level set-based method is developed to optimise garment shape and remove artifact.

Garment surface refinement: Elaborate garment surface is obtained by applying some mesh refinement schemes, which improves the visualisation results and enables to apply some further mesh editing tools such as decoration tools.

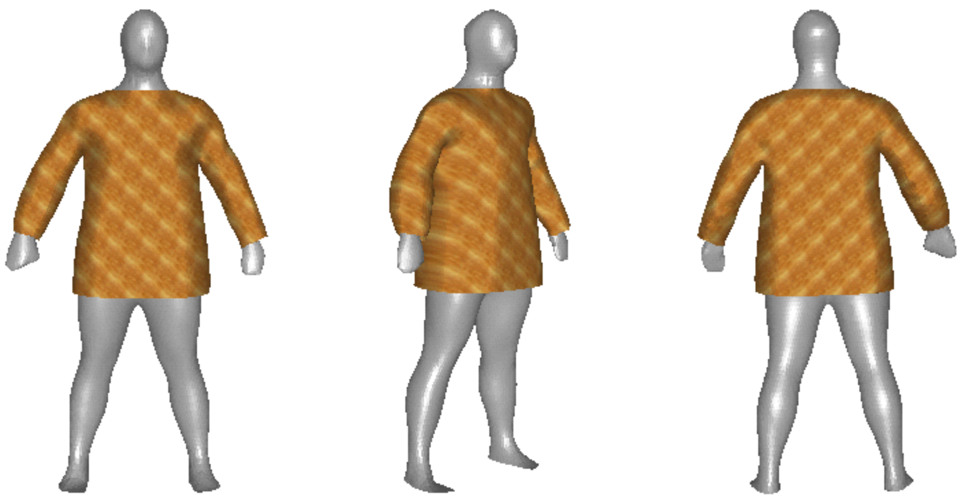
Decoration Tools: Two 3D decoration tools including 3D brush and 3D embroidery tool are provided, which allow users to add some 3D details to the garment surface by sketching directly on the garment surface. The decoration details are represented by refined mesh surface.

6.2 Overview of the Dressing Method

We have developed a new sketch-based platform to facilitate the modelling of human body and virtual garments. The input of the system is a 2D garment sketch profile representing the front view of a 3D garment. The output is an elaborate 3D garment geometric surface. The system first reconstructs semantic human models from real human body scan data obtained using a body scan system. Users draw 2D sketches around the semantic human models based on a sketch interface, which represents the front view shape of a garment. Then a 3D garment surface is generated from the 2D sketches after a series of processing (Figure 6.2).



(a) Sketch the profile of a sweater. (b) Complete mesh construction. (c) Mesh surface after refinement.



(d) Front view of rendering. (e) Side view of rendering. (f) Back view of rendering.

Figure 6.2 An example of a sweater generation

The sketching interface processes the input user-drawn strokes into Cardinal Spline curves, which are associated with a human body segment. These 2D Cardinal Spline curves are converted into 3D curves in the human model space, which are used to obtain the distance to the human model.

The human modelling framework reconstructs human models from scanned body data obtained using a laser scan system [TC]^2. Reconstructed human models are represented by polygon mesh surfaces. A human model has been segmented into six parts before constructing polygon mesh. Semantic human body features are extracted, which are used in garment surface construction.

When a human model wears a jacket, it neither always falls apart from the body nor always touches the skin. It actually touches some patches of the body and drapes downwards.

Our system convert garment profile curves into a proper 3D garment surface using the following five steps.

1. Garment’s vertices generation: Rough 3D vertex position is determined based on garment contour curves. Two methods are developed for inner contour curves and outer controur curves separately.

2. Shape optimisation processing: Rough vertex position is optimised to eliminate artefacts on garment surface.

3. Garment mesh generation: A mesh algorithm is applied directly on 3D garment points to construct a coarse mesh of garment segments, which are sewn up to obtain a complete garment surface.

4. Garment mesh processing: Elaborate mesh is obtained by applying smoothing and subdivision schemes.

5. Shape modification: Through dragging control points of spline curves, which represent the front shape of a garment, a garment shape can be modified without efforts to re-sketch a new one.

6.3 Garment‘s Vertices Generation

The generation of garment's vertices is based on the distances between these garment contour curves and the human body. Different algorithms are developed to generate 3D garment’s vertices for two kinds of curve: inner contour curve and outer contour curve (Figure 6.3 (a)).

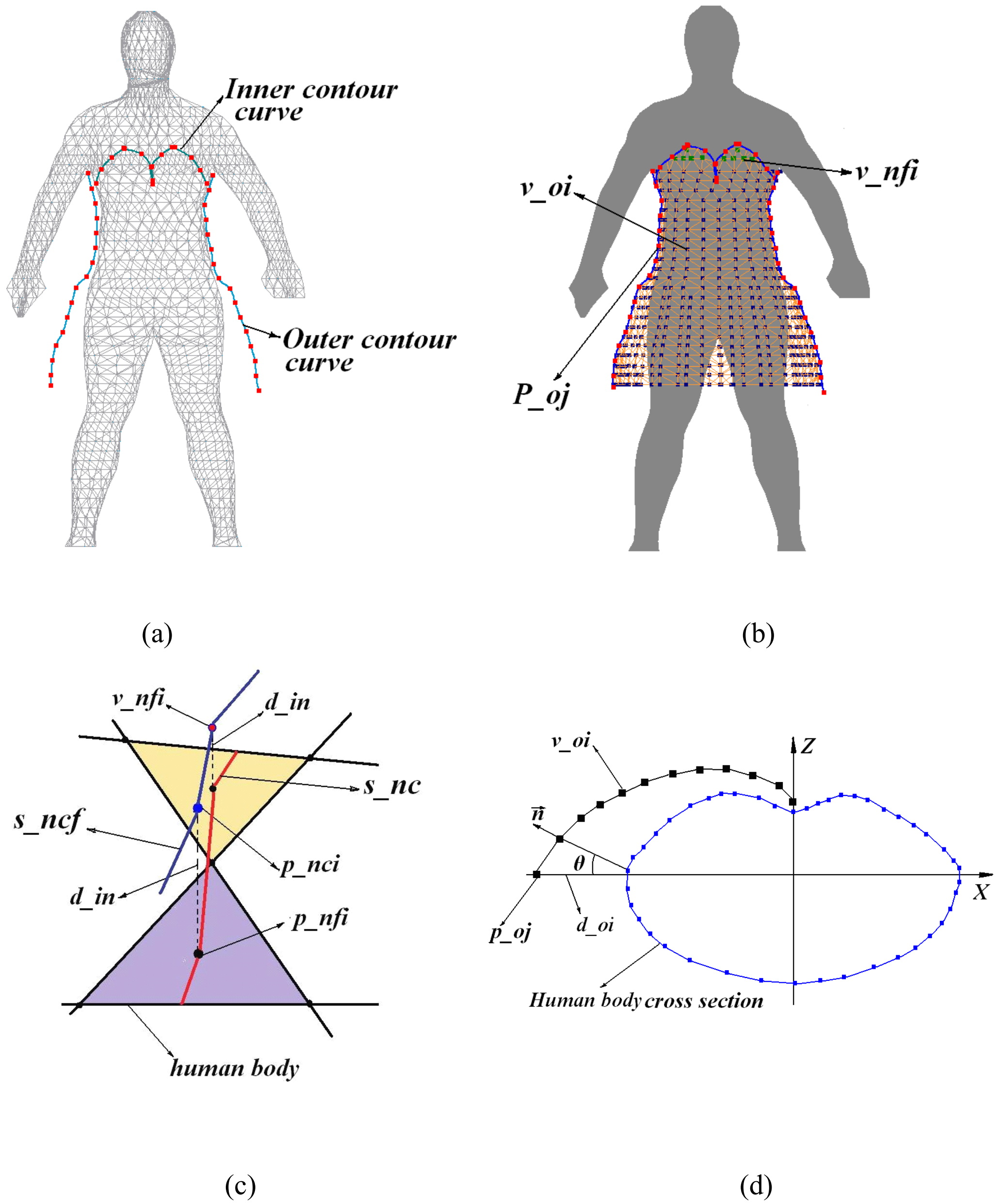


Figure 6.3 Contour curves and garment vertices

6.3.1 Sketch Recognition

Much research on sketch recognition has been done and many systems have been developed over the last years [Naya2002, Olsen2009]. In this thesis, sketch recognition is defined as association of sketches to a body segment.

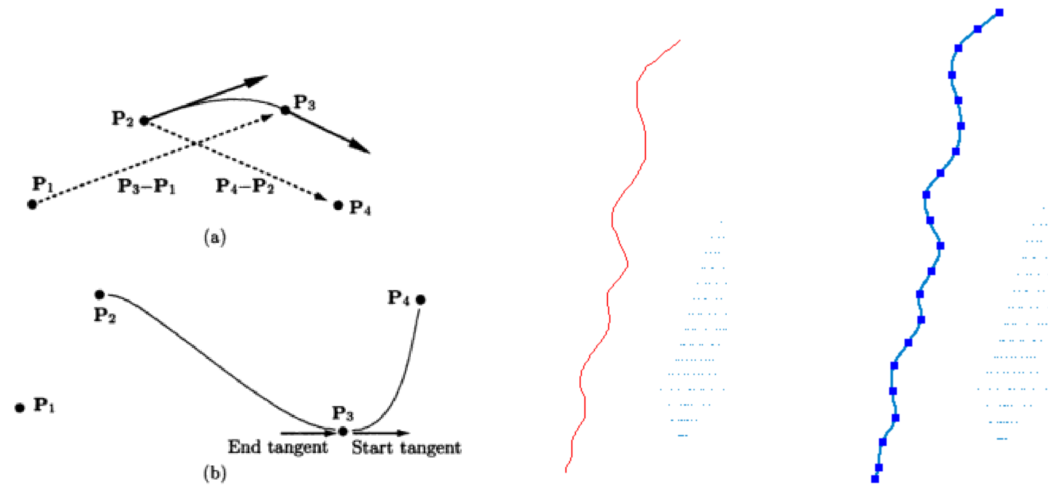
A user-drawn sketch is associated with a body segment through sketch recognition. Normally, each sketch of a garment profile contains multiple strokes. Therefore, through the procedure of sketch recognition, one human body segment can associate with more than one strokes.

Sketch recognition is based on the distance from a stroke to the human body. After completion of a sketch, distances between the sketch and all body parts are calculated. The stroke is always associated with a body part with the minimum distance. Only these points of a stroke which are within the body part region are calculated. A body part region is defined in terms of x-values of a body part and stroke points as follows:

Suppose the top point of a body part is q_{ti} and the bottom point is q_{bi}. For a point p_j of a stroke S ($p_j \in S$), if $Yp_j \geq Yq_{bi} \cup Yp_j \leq Yq_{ti}$, then, we can say that point p_j is in the region of the body part.

6.3.2 Spline Curve Fitting

The user-drawn sketchs are converted into 2D curves using Cardinal Spline fitting [Salomon99] (Figure 6.4 (a) and (b)). A Cardinal Spline is a series of individual curves between points joined to form a larger curve. It is actually a polynomial spline function with equally spaced knots. The spline is specified by an array of points and a tension parameter. A Cardinal Spline passes smoothly through each point in the array. Each point is involved in only four curve segments, which make it ideal for local control, because moving one control point of a spline curve only affects those related segments instead of the entire curve.



(a) Tangent vectors in a Cardinal Spline. (b) A sketching stroke. (c) Cardinal spline fitting with control points

Figure 6.4 Spline curve fitting.

One of the major advantages of cardinal splines is that cardinal splines have essentially only one B-spline of a given order. All the others of the same order are translates of the first one. There are no sharp corners and no abrupt changes in the tightness of the curve. By passing through every point, there are no missed points that can cause errors when fitting a surface over the splines. The individual segments connect with each other smoothly. And because the derivatives at the joint points are equal, the curve has C^1 continuity. However, generally the second derivatives of the segments are different, which means that the curve is not C^2continuity. Figure 6.4 (a) shows that the end tangent of the segment for point group $\{P_0, P_1, P_2, P_3\}$ is identical to the start tangent of the next point group $\{P_1, P_2, P_3, P_4\}$.

A segment $P(t)$ is defined as following:

$$P(t) = (t_1, t_2, t, 1) \begin{pmatrix} -s & 2-s & s-2 & s \\ 2s & s-2 & 3-2s & -s \\ -s & 0 & s & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix} \tag{6-1}$$

$$P(0) = P_2,\ P(1) = P_3,\ P(0) = s(P_3 - P_1),\ P(1) = s(P_4 - P_2)$$

The tension parameter T is defined as following:

$$T = 1\text{-}2s \qquad s \in [0,1] \tag{6-2}$$

The tension of a Cardinal Spline determines the fitting of each curve section in regards to the data points. The range for the tension value is 0.0 to 1.0 where 1.0 would deliver straight lines between the data points and 0.0 would produce the most flexible curves fitted through each data point. In our implementation, we set the tension parameter as 0.5.

There are several advantages brought by the curve fitting.

