

A research on automatic tailoring and cloth modelling for animation character

WENXI LI

A thesis submitted in partial fulfilment of the requirements of
Bournemouth University for the degree of

Doctor of Philosophy



Supervisor: Prof. Jian Jun Zhang, Dr. Xiaosong Yang

December, 2013

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Abstract

The construction of realistic characters has become increasingly important to the production of blockbuster films, TV series and computer games. The outfit of character plays an important role in the application of virtual characters. It is one of the key elements reflects the personality of character. Virtual clothing refers to the process that constructs outfits for virtual characters, and currently, it is widely used in mainly two areas, fashion industry and computer animation.

In fashion industry, virtual clothing technology is an effective tool which creates, edits and pre-visualises cloth design patterns efficiently. However, using this method requires lots of tailoring expertises. In computer animation, geometric modelling methods are widely used for cloth modelling due to their simplicity and intuitiveness. However, because of the shortage of tailoring knowledge among animation artists, current existing cloth design patterns can not be used directly by animation artists, and the appearance of cloth depends heavily on the skill of artists. Moreover, geometric modelling methods requires lots of manual operations. This tediousness is worsen by modelling same style cloth for different characters with different body shapes and proportions.

This thesis addresses this problem and presents a new virtual clothing method which includes automatic character measuring, automatic cloth pattern adjustment, and cloth patterns assembling.

There are two main contributions in this research. Firstly, a geodesic curvature flow based geodesic computation scheme is presented for acquiring length measurements from character. Due to the fast growing demand on usage of high resolution character model in animation production, the increasing number of characters need to be handled simultaneously as well as improving the reusability of 3D model in film production, the efficiency of modelling cloth for multiple high resolution character is very important. In order to improve the efficiency of measuring character for cloth fitting, a fast geodesic algorithm that has linear time complexity with a small bounded error is also presented. Secondly, a cloth pattern adjusting genetic algorithm is developed for automatic cloth fitting and retargeting. For the reason that that body shapes and proportions vary largely in character design, fitting and transferring cloth to a different character is a challenging task. This thesis considers the cloth fitting process as an optimization procedure. It optimizes both the shape and size of each cloth pattern automatically, the integrity, design and size of each cloth pattern are evaluated in order to create 3D cloth for any character with different body shapes and proportions while preserve the original cloth design.

By automating the cloth modelling process, it empowers the creativity of animation artists and improves their productivity by allowing them to use a large amount of existing cloth design patterns in fashion industry to create various clothes and to transfer same design cloth to characters with different body shapes and proportions with ease.

Contents

Copyright statement	i
Abstract	ii
Table of contents	iv
List of figures	vii
List of tables	xi
Abbreviations	xii
List of symbols	xiii
Acknowledgements	xv
Declaration	xvi
1 Introduction	1
1.1 Fashion Design and Manufacturing	2
1.2 Virtual Clothing in Fashion Industry	7
1.3 Virtual Clothing in Computer Animation	9
1.3.1 Cloth Modelling	11
1.3.2 Cloth Simulation	12
1.4 Motivation	12
1.5 Research Aims and Objectives	15
1.6 Contributions	16

1.7	Thesis Outline	17
2	Literature Review	19
2.1	Patternmaking in Fashion Design	20
2.2	Anthropometry	22
2.3	Virtual Clothing in Computer Graphics	27
2.3.1	Geometrical Based Virtual Clothing Methods	27
2.3.2	Physical Based Virtual Clothing Methods	34
2.3.3	Hybrid Virtual Clothing Methods	38
2.4	Geodesics	40
2.5	Genetic Algorithm	45
2.6	Summary	49
3	Character Measurements Extraction	53
3.1	Introduction	59
3.2	Geodesic Curvature Flow	63
3.3	Geodesic on Mesh	70
3.3.1	Tangent Space Constraint	70
3.3.2	Implicit Euler Scheme	71
3.3.3	Least Squares Scheme	72
3.3.4	Geodesic Path Projection	76
3.3.5	“Continuous Dijkstra” Propagation	81
3.4	Performance Analysis	95
3.4.1	Efficiency	95
3.4.2	Accuracy	98
3.5	Results and Conclusions	107
4	Cloth Modelling and Re-targeting	114
4.1	Introduction	114
4.1.1	Patternmaking	117
4.1.2	Pattern Resizing Criteria	120

4.2	Cloth Resizing Algorithm	122
4.2.1	Genetic Algorithm	127
4.2.2	Definition of Population	132
4.2.3	Crossover and Mutation	136
4.2.4	Evaluation and Selection	143
4.3	Pattern Assembling	153
4.4	Conclusions	163
5	Conclusion and Future Work	166
5.1	Conclusion	166
5.2	Future work	170
References		171