First, noises of the raw strokes have been filtered.

Second, the large number of raw stroke points has been significantly reduced and well distributed with evenly spaced intervals, which highly speeds up 3D point construction.

Thirdly, users are allowed to draw as many strokes as they like corresponding to one body segment. These strokes do not need to be drawn in sequence. And it allows users to leave a gap between two strokes. The system can group and fit these strokes with one spline.

Finally, due to the property of Spline curve, it can be easily modified by users to obtain different shape of garments. Figure 6.4 (c) shows a cardinal spline fitting result with control points and Figure 6.4 (b) is the original sketching stroke.

6.3.3 Curve Classification

After curve fitting, these curves are classified into two categories: inner contour curve donated by *s_nc* and outer contour curve represented by *s_oc* (Figure 6.3 (a)). Each curve is associated with a body segment after sketch recognition.

If a curve intersects with the human body and the intersection points exceed a certain number, it is classified as inner contour curve otherwise it is outer contour curve. As shown in Figure 6.3 (a), the curves constitute a skirt strap (on the body) are inner contour curve and those which are away from the body are outer contour curves.

These outer contour curves are projected onto xy-plane in world space and then are converted into 3D points sequences. Once the 3D points of the human model *M_h* are projected onto xy-plane, which are donated by *M_hp*, distance between the *s_oc* and related parts of *M_hp* is calculated. An outer contour curve is always associated with a body segment with the minimum distance between them.

During curve recognition, the distance from the corresponding body part and the curves have been stored. It is used in the procedure of 3D garment vertices construction later on, which highly speeds up the 3D construction procedure.

6.3.4 Garment Vertices from Inner Contour Curves

Generation of garment's vertices from inner contour curve is straightforward using the following 2 steps.

Step 1: Get the boundary.

Through using ray tracing techniques, a set of points denoted by $s\_nc = \{p\_nc_1, p\_nc_2, \ldots p\_nc_i \ldots\}$ are obtained by the intersection between inner contour curve and the human body. And then *s_nc* is offset to *s_ncf* by a distance *d_in* from the human body along each triangular faces that each point of *s_nc* is projected onto (Figure 6.4 (c)). The offset inner contour curve *s_ncf* is represented by $s\_ncf = \{p\_ncf_1, p\_ncf_2, \ldots p\_ncf_i \ldots\}$. The distance *d_in* is usually a small value, and it is subject to some predefined constraints. For example, *d_in* takes zero if the projected points offset from nipples using constriants *h_nipplel* and *h_nippler* or other hanging points, which means that the garment touches nipples or hanging poitns.

Step 2: Interior vertices generation.

Interior garment points, *v_nfi*, within the boundaries are interpolated using distance field of the human body (green points in Figure 6.3 (b)). The same distance parameter, *d_in,* is used during the interpolation (Figure 6.3 (c)). Constraints are applied to ensure that interior vertices do not lie outside of hanging points, which adopt the same principle as in step 1.

6.3.5 Garment Vertices from Outer Contour Curves

There are two steps for vertices generation from the outer contour curve. This method is applied to the curves corresponding to arms, chest, torso and legs.

Step 1: Calculation of 3D garment silhouette Position.

We use the same method as proposed by Turquin and Cani [Turquin2004] to infer the 3D position of garment silhoette. Calculate the distance d_o_i from the outer contour curve denoted by $s\_oc = \{p\_o_1, p\_o_2, ...p\_o_j...\}$, to the human body (Figure 6.3 (d)). The point depth of a outer contour curve is set to the z-value of the nearest point of the human body.

Step 2: Interior Vertices Generation.

To infer the rough position of interior vertices v_oi of a garment (Figure 6.3 (b)) based on the distance d_o_i obtained in step 1, we use the following equation to specify the distance between the body curve and the interior garment's vertices (Figure 6.3 (d)). Through experiments we find that this equation can generate a curve which can satisfy the needs for representing distance between the garment and the body.

$$dr_i = d\_o_i * \cos(\theta) * \lambda + \delta \qquad \text{(6-3)}$$

Where θ is the angle between the normal of a body curve at point p_i and the x-axis ($\theta \leq \pi / 2$), δ a small coefficient ($\delta \geq 0$). And λ is a coefficient used to adjust the distance through experiments, usually, $\lambda \leq 1$.

6.4 Garment Shape Optimisation Processing

Through interpolating 3D points based on distance to a human model and constraints, we can obtain a rough shape of a garment. In this part, we describe a technique to optimise the shape of garments to make garments look more realistic. We will demonstrate the ability of shape optimisation using a Level Set-based method.

6.4.1 Normal Calculation of Cross Section

Garment vertices obtained in previous section constitute a series of cross sections, which can be seen as a closed or open curve of 2D domain, represented by C_i. The normal of the cross section curve at a point p_j can be obtained by calculating the gradient of the cross section curve at this point, which can be achieved by the following equation:

$$\nabla C_i(x, y) = (\frac{\partial p_i}{\partial x}, \frac{\partial p_i}{\partial y}) \tag{6-4}$$

The cross section curve is actually a set of discrete points. The differential form in the continuous domain is replaced by the derivative in the discrete domain. The derivative of the curve at point p_j is approximated by the following equations:

$$\begin{cases} \nabla_x C_i(x_j, y_j) = p_i(x_{j+1}, y_j) - p_i(x_j, y_j) \\ \\ \nabla_y C_i(x_j, y_j) = p_i(x, y_{j+1}) - p_i(x_j, y_j) \;\; (\; p_j(x_j, y_j) \in C_i, p_{j+1}(x_{j+1}, y_{j+1}) \in C_i) \end{cases} \tag{6-5}$$

Where p_j and p_{j+1} are two neighbouring points on curve C_i.

The gradient is computed with the assumption that the relationship between two consecutive points of a cross section curve is linear. The gradient is regarded as the slope of the small segment of two neighbouring points.

The normal of a cross section curve at point p_j can be estimated using the following equation:

$$n = -\nabla_x C_i(x_j, y_j) / \nabla_y C_i(x_j, y_j) \tag{6-6}$$

The normal n is normalised by the following operation:

$$\hat{n} = \frac{n}{|n|} \tag{6-7}$$

6.4.2 Signed Curvature and Curve Energy

Signed curvature: Since the garment curve is represented by discrete sets of points, the curvature at point p_i can be obtained by the following equation:

$$\begin{cases} k(p_i) = \dfrac{2\sin(\frac{\beta}{2})}{\sqrt{\|v_1\| \bullet \|v_2\|}} \\ v_1 = p_i - p_{i-1} \\ v_2 = p_{i+1} - p_i \end{cases} \tag{6-8}$$

Where, p_i, p_{i-1}, p_{i+1} are three consecutive points on a curve. And β, the angle between the vector v_1 and v_2, can be calculated by the following equation:

$$\beta = \arccos\left(\frac{v_1}{\|v_1\|} \bullet \frac{v_2}{\|v_2\|}\right) \tag{6-9}$$

The signed curvature k^* is defined as follows:

$$k^* = \begin{cases} k(p_i), & \text{if point pi is a convex point;} \\ -\, k(p_i), & \text{if point pi is a concave point.} \end{cases} \tag{6-10}$$

Curve energy: Curve energy can be obtained by the following equation:

$$E = \sum k^2 \tag{6-11}$$

Where k is the curvature of curve s at a point.

6.4.3 Shape Processing Based on Level Set Method

We use level set-based method to achieve garment curve generation and curve shape optimisation.

The level set method devised by Osher et al. [Osher1988] is a versatile method for computing the motion of a curve. The purpose of this method is to calculate the subsequent motion of the curve using a velocity field. An oriented curve evolving in time is represented by the zero level set of a level set function. That is, in the level set method, curve $C(t):\Re^n \mapsto \Re$ represents implicitly the zero set of the following function:

$$\phi(C^*, t) = \pm d. \tag{6-12}$$

Where $C^* \in \Re^n$ and the variable d is a signed distance from C^* at time t to C(t). The orientation is denoted by defining the areas in which the level set function is negative in interior and positive in exterior.

The level set function Φ is defined in a domain Ω ($\Omega \in \Re^n$). The initial curve $C(t)$ is evolved based on a speed field F. This is usually represented by the following equation:

$$\frac{\partial C}{\partial t} = F\vec{N}. \tag{6-13}$$

Where F denotes the velocity and $\vec{N}$ is the normal of the curve. This velocity can depend on two kinds of properties: internal and external properties. Internal property includes curvature, position, the geometry etc. External property contains of time, external physics and so on. The curve is captured later time as the zero level set of a smooth function.

The curve is evolved in its normal direction with a speed F, which depends on solving the following differential equation:

$$\frac{\partial \phi}{\partial t} + F|\nabla \phi| = 0 \tag{6-14}$$

Where $\{\phi(C; t=0)\}$ defines the initial curve.

Mean curvature is widely used in evolving motion. An evolution model is proposed using mean curvature motion which is described by the following equation:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = g(|\nabla u|)|\nabla \phi|(div(\dfrac{\nabla \phi}{|\nabla \phi|}) + \nu) \\ \\ \phi(0, r, y) = \phi_0(x, y) \end{cases} \tag{6-15}$$

Where $g(|\nabla u|)$ is an edge function, ν a constant ($\nu \geq 0$). And Φ_0 represents the initial level set function.

If the level curves are propagating in a constant velocity, the level set $\{\phi = c \mid c \in \Re\}$ is a set which is away from the original curve with a distance c. In this occasion, the motion of level curves is subject to Huygens–Fresnel principle (Figure 6.5 (a)). Huygens-fresnel principle states that every point on the primary wavefront acts as a source of secondary spherical wavelets, with the same frequency and velocity as the primary wave.

In our application, we develop an algorithm based on the level set method. From above equations we can see that to evolve the curve we need to solve two problems: one is the evolution direction of the curve at a point, the other is the moving speed of points.

Considering the discrete representation of the zero level set, we move each interested point of the curve along its normal direction. The evolution of a curve is implemented through the speed field F. A positive F means that the point moves outwards and negative F indicate an inwards moving. The signed curvature of point p_i is obtained using equation 6-10.

The next position of a point of the interested point is determined by the relative distance along its normal direction.

The procedure of curve evolution is defined as following steps.

Step 1: Obtain the normal of each point using equation 6-7.

Step 2: Calculate the signed curvature k^* at each point using equation 6-10.

Step 3: the velocity field of the interested point is achieved using the following equation:

$$F = v \bullet \vec{n} \tag{6-16}$$

Where v is the speed of each interested point for movement.

To simplify the propagation of the discrete curve *C(t)* we define the speed as follows [Mulineux2007]:

$$v = \begin{cases} k \bullet w, \text{ if } p_i \text{ is an interested point;} \\ 0, \text{ if } p_i \text{ is not an interested point} \end{cases} \quad (6\text{-}17)$$

Where k is the curvature at point p_i and w is a coefficient ($w \in \Re$).

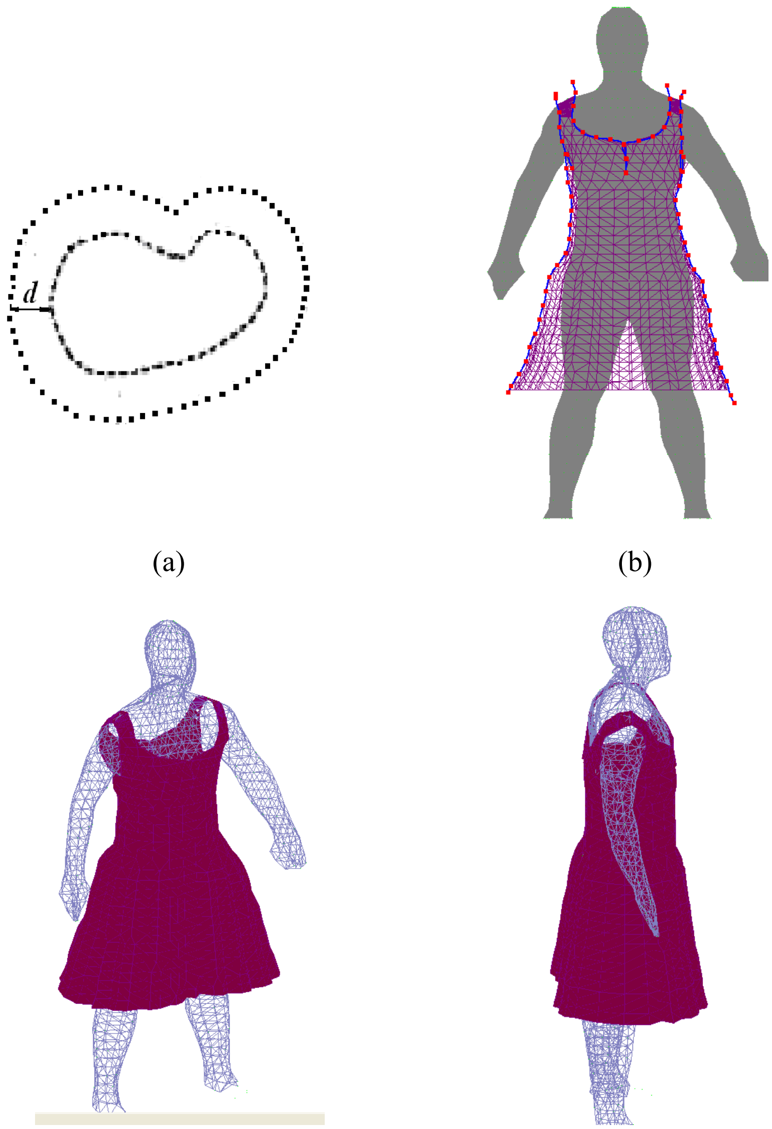
This method is used in the following two applications:

1. To generate garment curves below the waist line.

2. To optimise shapes of garment curves.

Garment curve generation: These curves of jackets below the waist line and skirts below the crotch are generated based on the level set method.

For example, during the construction of the lower part of skirts, curve evolving speed takes a constant value based on Huygens–Fresnel principle (Figure 6.5 (a)). The amplitude of the field at any point is the superposition of all these wavelets. The results of a skirt generation using evolving curves are shown in Figure 6.5 (b), (c) (d).



(a) (b)

(c) (d)

Figure 6.5 Garment curve generation using the level set method

Optimise Shapes of Garment Curves: The shape of a curve is optimised by moving the interested points. Since we use the distance from the profile curves to the body to refer positions of the vertices of a garment, the scraggy surface of the body will affect the shape of a garment, especially the area along the spine on the back of a human body. Thus, an optimisation procedure based on level set method is carried out to smooth the scraggy surface of garments' surface. For example, after curve optimisation, the back shape of a garment in the spine area looks more realistic. These points in the spine area of a garment are taken as interested points which are selected by calculating their signed curvature using equation 6-10. There are two steps to achieve optimisation.

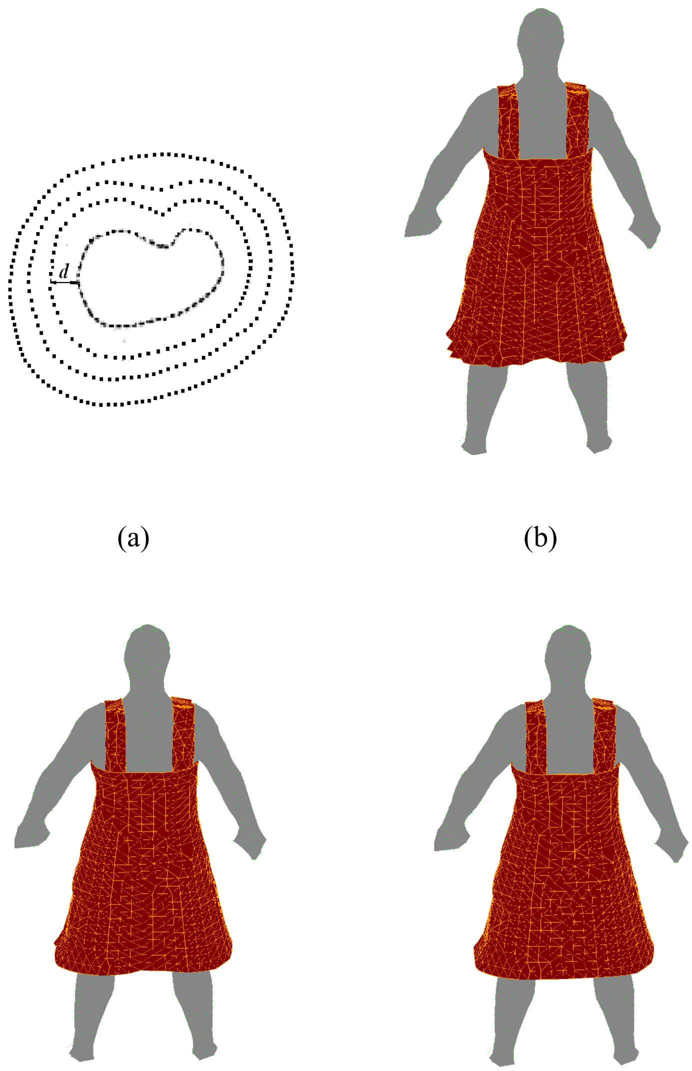
Step 1: Identify interested points on curves of a garment.

Step 2: Move these interested points of a curve based on the Level Set method.

To stop evolving curves, three criteria are used according to purposes of evolution, which are as follows: signed curvature threshold, curve energy threshold and body features.

Curvature and energy threshold values are determined through experiments. If the sign curvature at point p_i is greater than a threshold ($k^* \geq threshold\_curva$), the point p_i stops moving. If the curve energy exceeds the pre-defined value ($e \geq threshold\_eng$) the curve stops evolving. Figure 6.6 (a) shows a demonstration of curve evolving constrained by curve energy and curvature, which is used to mimic garment tension. Some examples of skirts with various tension in lower parts are demonstrated in Figure 6.6 (b), (c) and (d).

Feature constraints together with a curvature threshold are used to ensure that garments are hung on a body (Figure 6-2). Interested points of a curve stop moving when either feature constraints or curvature threshold are met.



(a) (b)

(c) (d)

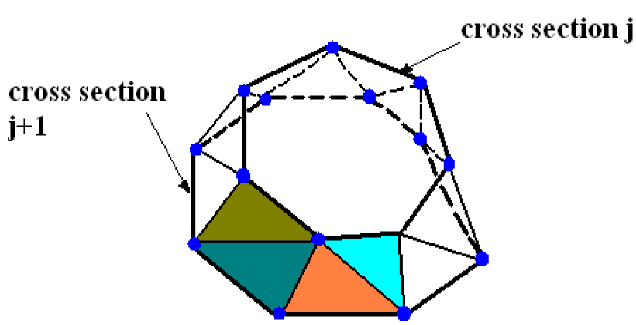
Figure 6.6 Garment tension mimic using the level set method

6.5 Mesh Generation

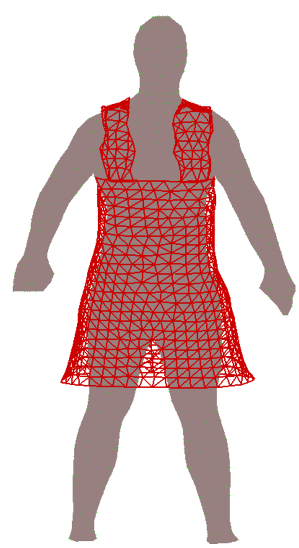
The 3D garment's vertices are generated corresponding to each part of a human body. Thus, these vertices have been already separated into several parts. Two steps are involved in the procedure of mesh construction algorithm. In the first step, correspondence triangulation algorithm is applied to each part. And then the surface of all those parts or patches of a garment is sewn together.

Step 1: Mesh generation for each part.

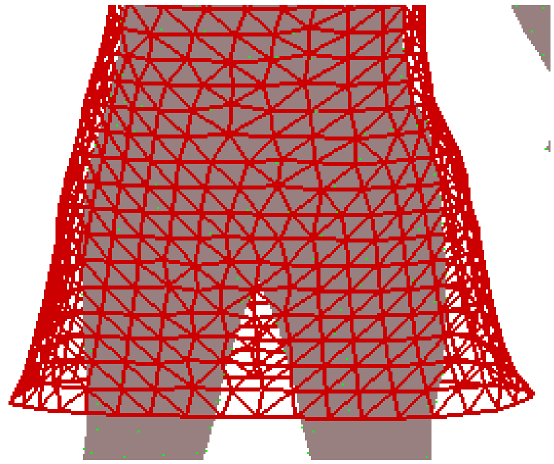
The shape of each part of a garment is relatively simple. Most of them are tubular shape or strap-like shape. Strap-like shape can be seen as a stack of open contours. To create mesh for the tubular and strap-like shape part, we use the correspondence triangulation method [Meyers1992].



(a)



(b)



(c)

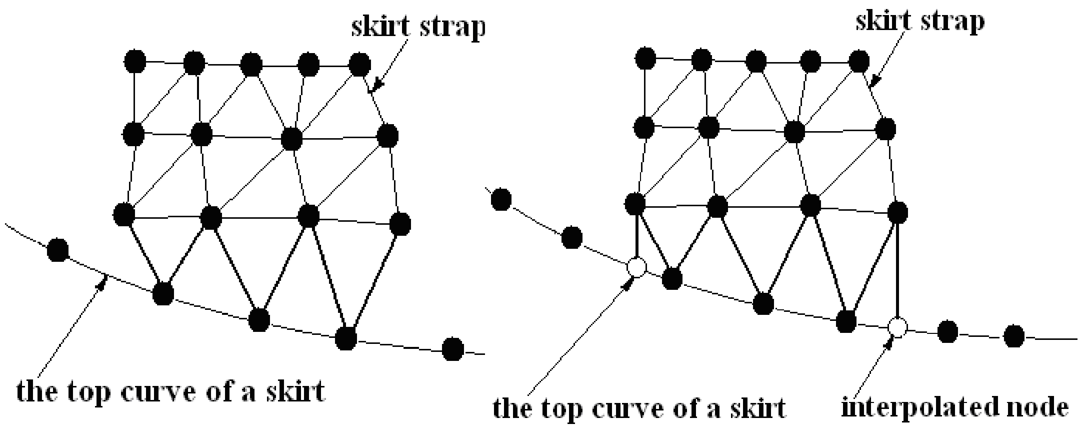
(a) Simply demonstrates the connection of two neighboring contours. (b) An example skirt of triangulation. (c) Zoom-in snapshot of a skirt mesh

Figure 6.7 Triangulation.

Step 2: Part surface connection.

After completion of triangulation of each part, an algorithm is applied to sew them together to create a complete garment surface. Two kinds of connection approach are used in terms of the shapes of the parts to be sewn: tubular-strap shape sewing and tubular shape sewing. For the first situation, there are usually more than two contours in the neighbouring planes. Normally, this happens to the connection among the chest, arms and the torso, which consisting of 4 contours. We use the arbitrary topology shape construction method [BCL96].

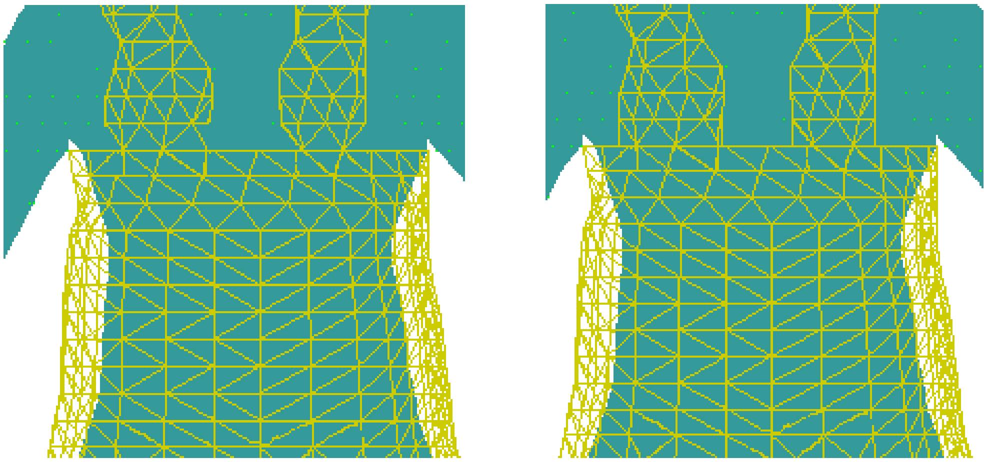
The second situation happens to strap skirts or cocktail skirt (Figure 6.8 (a) and (b)). After constructing the lower parts of a skirt and the strap or top upper parts of a skirt, we just need to sew the bottom edges of the strap or the bottom edge of the upper part of a cocktail skirt and the top edge of the torso surface. Two boundary points should be interpolated on the top edge of the torso surface. The point interpolation is straightforward. Project the boundary curve onto the torso surface along its normal direction. The intersection between the projected curve and the edge of the torso surface is the interpolated boundary node. Figure 6.8 demonstrates the sewing algorithm with and without the boundary node interpolation, where the white circles in (b) are interpolated boundary nodes. Then the adjacent points are connected by using the correspondence triangulation method [Meyers1992].



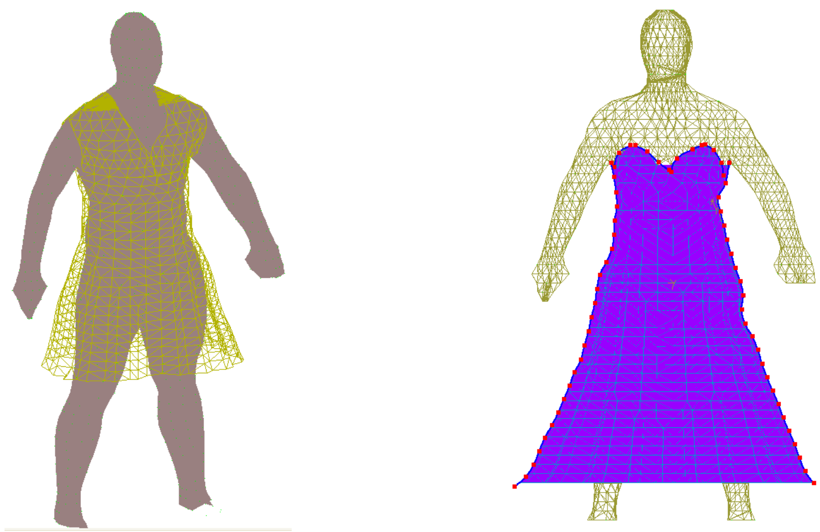
(a) Sewing without interpolating boundary nodes. (b) Sewing with interpolation new boundary nodes.