List of Figures

Figure 1.1	Standard Body Template in fashion design(Rosen 2004)	3
Figure 1.2	UK, US, Euro Size Chart convert table	6
Figure 1.3	Animation production pipeline(Chang 2014)	10
Figure 2.1	The peplos(left) and the chiton(right) was the common cloth wear by woman in ancient Roman (McManus 2003). . .	21
Figure 2.2	Body proportion in ancient Egypt	23
Figure 2.3	The Vitruvian Man	24
Figure 2.4	Anthropometry measurements	26
Figure 2.5	Catenary curves for simulating hanging cloth(Weil 1986)	28
Figure 2.6	Wrinkle patterns in (Decaudin et al. 2006)	30
Figure 2.7	2D coordinates on triangle	36
Figure 2.8	Deformation process in (Müller & Chentanez 2010) .	37
Figure 2.9	An example of character design	41
Figure 2.10	Taking of measurements in fashion industry (Brad Bird 2007).	42
Figure 3.1	Datum points on a female body(Xiong 2008)	55
Figure 3.2	Two circumference measurements on character	56
Figure 3.3	Measurements in different postures	58
Figure 3.4	Illustration of geodesic curvature	59
Figure 3.5	Classification of vertex on a polyhedral surface	61
Figure 3.6	Geodesic on a polyhedral surface	62
Figure 3.7	Geodesic curvature on a piecewise curve	66

Figure 3.8 Derived point caused by updating its location within tangent space.	76
Figure 3.9 Geodesic on a smooth surface and its projection on its inscribed polygon mesh	77
Figure 3.10 Projection of a geodesic path.	79
Figure 3.11 Formation of an initial path for a unvisited vertex	82
Figure 3.12 Tape measuring on human body	90
Figure 3.13 Curve normal and vertex normal on a geodesic path on the arm of character A	91
Figure 3.14 Demonstration of path correction process	93
Figure 3.15 A straightened path for measuring arm length.	94
Figure 3.16 Geodesic path on a bent arm of Character A	94
Figure 3.17 Shape corrected path with a section selected by user (green section) to be ignored by shape correction process	95
Figure 3.18 Running time comparison for geodesic algorithms	97
Figure 3.19 Geodesics on a sphere	99
Figure 3.20 Distribution of the relative errors of MMP and Algorithm 2	99
Figure 3.21 Distribution of the relative errors of MMP approximate algorithm and Algorithm 3	100
Figure 3.22 Error estimation of Algorithm 3	101
Figure 3.23 Error estimation of the approximate algorithm with different window size	103
Figure 3.24 The convergenecy of error to the window size of the approximate algorithm	104
Figure 3.25 Performance of Algorithm 3 with varying window size.	104
Figure 3.26 Histogram of the ratio ε on the bunny model	106
Figure 3.27 The distribution of estimated errors and real errors	107
Figure 3.28 Measurements on Character A	108

Figure 3.29 Measurements with geodesic shape correction on Character A	109
Figure 3.30 Measurements on Character B	110
Figure 3.31 Measurements on Character C	111
Figure 3.32 Measurements on Character D	112
 Figure 4.1 Workflow of pattern based cloth modelling method presented in this thesis	117
Figure 4.2 Block(left) and a pattern(right) made from it by adding details(Howland 2008).	118
Figure 4.3 Garment blocks for woman shirt(Rosen 2004).	119
Figure 4.4 Same shirt on different characters	120
Figure 4.5 An example of a cloth pattern	123
Figure 4.6 Pattern and its block	125
Figure 4.8 Blocks after proportional scaling is performed.	125
Figure 4.7 Blocks before and after the proportional scaling	126
Figure 4.9 Inconsistency of seam-line after scaled patterns proportionally	126
Figure 4.10 Single objective genetic algorithm	131
Figure 4.11 Association between body datum point and pattern landmarks	134
Figure 4.12 Crossover of parents	137
Figure 4.13 Block mutation validation	142
Figure 4.14 A seam-line on patterns	145
Figure 4.15 Shape evaluation for patterns	147
Figure 4.16 Dressing two characters	148
Figure 4.17 “Front” pattern for both character in Figure 4.16 . . .	149
Figure 4.18 Fit shirt modelled for characters	150
Figure 4.19 Domination between five solutions	151
Figure 4.20 Patterns in a box	155

Figure 4.21 Bounding surfaces on character	155
Figure 4.22 3D patterns	156
Figure 4.23 3D pattern sewing	156
Figure 4.24 Characters	158
Figure 4.25 Cloth patterns(Xiong 2008), standard shirt patterns(left); standard trousers patterns(right)	158
Figure 4.26 Dressing result for character A	159
Figure 4.27 Dressing result for character B	159
Figure 4.28 Dressing result for character C	160
Figure 4.29 Dressing result for character D	160
Figure 4.30 Front view of dressed up characters	161
Figure 4.31 Rear view of dressed up characters	161
Figure 4.32 Characters used for dressing various clothes	162
Figure 4.33 Tight dress on different characters	162
Figure 4.34 T-Shirt and short on different characters	163
Figure 4.35 Skirt on different characters	163

List of Tables

Table 2.1	The basic anthropometry measurements	25
Table 3.1	The definitions of the measurements and their associated datum points (Armstrong 2000; EN:13402 2001)	54
Table 3.2	Resolution of the bunny models	97
Table 3.3	Error of Algorithm 3 on character models	105
Table 3.4	Measuring time on different characters.	107
Table 3.5	Measurements for characters A.	109
Table 3.6	Measurements for characters B.	110
Table 3.7	Measurements for characters C.	111
Table 3.8	Measurements for characters D.	112
Table 4.1	The structure of gene	133
Table 4.2	Fitness values of the final solution	161