Figure 6.8 Part sewing

Generally, a complete skirt mesh surface is achieved by interpolating boundary nodes and sewing up mesh patches above and below armpits. Figure 6.9 (a) shows an example of sewing skirt the straps with the lower part without boundary node interpolation at top edge below the armpit. After interpolate nodes along the normal direction of the nearest triangular face on the lower part, we can obtain a complete garment mesh (Figure 6.9 (b)). More examples of node interpolation and sewing results with various patch shapes are shown in Figure 6.9 (c) and (d).



(a) Before sewing (b) After sewing



(c) A skirt after part sewing. (d) A evening skirt after part sewing

Figure 6.9 Examples of mesh sewing

6.6 Garment Mesh Refinement

After constructing a complete mesh of a garment using distance referring, interpolation etc, the visualisation of the constructed garment surface is usually not so pleasing. To improve visualization and get an elaborate garment mesh for further processing or simulation, we apply some mesh refinement schemes.

We smooth the coarse garment surface using an improved Laplacian algorithm [Vollmer1999]. And then the smoothed mesh is refined through applying a subdivision algorithm. We use a generalization of the subdivision scheme introduced by Loop [Loop1987], which is one of the simplest algorithms generating tangent plane smooth surfaces. Finally, a penetration detection approach is used to prevent the garment surface from colliding with the human body, which might be caused during both the construction and optimisation procedure.

6.6.1 Mesh Smoothing

Suppose the mesh of a garment *M*, which consists of a set of vertices $V = \{v_1, v_2, \ldots v_n\}$, $v_i \in \Re^3$, defining the shape of the garment mesh, is represented by a tuple (K, P). Where *K* is a simplicial complex describing the connectivity of the vertices, edges and faces. And $P: V \mapsto \Re^3$ is a function describing the topological type of the mesh. The set of vertices V is divided into two classifications: fixed vertices represented by V_{fix} and movable vertices denoted by V_{var}.

The key idea of the improved Laplacian algorithm is to push each vertex obtained by the Laplacian algorithm back by a distance d_i towards the previous points q_i and (or) the original points o_i, where d_i is the average of the difference b_i. This algorithm can generate the effect of smoothing and prevent the mesh from shrinking in some degree. This property ensures that the size and shape of a garment surface do not change dramatically after applying the smoothing scheme. The movable positions q_i ($i \in V_{\mathrm{var}}$) are replaced by p_i while the topology K of the mesh has kept unchanged. That is, the modified point p_i calculated using Laplacian algorithm are push back towards the previous points q_i and the original points o_i by the average of the differences. Point p_i is push back by a distance d_i (Figure 6.10). Values of b_i, q_i and d_i can be calculated by equation 6-18.

$$\begin{cases} b_i := p_i - (\partial o_i + (1-\partial) q_i \\ p_i = \begin{cases} p_i - \partial o_i + \frac{}{\left|Adi(i)\right|} \sum_{j \in Adj(i)} q_j, \\ q_i \qquad i \in V_{fix} \end{cases} \\ d_i := -(\beta b_i + \frac{1-\beta}{\left|Adj(i)\right|} \sum_{j \in Adj(i)} b_j \end{cases} \tag{6-18}$$

To prevent the boundary of a garment from shrinking during mesh smoothing, we set a rule that all boundary points of a garment mesh belong to fixed vertices.

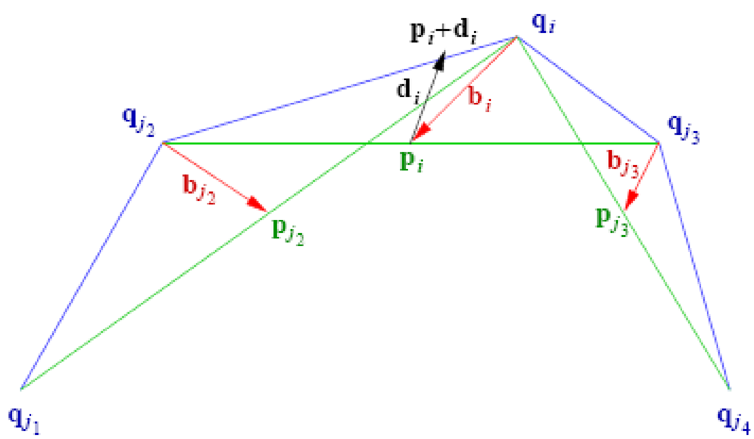
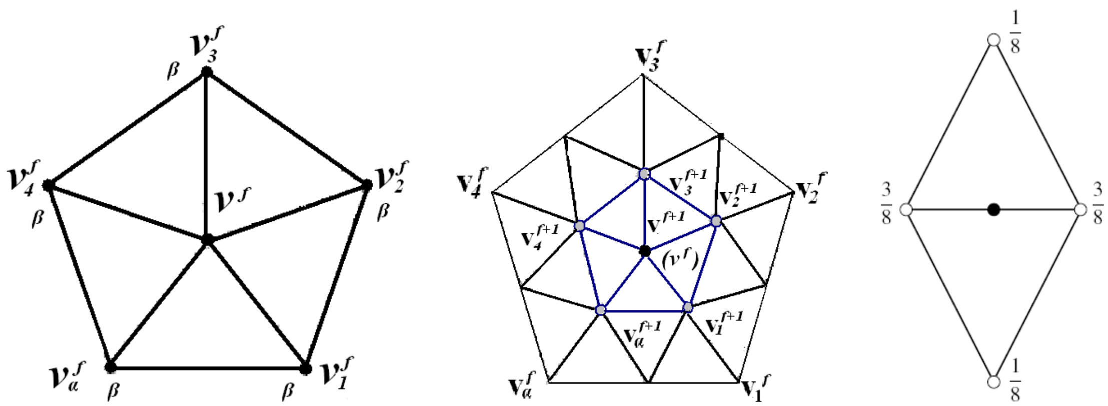


Figure 6.10 Point position modification

6.6.2 Mesh Subdivision

Loop's approximating subdivision scheme is a generalization of C^2 triangular B-splines, which creates four new triangles for each triangle in the mesh. The refinement step is carried out through splitting each triangular face into four sub-faces. The vertex positions of the reined mesh are then calculated using weighted averages of the vertices in the unrefined mesh. The location of a new midpoint of the edges and the new location of existing nodes can be updated using the edge mask and vertex mask (Figure 6.11).



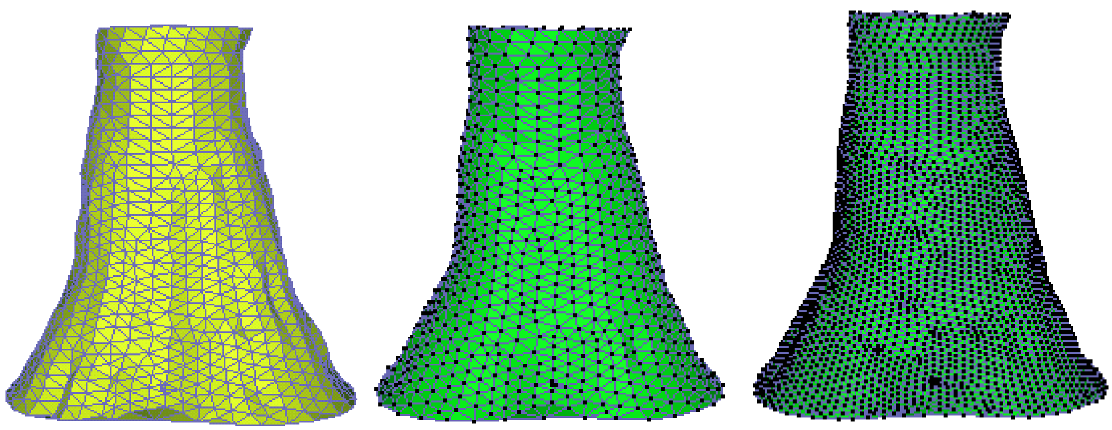
(a) The original mesh. (b) Vertex mask. (c) Edge mask.

Figure 6.11 Loop subdivision

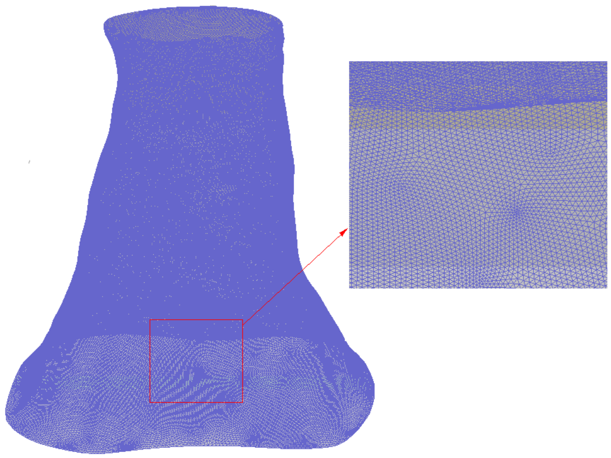
Let's denote a point set v_1^f, v_2^f,... v_α^f a neighbours of a vertex v^f of valance α. The new interpolated points can be obtained using the following equations [Loop1987]:

$$k^{*}= \begin{cases} \nu^{f+1} = (1-\alpha\beta)\nu^{f} + \beta \sum_{i}^{\partial} \nu_i^f \\ \nu_j^{f+1} = \frac{3}{8}\nu^{f} + \frac{3}{8}\nu_j^f + \frac{1}{8}\nu_{j+1}^f + \frac{1}{8}\nu_{j-1} \\ \beta = \frac{1}{\alpha}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{\alpha}\right)^2\right) \end{cases} \qquad (j=1,2,3\ldots,\alpha) \qquad (6\text{-}19)$$

The Loop scheme generates smooth surfaces everywhere, including at the boundaries of the mesh. To avoid boundary shrinkage we apply a boundary mask for boundary vertices of a garment mesh during subdivision [Zorin1996]. An example of garment mesh subdivision is shown in Figure 6.12.



(a) (b) (c)



(d)

(e)

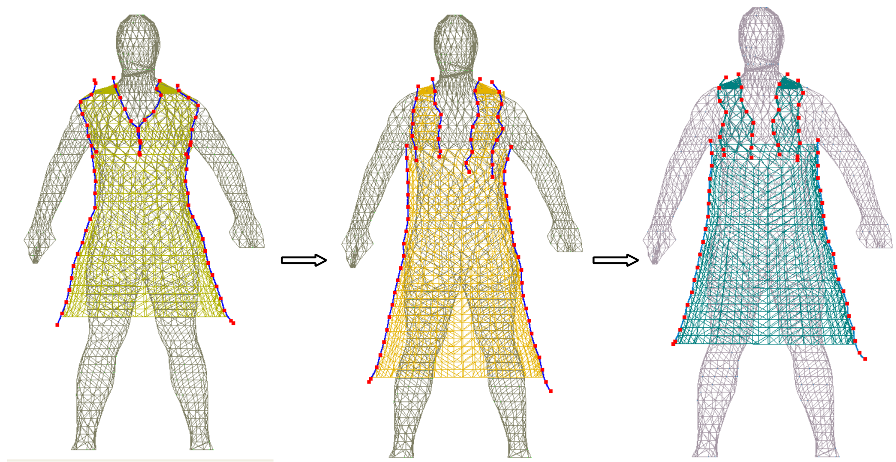
(a) The original skirt mesh. (b) Smooth using the improved Laplacian method. (c) Loop subdivision 1 iteration. (d) Loop subdivision 3 iteration. (e) The rendering result

Figure 6.12 An example of subdivision

6.7 Shape Modification

Due to the characteristics of spline curves, the profile shape of garments can be modified easily. Users can modify garments' profile shape through dragging the control points of spline curves to the desirable position. Users are allowed to modify a garment's shape both in 2D and 3D space. Through dragging and moving control points of the existing curves, users are not disturbed by changing the whole shape accidently. This gives users greater flexibility to modify garment's shape to obtain various shapes for comparison. The 3D garment surface is regenerated after modification (Figure 6.13).

We find that it is beneficial to allow users to modify the existing shape. A user wants to be sure that modifying some local parts is not going to affect some other part of the shape that is already deemed desirable. Our method allows users to modify the shape locally by dragging the control points of spline curves and even a user can change the whole shape through dragging more control points.