Abbreviations

<i>WWII</i>	Second World War
<i>B.C.</i>	Before Christ
<i>2D</i>	Two Dimensional
<i>3D</i>	Three Dimensional
<i>CAD</i>	Computer Aided Design
<i>PCA</i>	Principal Component Analysis
<i>KES</i>	Kawabata Evaluation System
<i>FAST</i>	Fabric Assurance by Simple Testing

List of Symbols

C	Smooth curve
L	Euclidean distance
S	Smooth surface
Tp	The tangent plane of a surface p
C'	First order derivative of C
C'_s	First order derivative of C at point that has arc length s to C
C''	Second order derivative of C
C''_s	Second order derivative of C at point that has arc length s to C
κ	Curvature
τ	Torsion
κg	Geodesic curvature
\mathbb{R}^3	Three-dimensional Euclidean space
$C(t)$	A curve in Euclidean space parametrized as a function of time
s	Arc length of a curve
$C(s)$	A point on curve C at arc length s to C
$C(s, t)$	Geodesic curvature flow
\vec{n}	Normal vector at a point on S
\vec{t}	Tangent vector at a point on curve C

\vec{n}_p	Principle normal vector at a point on curve C
\vec{n}_b	Binormal vector at a point on curve C
$\overrightarrow{p_ip_{i+1}}$	A vector from p_i to p_{i+1}
$\langle A, B \rangle$	Dot product of vector A and B
\times	Cross product of two vectors
$\widetilde{p_{i-1}p_ip_{i+1}}$	A section of the piecewise curve consisted by three points
μ	Iterative step length for geodesic curvature flow convergence
p_{i_x}	The x value of the three-dimensional Euclidean coordination of point p with index i
p'	The new location of point p after it is updated by geodesic algorithm presented in this thesis
\bar{p}	The projection of point p on to the tangent plane Tp
fp_i	Vertex on polyhedron with index i

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Prof. Jian J. Zhang and Dr. Xaosong Yang, whose encouragement, patience, devotion, guidance and support have enabled me to develop a good understanding of the subject throughout various stages of my research. They have great personalities and extensive knowledge. I would not have complete my research without their invaluable advice. I shall never forget them. I would also like to thank Dr. Hongchuan Yu for his excellent ideal and great support on the research of the geodesic algorithm. Special thanks to my friends Rudra Poudel and Sola Aina for proofreading my thesis carefully and provided many valuable suggestions on various aspects to my the research. I am also grateful to Mrs. Jan Lewis and for her daily support and kind assistance with academic matters. In addition, I would like to thank my best friends Safa Tharib, Kaiping Zhang, Meili Wang and Shaoyong Liu for the invaluable friendship.

I gratefully acknowledge Chinese Scholarship Council for their financial support. I would also extend my regards and blessings to all of those who supported me in any respects during the completion of my PhD. My great tribute is due to my parents, and my wife for their loving consideration and invaluable support through my long student life. Finally, I dedicate this thesis to my family, for their love and encouragement on my journey of growth.

Declaration

This thesis has been created by myself and has not been submitted in any previous application for any degree. The work in this thesis has been undertaken by myself except where otherwise stated.

Chapter 1

Introduction

Cloths not only reflect a persons social status but also act as a medium of communication in peoples daily life(Barnard 2002). In general, modern cloth making process involves five steps(Margolis 1964), choose a style, measure the body of a customer, adjust cloth patterns, assemble patterns and final try-on. For bespoke clothing, each pattern is cut slightly larger than desired size, before final assembling, patterns will be loosely stitched together and put onto the customer to further trim off the excess to achieve better fit.

In fashion industry, cloth patterns are the feasible solution of a fashion design concept. It is constituted by a set of 2D outlines which are the breakdowns of a complete cloth design(Rosen 2004). Cloth design process are usually carried out on an ideal figure which is called standard body template. In order to produce textile pieces that can be assembled to form a cloth for a costumer, cloth patterns need to be adjusted to a particular size for the customer by an experienced tailor. Nowadays, cloth pattern is the most important medium in fashion industry. Almost all the cloth designs are preserved or distributed in the form of cloth patterns. Cloth pattern also provides an intuitive instruction for users to implement a cloth. Moreover, by altering certain parts of a cloth pattern, the cloth can be adjusted into any desired size

for any wearer.

1.1 Fashion Design and Manufacturing

“Fashion design is the art of the application of design and aesthetics or natural beauty to clothing and accessories. Typically, fashion refers to the phenomenon of a regular pattern of change in the prevailing mode of dress.”(Steele 2005). In this thesis, the term “Fashion Design” only refers to cloth design. Cloth making is the process that transfers design concept into wearable objects. The history of cloth making follows tightly with the evolution of human. It is not known when human started to wear cloth, but the evidence shows that man start wearing cloth from 170,000 years ago(Dunn 2012). In Ancient times, clothes are usually made of natural elements such as animal skins or furs, plant leaves, bones and shells that was draped or tied onto body. The appearance of textile during the late stone age in Middle East(Lynch et al. 1985) not only improved both comfort and functionality of cloth at that time but also enabled human to produce wearable materials by themselves. The discovery of bone needles which can be dated to 61,000 BP(Backwell et al. 2008) reveals the evidence of sewn textiles or leathers used to make cloth at that age.

During Middle Ages, cloth has already become a symbol of wealth and social status. Fashion was the privilege of upper class only, who can afford to hire private dressmaker to make cloth for them(Crane 2012).

Modern fashion that every one could enjoy was brought to general public by Rose Bertin, who was the dressmaker to Marie Antoinette(2 November 1755 – 16 October 1793), Queen of France. It is said that she was the first one who starts the transition from low public profile private dressmakers to a well-known fashion designers. The cloth shop she opened in Paris had a considerable influence on Parisian style(Steele 1998). In 1863, Ebenezer Butter-

ick invented the first graded sewing pattern for cloth making (Hannah 1919; O'Loughlin 1899). Soon after his invention, cloth pattern became massively popular, as they made modern fashions much more easier to be accessed by the rapidly expanding lower middle class. This innovation provides a standard for fashion designs to be preserved and spread with much lower cost, moreover, it allows a design to be adapted to different human figures with ease.

The term “Fashion Design” is usually refers to the process of constructing the cloth design concept and producing the samples on the standard body template. Figure 1.1 demonstrates a standard body template which is the ideal human body that has ideal body proportion.

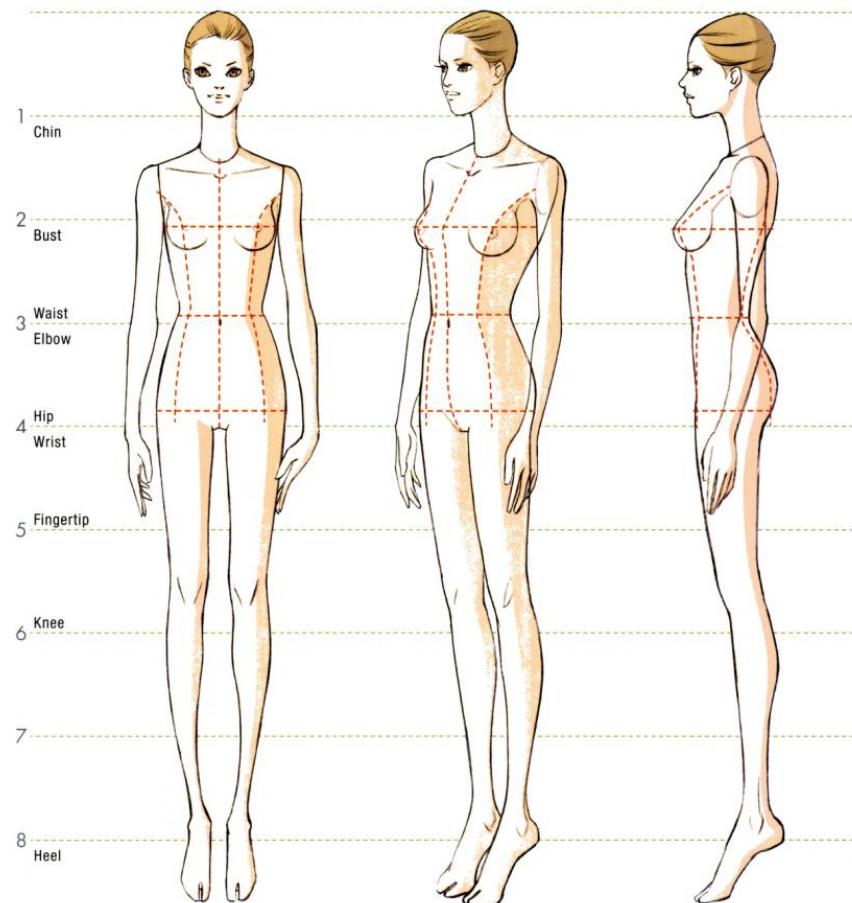


Figure 1.1: Standard Body Template in fashion design(Rosen 2004)

Modern fashion design process involves a five steps process to transfer a design idea into a wearable sample cloth (Stecker 1996) :

Design Inspiration and Sketches

Fashion designers constantly searching for the new idea to create new style. They record their inspiration of the new style by using rough and abstract sketches. Simple drawing allows more room for modifying and expressing their thoughts.

Fabric Selection

There are many types of textile materials with different method of interlacing and interloping fabric materials that provide different physical proprieties. Different types of textile materials also affect the appearance of cloth differently. Based on the type of cloth, designers select suitable textile materials for the design.

Final Sketches

By using standard body template, colours and details such as pockets, seam lines and other accessories are added onto the sketch. The final sketch is a detailed preview of the design worn by a model.

Patternmaking

Before patterns can be produced, fabrics are positioned or pinned on a dress form to reveal the structure of a garment design, this process is called “Draping”(Stecker 1996), it allows designer to visually exam the design on a human body. After the design has been draped, patternmaker decomposite draped cloth design and transfer it onto paper. Both cloth details such as seams, darts, cuttings and sewing instruction are indicated on the paper pattern.

Samples and Flat Drawings

After paper patterns are produced, they first pinned on top of the fabric for cutting and then removed allowing fabric pieces to be sewn together.

For design storage and distribution, both patterns and flat drawing are required. Flat drawing acts as a blueprint of a design which contains all the details for constructing clothes.

The aforementioned fashion design process involves many iterations of transferring design concept from 2D sketches to 3D draping, then back to 2D patterns. In other words, designers express their idea by 2D sketch, patternmaker reconstructs the 3D cloth structure in real world and decomposes the structure back to a set of 2D patterns. Finally, 3D sample cloth is produced based on 2D patterns. Often the resulting sample cloth reveals desired modification in the 3D form of the design that, in turn, the revisions of the 2D patterns also need to be carried out. Finally the new sample is created for further inspection. This iteration continues till the final sample is signed off by designer.