(a) (b) (c)

Figure 6.13 Examples of garment shape modification

6.8 Penetration Detection

Penetration between the 3D human model and garment surfaces might happen during the procedure of garment mesh construction, shape optimisation and mesh processing, For example, penetration occurs between a skirt mesh refined from the original skirt mesh in Figure 6.14 (a) and the human model (Figure 6.14 (b)).

We develop a layer-distance-based algorithm to perform penetration detection and response. Both a human model and garments are represented by surfaces, which are all marked as different layers of our systems. We check penetration by calculation distance between two layers. The inner layer is usually taken as the reference layer during the procedure. As a dressed human body, the body is always the inner most layer. In this case, we compute signed distances denoted by d_c from the vertices of a human body to the garment surface represented by S_g along the normal direction of the body vertices. Here is the equation for penetration judgment as following:

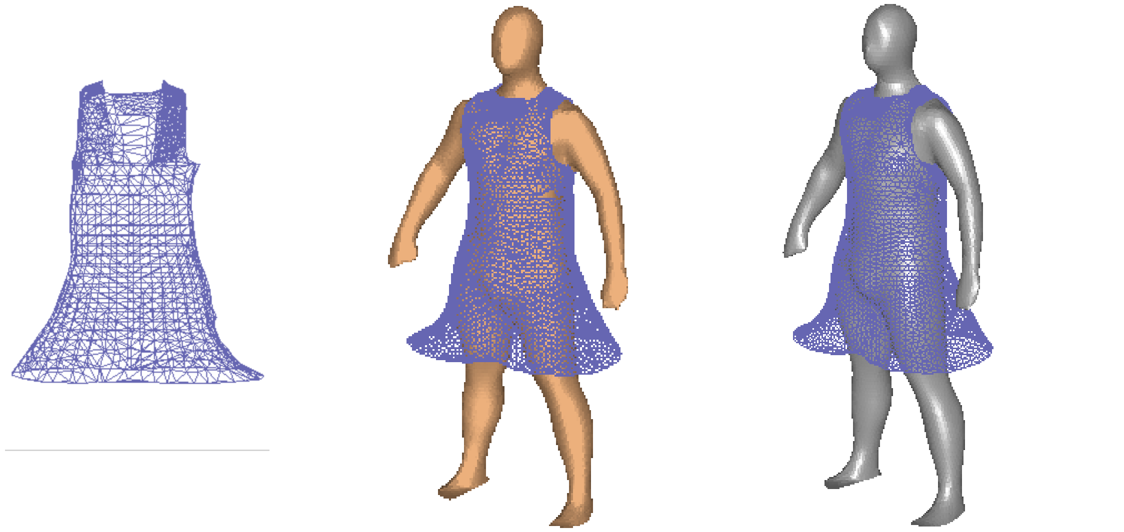
$$k^* = \begin{cases} d_c \geq 0, \text{ no penetration;} \\ d_c < 0, \text{ penetration} \end{cases} \qquad (6\text{-}20)$$

If penetration occurs, these related vertices of the garment are moved outwards by Δd in the direction of the face normal. The value of Δd is determined by following equation:

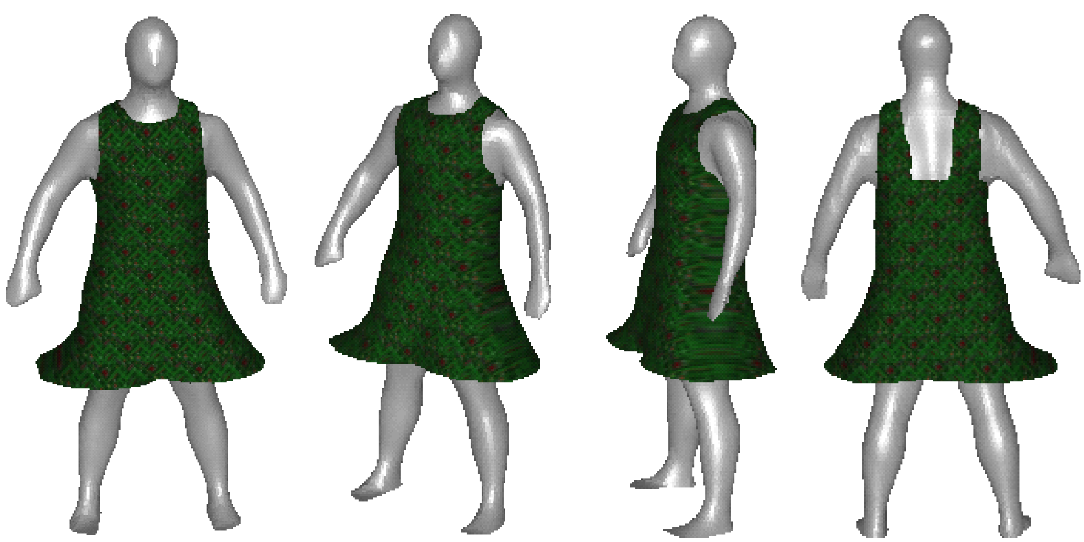
$$\Delta d = |d_c| + \sigma \qquad (6\text{-}21)$$

Where the variable σ is a small tolerance.

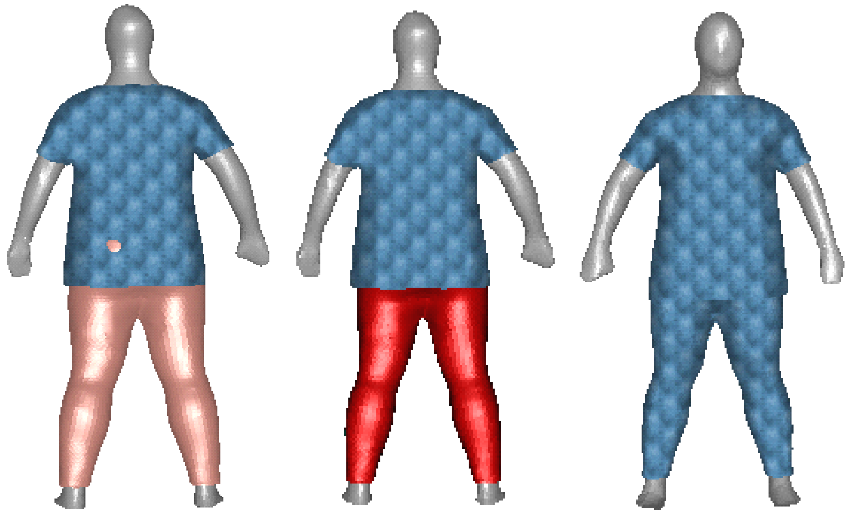
Results of an example of penetration response are shown in Figure 6.14 (c)-(g). This algorithm can also be used to detect penetration between garment surfaces with one garment surface is designated as the inner layer (Figure 6.14 (h)-(i)).



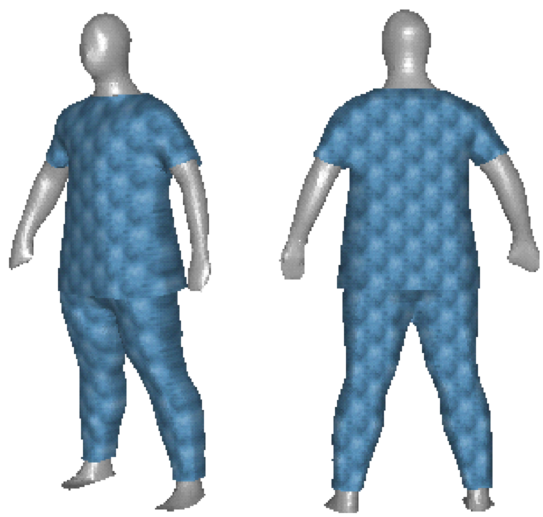
(a) (b) (c)



(d) (e) (f) (g)



(h) (i) (j)



(k) (l)

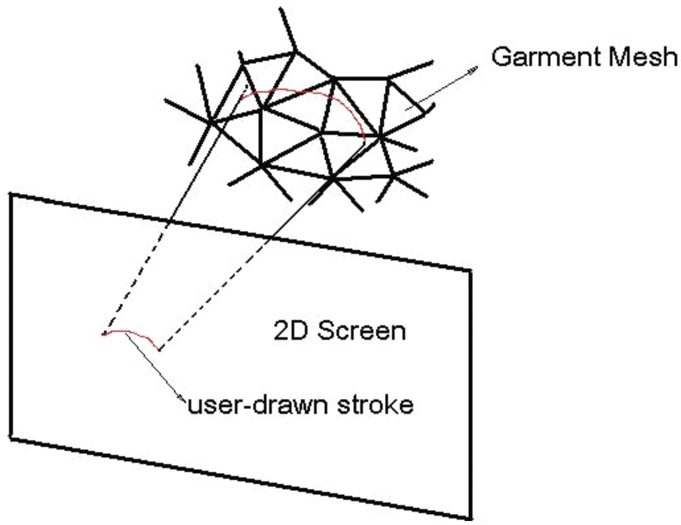
Figure 6.14 Examples of penetration detection and response

6.9 Decoration Tools

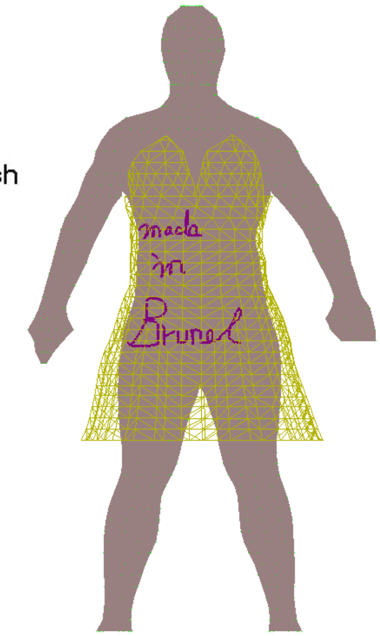
This system provides three 3D decoration tools to enable users to add some details to garments. As in reality, almost all garments have some details such as brand name, brand label and some colourful strap etc. In current system, three decoration tools are provided: 3D pen, 3D brush and 3D embroidery tool.

6.9.1 Three Dimensional Pen

The 3D pen draws curves on a garment surface. When a user draw a 2D stroke on the screen, it project stroke points onto the garment surface (Figure 6-15 (a)). The drawing curve is created by connecting all points of each projected points of a stroke on the garment surface. The points of a 2D stroke are projected onto the garment surface along the view direction using ray tracing techniques. Generally, there are at least two intersection points between a ray and the garment mesh surface. The intersection point which is closest to the viewer is kept as a drawing point. Examples of pen drawing on garments are shown in Figure 6.15 (b).



(a) Demonstration of pen drawing.



(b) Examples of pen drawing

Figure 6.15 Illustration of pen drawing

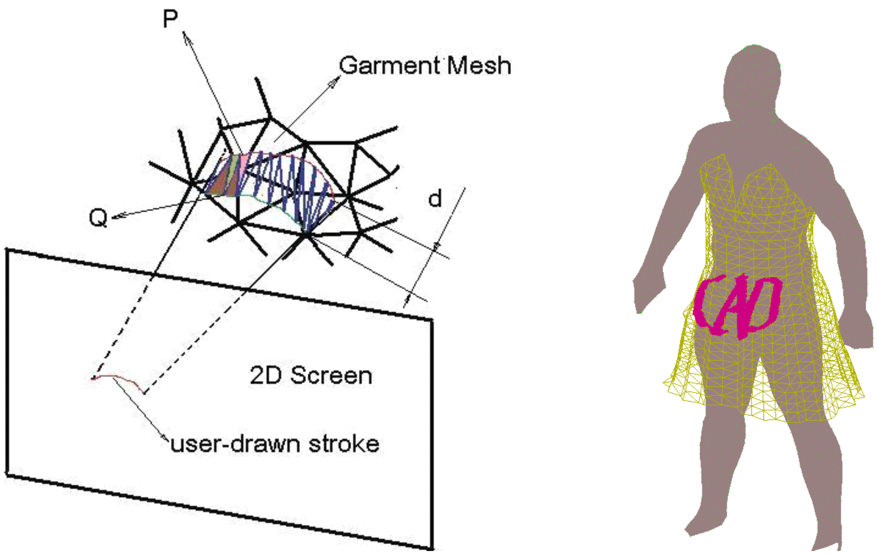
6.9.2 Three Dimensional Brush

The sketches are projected onto the garment's surface along the view direction, which constitutes the base curve denoted by $P = \{p_1, p_2, ... p_m\}$. The curve P is raised along the normal direction of each triangular face on which each point is located. The raised boundary curve point q_i of $Q = \{q_1, q_2, ... q_m\}$ can be represented as follows:

$$q_i = p_i + d_r * N \qquad (6\text{-}22)$$

Where N is the normal of a triangular face and d_r is the distance that each point p_i is raised.

A mesh surface of the decoration, which is constructed by using contour correspondence method with two curves P and Q as boundary curves is generated using the 3D brush (Figure 6.16 (a)). An example of decoration created by 3D brush tool is shown in Figure 6.16 (b).