After cloth design process is completed, next step is to produce cloth for customers, and “Cloth Making” refers to the process of producing clothes based on cloth patterns. This process is aiming for producing clothes with certain sizes for actual customers and it is constituted by four steps:

Measuring

Different individual has different body shapes and proportions. In order to fit a cloth to a customer, body measurements need to be extracted. This step acquires measurements of the body parts that are associated with the cloth. Tape ruler is the most common tool for performing measuring on a customer.

Pattern Grading

Because cloth patterns and flat-drawing are developed on the standard body template, the size of each patterns need to be altered in order to fit a customer. Based on the measurements of customer, corresponding parts of patterns are adjusted to meet measurements.

sewing

After patterns are adjusted to desired size, fabric pieces are cut out slightly large than required size. Then patterns are stitched loosely to form the initial cloth.

Try-on

At this step, cloth is fitted to customer to further adjust the stitch and trim off the excess in order to achieve best fit. Finally, all the patterns are sewn together to form the final garment.

For the massively produced cloth, in order to make clothes that can be wore by general public, cloth patterns are scaled to several predefined sizes based on a size chart. Size chart contains body size data from a particular ethnic or group of people who shares the similar body proportion. The size chart various among different countries or areas, i.e. US Size Chart (ISO/TR-10652 1991), UK Size Chart (EN:13402 2001), Euro Size Chart (Ashdown 2007). Table1.2 illustrates converting rule between different size chart and major measurements used to define cloth size in a size chart.

UK	USA	Europe	Bust	Waist	Hips
8	4	34	31/32" - 79/81cm	23/24" - 58/61cm	33/34" - 84/86cm
10	6	36	33/34" - 84/86cm	25/26" - 64/66cm	35/36" - 89/91cm
12	8	38	35-36" - 89/91cm	27/28" - 69/71cm	37/38" - 94/96cm
14	10	40	37/38" - 94/96cm	29/30" - 74/76cm	39/40" - 99/101cm
16	12	42	39/40" - 99/101cm	31/32" - 79/81cm	41/42" - 104/106cm
18	14	44	41/42" - 104/106cm	33/34" - 84/86cm	43/44" - 109/111cm

Figure 1.2: UK, US, Euro Size Chart convert table

1.2 Virtual Clothing in Fashion Industry

Traditional fashion design and cloth making procedure are considered as highly skilled, labour intensive process that require many iterations to complete. Followed with the advancement of computer hardware and computer graphic techniques, constructing and evaluating cloth design in virtual environment instead of actually performing the cloth making process repetitively dramatically cuts down labour and time consummation required in traditional cloth design and cloth making process.

Virtual Clothing is the generic term of the process which constructs both visual effects and physical behaviours of textile objects(Volino & Thalmann 2000). It is the foundation of cloth CAD/CAM technologies which are widely used in fashion industry to assist the design of garment product. In terms of composition, cloth is constituted by a set of shape predetermined textile pieces which are assembled following a predefined order to cover a specific body area. The aim of virtual clothing techniques in fashion industry is to produce cloth patterns and evaluate cloth design effectively and efficiently.

In order to construct cloth in a virtual environment, two fundamental requirements need to be fulfilled, 2D cloth patterns from fashion designers and 3D virtual mannequins that are used for cloth assembling and 3D evaluation. In general, four steps are involved in the process of constructing virtual cloth, 2D cloth pattern generation, assembling 2D patterns to 3D cloth, 3D cloth simulation, comfort evaluation. The process of constructing virtual cloth generally follows the procedure of the making real cloth, however, because cloth patterns are created, edited, graded and pre-visualised in virtual environment, implementing cloth design into a real cloth for evaluation can be avoided, therefore, improve the efficiency and reduce the cost of fashion design.

There are various applications targeting to each stages of cloth design to improve the productivity of designers.

2D Pattern Editing

Computer aided cloth pattern design system such as Kaledo(Lectra 2014) and Cameo(Wid Ginger Software 2013) have been widely used in various fashion brands across the world. Traditionally, cloth patterns are usually made of paper boards cut out by patternmakers. In order to make cloth for different customers with different body shapes and proportions, the standard cloth patterns need to be adjusted into different sizes by experienced tailors based on the measurements of customer. With the help of cloth pattern CAD/CAM system, creating, editing and grading cloth patterns can be performed in virtual environment. Especially in large-scale cloth production, cloth pattern CAD/CAM methods not only save the materials but also provides an intuitive and convenience approach for cloth pattern manipulation.

Cloth 3D visualization

Cloth 3D visualization techniques generate 3D cloth object in virtual environment either based on 2D cloth patterns(Fontana et al. 2005; Marvelous 2014; Optitex 2014) or designer's sketches(Turquin et al. 2007b). Traditionally, visualizing 2D cloth patterns requires the most iterations and labours in traditional fashion design process. Any changes on 3D form need to be amended on 2D patterns, and a new sample cloth needs to be produced in order to visualize the amendments. These iterations between 2D and 2D perspective consume materials and time. 3D visualization techniques eliminate these costs by creating and assembling cloth patterns in virtual environment. Cloth pattern can be edited and visualized without need of actually producing cloth from textile pieces, which saves both time, materials and labours.

Virtual Try-on

Virtual try on techniques enables designers to fit cloth onto avatars with different body shapes and proportions to review the appearance, fit and degree of comfort of clothes. It also allows customers to visually try clothes without walking into local shops. Different from aforementioned virtual clothing applications which focus at cloth design on the standard body template, this type of techniques usually involves body measurements acquisition from real customers, 3D body reconstruction, cloth modelling and cloth simulation. It provides the most intuitive visual experiences of a cloth design. Moreover, thanks to the advancement of 3D scanning techniques, the body shape and its measurements can be acquired automatically from customers. Combining with the 2D pattern editing techniques, fashion designers are able to produce custom made cloth with much higher efficiency and lower costs.

Textile Engineering

Textile is a type of complex structure materials which is made of fabric threads that are inter laced or looped together. Cloth simulation utilises physical models to describe mechanical and structural properties of textiles and provides an effective tool for textile engineers to create, simulate and analyse the dynamic properties of textiles constructed with various weaving techniques and materials.

1.3 Virtual Clothing in Computer Animation

Differ from virtual clothing techniques in fashion industry, virtual clothing techniques in computer animation mainly aim at producing realistic appearance of 3D cloth objects. In animation, apart from the character's body and facial expression, costumes also play a very important role in acting. It is the symbol of social statue and personality of characters.

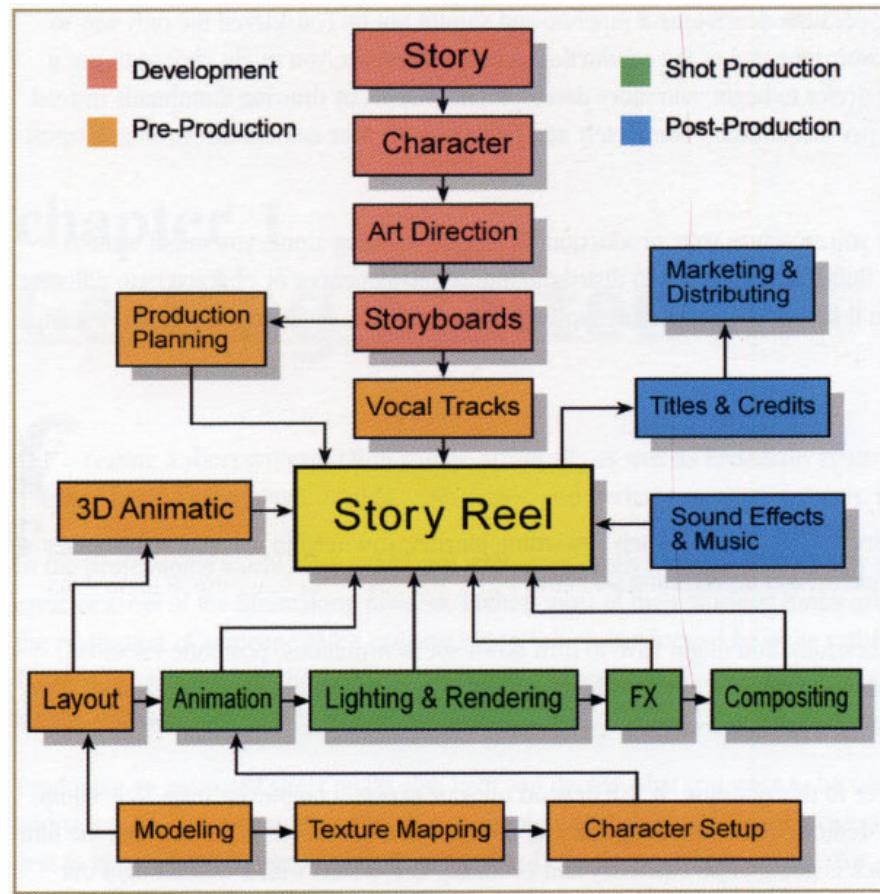


Figure 1.3: Animation production pipeline(Chang 2014)

Figure 1.3 demonstrates the production pipeline of animation film. The process of creating outfits starts from character(Character Design) phase which is at very early stage of the development process. Outfits are designed together with characters. In the modelling stage, clothes are modelled after the completion of character modelling. At this point, animators utilise various modelling method to build 3D cloth. During character setup, deformers are assigned to character model to drive its movement. Depends on the requirement of the film, some clothes are driven by skeleton system. For realistic clothes, they are simulated by cloth simulation module. Therefore, based on the animation production pipeline, virtual clothing in computer animation is constituted by two parts, cloth modelling and cloth simulation

1.3.1 Cloth Modelling

Cloth modelling is the process of constructing geometrical shape of cloth in a virtual environment. Due to the vast variety of the character designs, the intuitiveness and controllability of cloth modelling method are crucial to animation artists. In computer animation, beside the pattern based cloth modelling method which is widely used in fashion industry, the most popular modelling method for cloth is general-purpose geometric modelling method due to its intuitiveness and ease of use. By using this method, cloth can be modelled same way as other objects are modelled such as character, building and vehicles. Without the need of tailoring knowledge, this type of method operates directly on the basic element of the geometrical representation of clothes, therefore, it gives the user maximum controllability throughout the modelling process and the appearance of 3D cloth can be viewed throughout the entire modelling process. However, the quality of cloth model depends on the artistic scenes of its modeller. Different modeller may produce different cloth from same cloth design. Moreover, because cloth is a soft object that follows the profile of character body, it cannot be dressed onto another character which has different body shapes and proportions without major modifications. In order to modify a cloth for a different character, heavy workload is required to perform such a process.