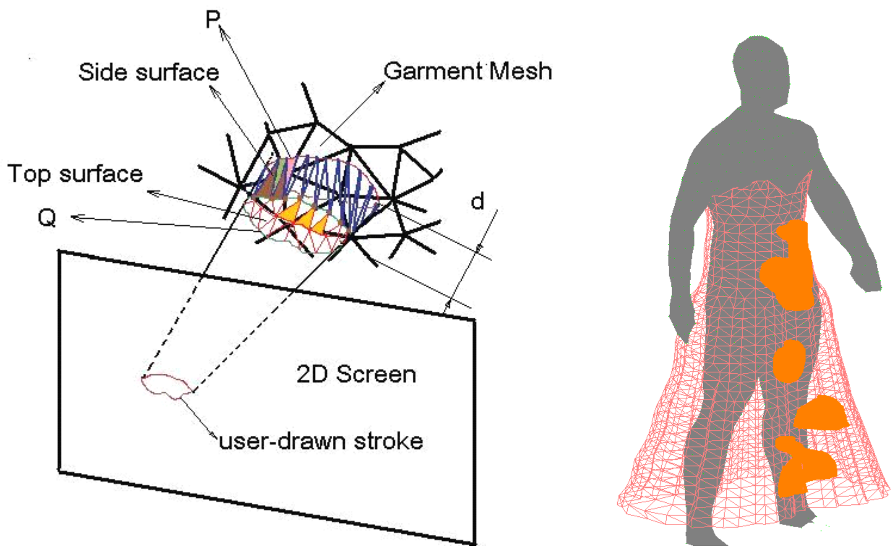


(a) Demonstration of side surface creation. (b) Examples of side surface

Figure 6.16 Illustration of three dimensional bursh tool

The embroidery decoration contains of two surfaces: the top surface and the side surface . The side surface is constructed by the two boundary curves P and Q (Figure 6.17 (a)). The top surface is based on the raised boundary curve Q.

The raised boundary curve Q is projected onto a proper plane in its major direction using PCA method [Gorban2007]. In this case, the 3D boundary curve Q is converted into 2D domain denoted by Q_p. Through applying 2D Delaunay triangulation to the projected curve Q_p, we obtain a rough triangular mesh. Then constrained Delaunay triangulation and Ruppert's refinement algorithm are used to generate quality surface [Shewchuk1996, Shewchuk2002, Ruppert1995].

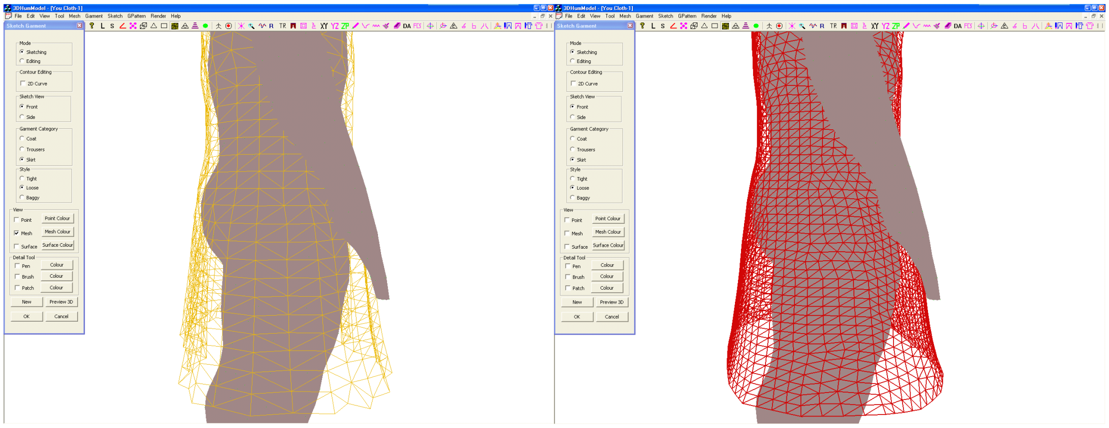


(a) Illustration of top surface generation (b) Examples of top surface

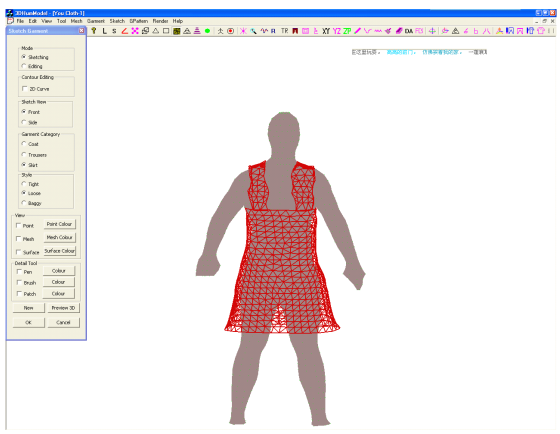
Figure 6.17 Illustration of embroidery tool

6.10 Results

Figure 6.18 illustrates examples of a strap skirt. Figure 6.18 (a) shows an original skirt generated from sketches and (b) shows the skirt after mesh processing and penetration detection. Figure 6.18 (c) shows the front view of the skirt.



(a) (b)

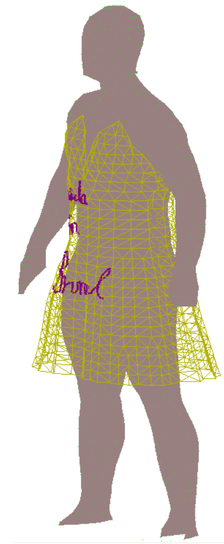


(c)

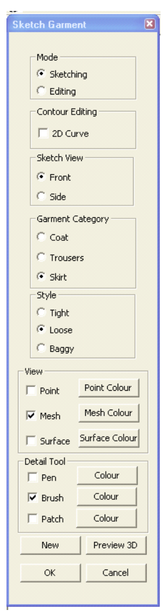
Figure 6.18 An example of a strap skirt

Figure 6.19 shows some results of using detail tools. After selecting "Pen" from "Detail Tool" panel (Figure 6.19 (a)), user can draw some symbols on garment surfaces (Figure 6.19 (b)). Three dimensional paintings are obtained by selecting "Brush" and painting directly on garment surfaces, such as A toy pinwheel pattern, a brand name “Nike” and its symbol (Figure 6.19 (c) and (d)). Some arbitrary sketching patterns are embedded on a skirt surface using the embroidery tool after selecting "Patch" on the panel (Figure 6.19 (e) and (f)). An example of combination use of these detail tools can be found in Figure 6.19 (g) and (h).

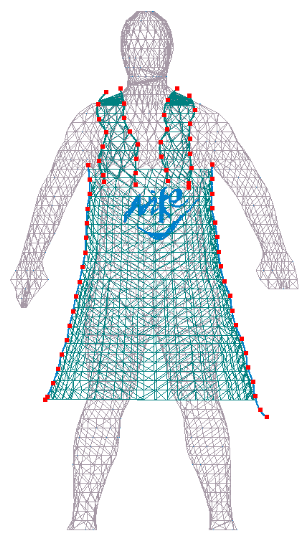
(a)



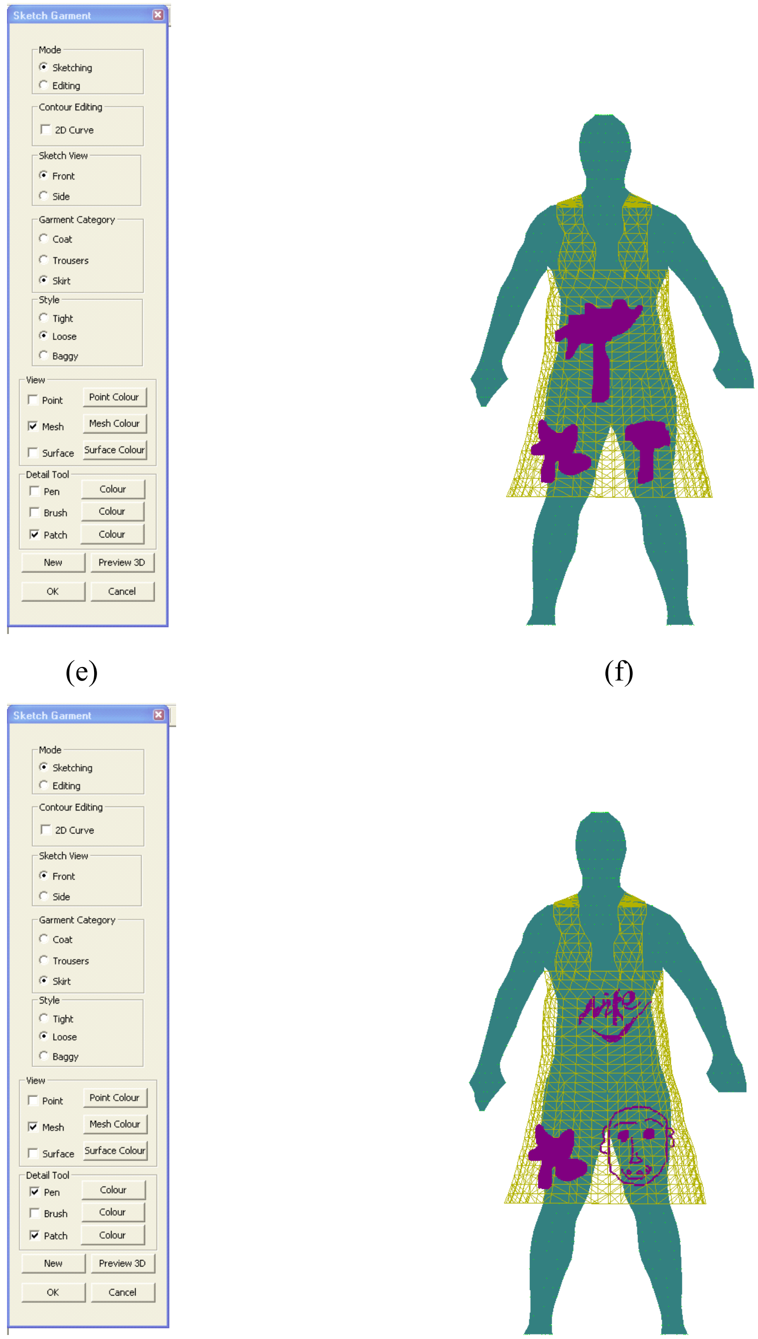
(b)



(c)



(d)



(e) (f)

(g) (h)

Figure 6.19 Demonstration of using decoration tools

6.11 Conclusion

The construction of the curves of 3D garments is based on distances from strokes to the human body, which are constrained by key body features. The general shapes of garments are optimised using level set method. Mesh processing methods are applied to refine garment mesh surface. Finally, three decoration tools are provided to generate some 3D decorations on garment surfaces. Our system generates refined garment mesh surface, which make it possible to deform or change garments arbitrarily by applying mesh editing algorithms. And garments generated by our system could be flattened into 2D patterns for garment design in the future by adopting surface flattening techniques [Wang2002]. Since in our method, garments are generated using user-drawn strokes, there are still some artificial effects on garments. Furthermore, this approach does not take garment material into account. Garments generated by our method do not have any material properties. Therefore, some algorithms can be applied to simulate different garments materials such as cotton, silk or nylon in the future.

Chapter 7. Conclusion and Feature Work

7.1 Summary of the Thesis

Dressing virtual humans is gaining more attention as demands for the applications of realistic virtual humans increase. However, the complexity for virtual dressing using current software packages can only be overcome by skilled or professional users. Therefore, it is necessary to develop new techniques to reduce the complexity for virtual dressing and to enhance ease of use of the interfaces.

Research presented in this thesis investigated an intuitive interface for dressing virtual humans from body scan data. Generally, there are two stages in the modelling process: semantic human model construction and 3D garment generation.

The thesis started with an introduction to building semantic virtual human models from real human body data. We used a $[TC]^2$ body scanning system is used to obtain human body data from real humans. The body scan data was pre-processed using a commercial software package Geomegic software. During the procedure of pre-processing, noises of the raw human scan data were removed and holes on human bodies were filled in with interpolated data points. After removing noises and filling all holes on the human body data, it was sliced horizontally into cross sectional data using Geomegic software [Geomegic]. Four major steps are employed to generate a human model with semantic information. It started with segmentation of a human model into six parts. The second step was to sample points on the point human body to obtain a human model with different levels of detail. The third step is to triangulate the point cloud human model. The cross sectional human body data was then reconstructed into human model represented by triangular mesh. Finally, a simple but efficient algorithm was developed to automatically extract key body features from the reconstructed human model surface. With completion of semantic information extraction, a semantic human model constructed. A skeleton for generating simple garment is constructed based on the structure of the cross sectional human model. Key features are defined based on the application of garment generation.