On the contrary, pattern based cloth modelling method operates on cloth patterns. Cloth patterns define the appearance of cloth, thus the consistency of the style of a cloth modelled different animators can be maintained. However, to form a 3D cloth from 2D cloth patterns requires expertises in tailoring and few animation artists possess such a skill. Moreover, in order to evaluate a cloth, the pattern assembling process need to be completed to form the 3D cloth and any modification that are required need to be executed on 2D cloth patterns which makes editing a cloth notoriously tedious due to the repetition of modelling process for modifying cloth patterns.

1.3.2 Cloth Simulation

Cloth simulation is the process of reproducing dynamic behaviours of cloth objects in a virtual environment. Based on methods used for driving deformation of a cloth, in general, there are two types of cloth simulation methods used in the computer animation.

Physical cloth simulation utilises a physical model such as mass-spring model or energy model to describe mechanical properties of a piece of cloth. By using the mechanical model developed from the real world, this method is able to reproduce realistic dynamic behaviours of clothing. However, because the detail of cloth deformation is determined by the number of the basic elements of the geometrical representation of cloth, this process requires large amount of computational power and time to generate fine detailed cloth deformation.

Hybrid cloth simulation methods utilise other types of deformation models such as data-driven method or geometrical modelling method to generate fine deformation details of cloth instead of performing high resolution physical simulation. The global deformation of cloth is usually generated by using coarse physical simulation. The efficiency of cloth simulation can be improved significantly by using hybrid cloth simulation method rather than performing physical simulation solely.

1.4 Motivation

In the past, the number of characters that can be handled in a virtual environment is limited by the computational power of hardware. In order to cope with this limitation, cloth of a virtual character is usually considered as a second layer skin of a character. Therefore, character and its cloth cannot be separated. However, because texture and dynamic properties of clothes are

much differ from skins, the bound between the cloth and the character makes cloth simulation infeasible.

The latest advancement of computer hardware results an increase of the computational power. The physical simulation of cloth became possible. Although cloth can be modelled and simulated separately from character skin, the current methods for constructing cloth for virtual character still requires large amount of manual operations. In order to reuse a cloth to different characters, substantial amount of work need to be carried out. The duplication of effort cannot be eliminated when dressing different characters with different body shapes and proportions.

For virtual clothing in fashion industry, pattern based cloth modelling techniques are widely used . While in film and gaming industry, most artists still rely on the general-purpose modelling software packages such as Maya, Softimage, 3D Studio MAX to construct cloth for their characters.

In general, the workflow of pattern based cloth modelling technique consists of two steps, 2D pattern generation and pattern assembling. The first step correlates to the patternmaking process in real cloth production. Patternmaking in fashion industry is a highly skilled task, that only the well trained experts are able to master. However, few animation artists possess this skill which makes it difficult for animators to produce correct cloth pattern for a cloth. Moreover, body shapes and proportions varies largely among animation characters, and often, animation characters have very exaggerated body proportion which is far away from the normal human body proportion. In order to dress a character, cloth patterns need to be resized to cope with the body of character. In real world, this process is done by either professional patternmaker or experienced tailor. However, in computer animation, size chart used for resizing cloth pattern in fashion industry no longer applies to animation character.

During cloth modelling process, there are two key factors directly re-

lated to the appearance of a modelled cloth, the preservation of cloth design and the fitting of the cloth. In order to achieve these two objectives to improve the reusability of a modelled cloth, several techniques have been developed for modelling clothes for different characters with different body shapes and proportions, such as geometric morphing techniques (Ebert et al. 2006; SmithMicro 2012) and constrained optimization method (Brouet et al. 2012). For geometric morphing techniques, although it can fit clothes to different characters, but the design of cloth can be largely different from its original when topology of character models are largely different. Furthermore, Brouet et al. (2012) have proposed series of methods that directly operate the geometry of the 3D cloth on the new character in order to fit the cloth while maintaining the cloth design. However, the final 2D cloth design patterns need to be extracted by using surface flattening techniques(Sheffer et al. 2005). It not only requires an excess computation, but also the shape of the pattern can not be preserved from the surface flattening process.

With the fast development of computer power, more and more characters can be handled simultaneously. Modelling is a very labour intensive process in the film production pipeline.

The tediousness and unintuitiveness of current cloth modelling method have become the bottleneck of improving the reusability of cloth models which further affect the cost of film production. Moreover, for pattern based cloth modelling method, the requirement of deep tailoring knowledge has become the largest Obstacle for applying it into animation production. The research presented in this thesis aims at providing an efficient and easy-to-use pattern based cloth modelling method for computer animation. By automating the process which requires tailoring knowledge in pattern based cloth modelling method, the large amount of cloth designs exist in the fashion industry will become usable for the construction of cloth for animation character. Also, by defining cloth shape in 2D cloth patterns, clothes can be stored and distributed

with ease. Moreover, this method adjusts cloth pattern sizes and shapes based on the character body measurements, cloth designs can be reused and retargeted to different characters while maintaining its style.