The second stage following the completion of semantic human model construction was to sketch and construct 3D garments. This procedure began with drawing garment profile sketches by a user around the semantic human model on 2D screen. To improve the quality of user-drawn strokes and give more flexibility to users, sketches of the garment profile were processed and converted into spline curves. Users were able to change a garment shape by dragging control points of these spline curves. Each segment of these spline curves was associated with a corresponding part of the semantic human body. The creation of vertices of a garment was based on the distance from spline curves to the human body. These 3D garment vertices were classified into two categories according to their interpolation modes: silhouette vertices and internal vertices. Both of the two kinds of vertices were interpolated based on distance field of the human body. Positions of these garment vertices were then optimized using a Level Set method. A complete garment surface was obtained by sewing up neighbouring cross sections of the garment. Some more vertices were interpolated when they are necessary.

In this thesis, some more algorithms were applied or developed to further beautify and optimize the constructed 3D garments. In the first hand, to improve visualization of a garment surface and help further surface editing, some mesh processing methods such as mesh smoothing and mesh subdivision algorithms were applied. These mesh processing methods enhanced the mesh quality of garment surfaces. Furthermore, a more detailed garment mesh surface was achieved by applying mesh subdivision.

Three detail tools, which were 3D pen, 3D brush and 3D embroidery tool, were provided in this thesis allowing users to add some 3D garment details such as brand name and symbol.

7.2 Contributions

This thesis has primarily focused on dressing virtual humans based on a sketching interface. A sketching interface provides an intuitive way for users to convey 3D shapes. Body scan data captured from real human body is a good resource for generating virtual human models.

In this thesis, we have made several contributions to the knowledge of dressing virtual humans from scanned data.

The main contributions of this thesis are summarized as follows:

- A prototype framework for dressing humans from body scans based on sketches

  This framework presented enable users to dress human models reconstructed from body scan data. It allows users to specify garment parameters by drawing on the body or directly sketch garment contour around the body. The output is a full geometric garment mesh based on sketches.

- Use body features as constraints for generation of 3D garment vertices.

  In this thesis, generation of 3D garment vertices is based on the distance from the garment contour curves to the human model constrained by body features. Garment vertices are classified into two categories: external vertices and internal vertices. External vertices are created directly from the garment contour curves. Internal vertices are interpolated using a non-linear equation based on the distance from the garment contour curves to the human model.

- A method for locally modifying garment shape.

One of the easiest ways for manipulating a garment shape is to modify the exiting one. In this thesis, we provide 3D garment shape modification based on the front-view garment contour. Users can modify the existing garment shape by dragging the control points of contour curves of the existing garment to the desired positions.

- A method for 3D garment shape modification.

We develop a method for modifying positions of 3D garment vertices based on a Level Set method. This modification procedure helps to prevent the garment shape from exactly following the body shape. Cloth tension is mimicked during the procedure of modification controlled by curve energy and curvatures. In addition, we have provided three 3D decoration tools allowing users to add some little 3D decorations to the garment surface. These 3D decorations represented by 3D curves or triangular mesh surfaces enhance realism of a garment compared with texture mapping detail.

- Sketching interface

A sketching interface has been developed using Microsoft Visual Studio C++ and OpenGL library. The interface allows users to draw directly on 2D screen. Users can sketch 2D garment contours for 3D garment generation and simple strokes to specify garment size for garment patterns.

7.3 Limitations and Future Work

There are several major directions that the work could be extended and improved in the future.

- More complicated garment shape

Garment shapes are constructed from cross sections. Some algorithms can be applied for constructing more completed garment shapes or for garment deformation [Hyun].

- Material properties

Perhaps the primary limitation of the proposed method in this thesis is that these garments demonstrated do not show any material properties. Therefore, the proposed method could be improved by applying material algorithms can be applied.

- More flexible Modification

In this thesis, users are allowed to modify a 3D garment shape through modifying the garment contour from the front view. This, to some degree, improves users' flexibility to modify an existing garment. However, to obtain a better garment shape, the algorithm can be improved to intuitively modify garment local or overall shapes by directly editing the garment surface.

- Multi-view drawing and editing

The approach presented in this thesis allows users to draw and modify garment contour from the front view. The future work can be carried out to allow users to draw and modify garment from multiple views. The 3D garment surface could be deformed according to a user's over-sketches from any view.

- Editing and Decoration Tools

This thesis explored three decoration tools to improve garment surface decoration. We believe these tools can be improved to allow arbitrary embroidery, cutting and to add some cloth fringes.

References

[Abdel-Malek2000] Abdel-Malek, K., Bleckmore, D., and Joy, K.: Swept volumes: Foundations, perspectives and applications. *International Journal of Shape Modeling*. 12(1), 87-127. 2000.

[Allen2002] Allen B., Curless B. and Popovic Z..: Articulated Body Deformation From Range Scan Data, *Proc. ACM SIGGRAPH*, 612–619.2002.

[Allen2003] Allen B., Curless, B., and Popovic Z..: The space of all body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (ACM SIGGRAPH 2003)*, 22, 3, 587-594.2003.

[Animazoo] http://www.animazoo.com/capturedSystems.aspx

[Badler1979] Badler, N., O'Rourke, J. and Toltzis, H.: A Spherical Representation of A Human Body For Visualizing Movement, *Proc. IEEE*, 67, 1397–1403. 1979

[Barthe2003] Barthe L., Dodgson N. A., Sabin M. A., Wyvill B., Gaildrat V.: Two-Dimensional Potential Fields For Advanced Implicit Modeling Operators. *Computer Graphics Forum 22*, 1, 23–33.2003

[BCL1996] Bajaj C.L.,Coyle E.J., Lin K.N.: Arbitrary Topology Shape Reconstruction from Planar Cross Sections, *Graphical Models and Image Processing*, Volume 58, Number 6, November, 524-543(20.)1996

[Blackmore1997] Blackmore, D.,Leu, M., and Wang, L.: The Sweepenvelopedifferential Equation Algorithm And Its Application To Nc Machining Verification. *Computer-Aided*

Design 29, 9, 629–637.1997.

[Bourguignon2001] Bourguignon D., Cani M.P., and Drettakis G.: Drawing for Illustration and Annotation in 3D, *Computer Graphics Forum*, 20, no. 3, 114-122.2001.

[Chadwick1989] Chadwick. E. John, Haumann D. R. and Parent R. E.: Construction for Deformable Animated Characters, *Proc. ACM SIGGRAPH,* 243-252. 1989.

[Cherlin2005] Cherlin Joseph J, Samavati F., Sousa Mario C, Joaquim A.J.: Sketch-Based Modeling With Few Strokes, A Jorge, *In SCCG '05: Proceedings Of The 21st Spring Conference On Computer Graphics*, 137-145.2005.

[Company2004] Company P, Contero M, Conesa J, Piquer A.: An Optimisation-Based Reconstruction Engine For 3D Modeling By Sketching. *Computer & Graphics,* 28: 955–79.2004

[Cordier2002] Cordier, F., Magnenat-Thalmann, N.: Real-time Animation of Dressed Virtual Humans. *Eurographics* 2002.

[Cordier2003] Cordier, F., Seo, H., Magnenat-Thalmann, N.: Made-To-Measure Technologies For An Online Clothing Store, *IEEE Computer Graphics and Applications*, 23 1, 38-48. 2003

[Cyberware] Cyberware. http://www.cyberware.com

[Daanen1998] Daanen H., and G. Jeroen van de Water: *Whole Body Scanners. Displays*, Volume 19, Issue 3, 30 November, 111-120.1998.

[Da1972] Da Vinci L., Paul Richter J.: *The Notebooks of Leonardo da Vinci*, 1 182-183, Dover, 1972.

[Demirel2006] Demirel, H.O., Bachle, C., Rodriguez, A.M., Siegler, J.: Cummins MidRange Engine Plant -Defining New Training Standards for New Product Launches, *IE431 Senior Design Project Purdue University* .2006.

[Eggli1997] Eggli L., Hsu C., Brüderlin B.D., Elber G.: Inferring 3D Models from Freehand Sketches and Constraints. *Computer-Aided Design*, 29(2), 101-112, 1997.

[Ferguson1992] Ferguson, E. S.: Engineering And The Minds Eye. *MIT Press*.1992

[FRC05] Fontanaa M., Rizzib C. and Cuginic U.: 3D Virtual Apparel Design For Industrial Applications, *Computer-Aided Design*, Volume 37, Issue 6, Pages 609-622. 2005

[Fuhrmann03] Fuhrmann A., Groß C., Luckas V., and Weber A.: Interaction-free Dressing of Virtual Humans. *Computers & Graphics*, 27(1):71–82, January 2003.

[Geomagic] Geomagic Studio. http://www.geomagic.com/en/products/studio/

[Gorban2007] Gorban A., Kegl B., Wunsch D., Zinovyev A. (Eds.): Principal Manifolds for Data Visualisation and Dimension Reduction, *LNCSE 58, Springer*, Berlin – Heidelberg – New York, . ISBN 978-3-540-73749-0. 2007.

[Göbel 2004] Göbel S., Schneider O., Iurgel I., Feix A., Knöpfle C., Rettig A.: Virtual Human: Storytelling and Computer Graphics for a Virtual Human Platform, Technologies for Interactive Digital Storytelling and Entertainment, *Second International Conference, TIDSE 2004*, Darmstadt, Germany, June 24-26, 2004

[Hamamatsu] Hamamatsu. http://usa.hamamatsu.com/sys-industrial/blscanner

[Hinds1990] Hinds B.K., McCartney J., Interactive Garment Design: *The Visual Computer*, 6, 53-61.1990

[HM90] Hinds B.K. and McCartney J.: Interactive Garment Design, *The Visual Computer*, 6, no. 2, 53-61. 1990.

[Hoppe1993] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W.: Mesh Optimization. *Proceeding SIGGRAPH* , New York, USA: ACM. p. 19–26.1993

[Hyun] Hyun D. E, Yoon S. H., Chang J. W., Seong J. K., Kim M. S. and Jüttler B.: Sweep-based human deformation. *The Visual Computer*.542-550, 2005.

[Igarashi1999] Igarashi T.., Satoshi. M., Hidehiko. T.: Teddy: A Sketching Interface for 3D Freeform Design," *Proc. 26th Conf. Computer Graphics and Interactive Techniques*, ACM Press, 409-416.1999.

[Igarashi2002] Igarashi T., Hughes J. F.: Clothing Manipulation, *15th Annual Symposium on User Interface Software and Technology, ACM UIST'02*, Paris, France, October 27-30, 91-100.2002

[Kara2006] Kara LB, D'Eramo CM, Shimada K.: Pen-Based Styling Design For 3D Geometry Using Concept Sketches And Template Models. *In: Proceedings of the 2006 ACM symposium on solid and physical modeling*. p. 149–60.2006

[Kep1975] Keppel E.: Approximating Complex Surfaces By Triangulation Of Contour Lines. *IBM J Res Develop*, January, 2–10.1975.

[Kim2005] Oh, S. Kim, H. Thalmann N. M.,. Wohn K.: Generating Unified Model For Dressed

Virtual Humans Author. *Visual Computer*, VOL 21; Number , 8-10, pages 522-53. 2005.

[Kim2006] Kim DH, Kim MJ.: A New Modeling Interface For The Pen-Input Displays. *Computer-Aided Design.* 38:210–23.2006.

[Kolja2002] Kolja K¨ahler J¨org Haber Hitoshi Yamauchi Hans-Peter Seidel. Seidel: Head Shop: Generating Animated Head Models with Anatomical Structure, *Proc. ACM SIGGRAPH Symposium on Computer Animation*, 55–64, 2002.

[Ku2008] Ku D., Qin SF*, Wright DK, Ma C.: Online Personalised Non-Photorealistic Rendering Technique For 3D Geometry From Incremental Sketching, *Journal of Computer* Graphics Forum, 27. 1861-1868, 2008.

[Lämkull2007] Lämkull D., Hansonc L. and Örtengren R.: The Influence Of Virtual Human Model Appearance On Visual Ergonomics Posture Evaluation. *Applied Ergonomics*,Volume 38, Issue 6, Pages 713-722,2007.

[Li1997] Li P, Jones P. R. M.: Automatic Editing and Curve fitting of 3-D Surface Scan Data of the Human Body. *Proc. International Conference on Recent Advances in 3-D Digital Imaging and Modelling*, 296-301, Ottawa, Canada.1997

[Lipson1996] Lipson H., Shpitalni M.: Optimization-Based Reconstruction of a 3D Object from a Single Freehand Line Drawing. *Computer Aided Design*, 28 (8), 651-663, 1996.

[Loop1987] Loop C.: Smooth Subdivision Surfaces Based On Triangles. *Master's thesis, Department of Mathematics, University of Utah*, August 1987.

[Maya2004] Maya cloth: In The User Manual For Designing Clothes With Maya, *http://caad.arch.ethz.ch/info/maya/manual/MayaCloth*. 2004.

[Meyers1992] Meyers D., Skinner S. and Sloan K.: Surface From Contours. *ACM Trans Graph* 11 3, 228–258.1992

[Moccozet2004] Moccozet L., Dellas F., Thalmann N. M., S. Biasotti, M. Mortara, B. Falcidieno, P. Min and R. Veltkamp: Animatable Human Body Model Reconstruction from 3D Scan Data using Templates, *In CapTech Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, Zermatt, Switzerland. December 2004.

[Mulineux2007] Glen Mulineux, Sebastian T. Robinson, Fairing point sets using curvature. *Computer-Aided Design.* Volume 39 , Issue 1 Pages: 27-34, 2007

[Naya2002] Naya, F., Jorge, J. A., Conesa, J., Contero, M., Gomis, J. M.: Direct Modeling: From Sketches To 3d Models. *Proc. of the 1st Ibero-American Symposium in Computer Graphics*, 109–117.2002.

[Nedel1998] Nedel, L.P.Thalmann, D.: Modeling And Deformation Of The Human Body Using Ananatomically-Based Approach, *Proceedings of Computer Animation* 98. 8-10 page(s): 34-40, Philadelphia, PA, USA.1998,

[Nurre2000] Nurre J.H., Connor J., Lewark E. and Collier J.S.: On Segmenting The Three-Dimensional Scan Data Of A Human Body, *IEEE Transaction on Medical Imaging.* 19 (8), 787–797. 2000

[Oh2003] Oh BS, Kim CH.: Progressive 3D Reconstruction From A Sketch Drawing. *Computer & Graphics*; 27:581–92.2003

[Olsen2009] Olsen L., Faramarz F. Samavati, Sousa M. C., Jorge J. A.: Sketch-Based Modeling: A Survey. *Computers & Graphics*, Volume 33, Issue 1, Pages 85-103, 2009

[OpenGL] OpenGL, http://www.opengl.org/

[Osher1988] Osher S. and Sethian J. A.: Fronts Propagating With Curvature-Dependent Speed: Algorithms Based On Hamilton–Jacobi Formulation, *J.Comput. Phys.*, 79, 12–49, 1988.

[Parsons2008] Parsons T. D., Kenny P. and Rizzo A. A.: Virtual Human Patients For Training Of Clinical Interview And Communication Skills. *Proceeding of 7th ICDVRAT with ArtAbilitation*, Maia, Portugal, 2008.

[Pereira2000] Pereira J., Jorge J., Branco V., Nunes F.: Towards Calligraphic Interfaces: Sketching 3D Scenes With Gestures And Context Icons. WSCG'2000. *Conference Proceedings*, Skala V. Ed., 2000.

[Pelechano2008] Pelechano N., Stocker C., Allbeck J. and Badler N.: Being A Part Of The Crowd: Toward Validating VR Crowds Using Presence. *Proceedings of Autonomous Agents and Multiagent Systems,* pp 136-142. 2008.

[Qin2000] Qin S.F, Wright D.K, Jordanov I.N.: From On-Line Sketching To 2D And 3D Geometry: A System Based On Fuzzy Knowledge. *Computer-Aided Design* ; 32:851–66.2000

[Ratner2003] Ratner P. 3-D Human Modeling and Animation. 2nd Edition, *New York: Wiley*; 2003.

[Realcolorwheel]http://realcolorwheel.com/human.htm

[Rockstar] Rockstar North Co. Ltd. http://www.rockstarnorth.com/

[Requicha1980] Requicha, A. A.G.: Representations For Rigid Solids: Theory, Methods And Systems. *Computing Surveys* 12, 4, 437–464.1980

[Rose2007] Rose K., Sheffer A., Wither J., Cani M. P., Thibert B.: Developable Surfaces from Arbitrary Sketched Boundaries. *Proc. Fifth Eurographics Symp. Geometry Processing (SGP*

'07), 163-172, 2007.

[Ryokai2003] Ryokai K., Vaucelle C., Cassell J.: Virtual Peers As Partners In Storytelling And Literacy Learning. *Journal of Computer Assisted Learning* 19, 195-208, 2003.

[Ruppert1995] Ruppert J..:A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation. *Journal of Algorithms* 18(3):548-585, May 1995.

[Salomon99] Salomon D.: Computer Graphics And Geometric Modeling, *Springer*, ISBN 0387986820, 9780387986821, 1999

[Savant2001] Karunaratne S., Yan H.: 3D Animated Movie Actor Training Using Fuzzy Logic, cgi, 0023, *Computer Graphics International 2001* (CGI'01), 2001

[Shewchuk1996] Jonathan R. S., Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, in Applied Computational Geometry: Towards Geometric Engineering, volume 1148 of *Lecture Notes in Computer Science*, pages 203-222, Springer-Verlag (Berlin), May 1996.

[Shewchuk2002] Jonathan R.: Delaunay Refinement Algorithms for Triangular Mesh Generation, *Computational Geometry: Theory and Applications* 22(1-3):21-74, May 2002.

[Schmidt2005] Schmidt R., Wyvill B., and Sousa M. C.: Sketch-Based Modeling With The Blobtree. In *SIGGRAPH '05 Technical Sketch*, Los Angeles, CA, USA, 2005.

[Schweikardt1998] Schweikardt E., Gross M.D.: Digital Clay: Deriving Digital Models From Freehand Sketches. *ACADIA '98, Seebohm T. and Wyk S. V. eds.*, Quebec City, Canada, 202-211, 1998.

[Seo2003] Seo H., Thalmann N. M.: An Automatic Modeling of Human Bodies from Sizing Parameters, *ACM SIGGRAPH Symposium on Interactive 3D Graphics*, ACM Press, 19–26, 2003.

[Syflex2004] Syflex: In The Cloth Simulator Behind Many 3D Productions, 2004. *http://www.syflex.biz/index.html*.

[Textile] Textile and C. T. C. (TC2).: Products - Body Scanner [www document]. URL http://www.tc2.com/Products/Product%20Bodyscan.htm.

[Thalmann1987] Thalmann N. M.., Thalmann D.: The Direction of Synthetic Actors in the film Rendez-vous à Montreal, *Computer Graphics and Applications*, *IEEE Computer Society Press*, 7, No.12, 9-19, 1987.

[Thalmann1995] Thalman D.: Applications Of Virtual Humans In Virtual Reality Virtual Reality Applications ISBN:0-12-227755-4 Pages: 271 - 282 1995

[Thalmann1997] Thalmann N. M. and Volino P.: Dressing Virtual Humans, *Lecture Notes in Computer Science, Springer Berlin / Heidelberg*, 1195 9-1997.

[Thalmann2004] Thalmann N. M., Seo H., Cordier F.: Automatic Modelling of Virtual Humans and Body Clothing. *J. Comput. Sci. Technol.* 19(5): 575-584. 2004

[Turquin2004] Turquin E., Cani M. P., and Hughes J.: Sketching Garments for Virtual Characters, *Proc. Eurographics Workshop Sketch-Based Interfaces and Modelling*, T. Igarashi and J.A. Jorge, eds., Eurographics, 175-182.2004

[Turquin2007] Turquin E., Wither J., Laurence Boissieux, Marie-Paule Cani, John Hughes: A Sketch-Based Interface For Clothing Virtual Characters,. *IEEE Comput. Graph. Appl.*, 27 (1), 72–81, 2007.

[Varley2000] Varley, P., Suzuki, H., Mitani, J., and Martin, R.: Interpretation Of Single Sketch Input For Mesh And Solid Models. *International Journal of Shape Modeling 6*, 2, 207–240. 2000.

[Vassilev2000] Vassilev T.: Dressing Virtual People. *In: Proceedings of Systemics. Cybernetics and Informatics*, Orlando, July 23-26, 2000.

[vicon] http://www.vicon.com/products/system.html

[Volino1997] Volino P, Magnenat-Thalmann N.: Developing Simulation Techniques For An Interactive Clothing System. In: *Proceedings of Virtual Systems and MultiMedia '97*, . p. 109–18.1997

[Vollmer1999] Vollmer J. ; Mencl R., Müller H.: Improved Laplacian Smoothing of Noisy Surface Meshes, *Computer Graphics Forum*, Volume 18, Number 3, September , 131-138(8). 1999

[Wang2002] Wang C. C. L., Smith S. S. F., and Yuen M. M. F.: Surface Flattening Based On Energy Model, *Computer-Aided Design*, 34, no.11, 823-833, 2002.

[Wang2003A] Wang C.C.L., Chang T.K.K. and Yuen M.M.F.: From Laser-Scanned Data To Feature Human Model: A System Based On Fuzzy Logic Concept, *Compute-Aided Design* 35, 241–253.2003.

[Wang2003B] WANG C. C. L., WANG Y., YUEN M. M. F.: Feature Based 3D Garment Design Through 2D Sketches. *Computer Aided Design* 35, 7, 659-72.2003

[Wang2007] Wang H. X., Markosian L.: Free-form Sketch, *In: 4th EUROGRAPHICS Workshop On Sketch-Based Interfaces And Modeling*. Pages 53-58.2007

[Weidlich2009] Weidlich D., Zickkner H., Riedel T., Bohm A..: Real 3D Geometry and Motion Data as a Basis for Virtual Design and Testing. *Proceeding of the 19th CIRP Design Conference-competitive Design*, pp583. 2009

[Werghi2007] Werghi, N.: Segmentation and Modeling of Full Human Body Shape From 3-D Scan Data: A Survey, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Volume: 37, Issue: 6, On page(s): 1122-1136, ISSN: 1094-6977.2007

[Werner1993] Werner H.M., Thalmann N. M., Thalmann D.: User Interface for Fashion Design, *Graphics Design and Visualisation, IFIP Trans*. North Holland, pp 197-204, 1993.

[Xiao2003] Xiao Y, Siebert P, Werghi N.: A Discrete Reeb Graph Approach For The Segmentation Of Human Body Scans. *In: Proceedings of the fourth international conference on 3-D digital imaging and modeling. IEEE Computer Society Press: Silver Spring, MD*. p. 378–85.2003

[Xiao2003] Xiao Y., Werghi N., and Siebert P.: A topological approach for segmenting human body shape, *in Proc. Int. Conf. Image Anal. Process.*, Mantova, Italy, 82–87.2003

[Xiao2003] Xiao Y., Werghi N., and Siebert P.: A topological approach based on discrete Reeb-graph for the segmentation of human body scans, *in Proc. IEEE Int. Conf. 3-D Imag. Model.*, Alberta, QC, Canada, 378–385. 2003.

[Yang2005] Yang C, Sharon D, Panne M. Sketch-Based Modeling Of Parameterized Objects. *In: 2nd EUROGRAPHICS Workshop On Sketch-Based Interfaces And Modeling*. 2005.

[Yu2006] Yu H., Qin S. F. and Wright D. K, Conceptual Framework For Interactive Virtual Storytelling, *Dcc'06 Intelligent Virtual Design Enviornment Workshop*, 2006.

[Yu2007] Yu H., Qin S. F., Wright D. K.: Generation of 3D Human Models with Different Levels of Detail through Point-Based Simplification. *In Proceedings of the International Conference on "Computer As a Tool" IEEE EUROCON 2007*, 1882-1886.2007

[Zeleznik1996] Zeleznik, R. C., Herndon, K. P., and Hughes, J. F.: SKETCH: An Interface For Sketching 3D Scenes. *In Proc. of SIGGRAPH '96*, 163–170.1996.

[Zhong2006] Zhong Y. Q., Xu B. G.: Automatic Segmenting And Measurement On Scanned Human Body, *International Journal of Clothing Science and Technology*, Volume:18, Issue:1 Page:19 - 30. 2006.

[Zimmermann2007] Zimmermann J., Nealen A., Alexa M.: SilSketch: Automated Sketch-Based Editing Of Surface Meshes, *In: 4th EUROGRAPHICS workshop on sketch-based interfaces and modeling*. Pages 23-30. 2007.

[Zorin1996] Zorin D, Schroder P, Sweldens W.: Interpolating Subdivision For Meshes With Arbitrary Topology. *SIGGRAPH 96 Proceeding*, New York, USA, ACM; p. 189–92.1996.

Appendix A Publications

Hui Yu, Shengfeng Qin, Guangmin Sun,David Wright, A Sketch-based Method for Dressing Virtual Humans (under review).

Hui Yu, Shengfeng Qin, Guangmin Sun, David Wright, Design Your own Garments: An Intuitive Clothing Platform (under review).

Hui Yu, Shengfeng Qin, David Wright, Dressing Virtual Humans From 3D Scanned Data, *Biomedical Sciences Instrumentation*, Vol. 44, 2008. pp.311-316. ISSN 0067-8856.

Hui Yu, Shengfeng Qin, Guangmin Sun, David K. Wright. Automatic Segmentation Of Scanned Human Bodies For Building LOD Models. *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference* on Volume, Issue, 13-16 July 2008 Page(s):1426 - 1431

Hui Yu, Sheng-feng Qin, David K.Wright, Jin-sheng Kang, Generation of 3D Human Models with Different Levels of Detail through Point-Based Simplification, *IEEE EUROCON 2007 "Computer As a Tool"*, Warsaw, September 9-12, 2007, ISBN 1-4244-0813-X. pp.1982-1986.

Yu Hui, Qin SF, and Wright DK, A Conceptual Framework For Interactive Virtual Storytelling, *Proceedings of Second International Conference on Design Computing and Cognition (DCC06)'s Workshop on Intelligent Virtual Design Environments (IVDE)*, July 9, 2006, Technical University of Eindhoven, Netherlands.