1.5 Research Aims and Objectives

Although the geometric cloth modelling method provides lots of controllability to the animation artists, the quality and diversity of clothes is largely limited by the skill of animation artists. The current pattern based cloth modelling method requires many repetition for creating and editing 3D cloth. However, its ability of using a large amount of existing cloth design patterns in fashion industry and stores cloth design in a uniformed form which can be reused later is still a very attractive advantage to animation artists.

The research presented in this thesis focuses at bridging the gap between real cloth making techniques and cloth modelling method in computer animation to provide an easy-to-use and intuitive pattern based cloth modelling methods for animation production use allowing animation artists to directly use existing cloth design patterns in fashion industry.

This method includes several techniques: character body measuring method which utilises a novel fast geodesic algorithm to extract length measurements, a evolutionary cloth pattern adjustment framework for automatic cloth pattern adjusting and cloth pattern assembling method. Given a character model and 2D patterns of a cloth design from fashion industry, the method presented in this thesis is able to fit the cloth to any character while maintaining the original cloth design. Moreover, once cloth pattern is created, it can be used on different character with little manual operation. The objective of this thesis are to:

1. Review recent works on different virtual clothing techniques for both

fashion industry and animation production and analyse their advantages and disadvantages for modelling cloth for characters with different body shapes and proportions.

2. Develop an efficient measuring technique for characters regardless of their postures.
3. Develop a measuring technique for the point cloud character model to cope with the increasing number of 3D scanned character models.
4. Develop a technique to adjust each cloth pattern automatically based on the measurements and preserve the style of cloth.

1.6 Contributions

The cloth modelling method presented in this thesis consists of two major parts, character measuring and cloth pattern adjustment. For character measuring, a geodesic based measuring method has been developed for length measurements and convex-hull is used for measure the circumference of character. In order to improve the efficiency of geodesic calculation, a linear time complexity approximate geodesic algorithm is also proposed. For cloth pattern resizing, in order to cope with the vast shapes and proportions difference among animation characters, a pattern adjustment genetic algorithm have been developed. The contributions of this thesis are as follows

1. A detailed review of work covering traditional cloth making techniques, recent research progress on computer-aided cloth design, and existing cloth modelling and simulation techniques.
2. This thesis describes an automatic pattern based cloth modelling method which bridges the gap between traditional tailoring techniques and cloth modelling techniques in computer animation. This enables the usage of

a large amount of existing cloth design in fashion industry to animation artists.

3. This thesis also describes a geodesic curvature flow based geodesic scheme for the measurements extraction. This scheme consists of two algorithms, one with high measuring accuracy and the other incorporates a small bounded error into the geodesic calculation to achieve faster measurement with linear time complexity.
4. A pattern adjusting genetic algorithm is proposed. Considering the measurement, seam-line among the patterns as well as the shape of each pattern, the original design of cloth can be preserved through out the fitting process. This is the first attempt to utilize the genetic algorithm in 2D cloth pattern adjusting problem for dressing different characters.

1.7 Thesis Outline

The reminder of the thesis is organised as follow:

Chapter 2 reviews the techniques related to the research presented in this thesis. Firstly, the history of tailoring techniques are introduced. Secondly a brief introduction of the anthropometry is conducted which is the key element to gain the correct human body measurement data from character model. Thirdly, the related works in cloth modelling and cloth simulation are reviewed. Finally, the recent research achievements in geodesic calculation and genetic algorithm are reviewed.

Chapter 3 introduces character body measuring method which utilises convex-hull computation for circumference measurements and a geodesic computation scheme for length measurements. Two geodesic algorithms that have different measuring accuracy and efficiency are described in detail. Fi-

nally, experiments that validate the accuracy of geodesic computation and its efficiency are demonstrated. The comparison between the presented algorithms and two most popular geodesic algorithms are presented.

Chapter 4 introduces an automatic cloth modelling and re-targeting method. The main functionalities of this method are outlined. This method utilises the genetic algorithm to adjust each cloth pattern in order to fit a cloth onto a character. The design of this genetic algorithm is explained in detail. A pattern assembling method is also introduced here. Finally, the same cloth design is fitted on to different characters with largely different body sizes and proportions using this method and the results are discussed.

chapter 5 draws the conclusions and the future works are discussed.

Chapter 2

Literature Review

Clothing is one of the most distinctive features of human being which differs us from other creatures. The history of cloth can be traced back to 107,000 years ago(Kittler et al. 2003; A.Toups et al. 2011). During the development of cloth, the basic function of cloth remains, which is to provide protection to the wearer from environment. Cloth also performs a wide range of cultural and social functions which expresses the personality, occupation, sexual differentiation, and social status of the wearer (Harms 1938).

In the world of today, with high speed development of computer hardware and computer graphic techniques, realistic virtual character has been wildly used in film, TV and game productions. Apart from body motions and facial expressions, same as in reality, cloth plays a very important role in acting. In order to achieve high visual realism, many techniques have been developed in areas such as motion capture, muscle simulation, or skin deformation, etc. However, in computer animation, virtual clothing which involves both textile engineering knowledge and artistic expertise, is considered as a challenging task. Especially for cloth modelling, it still requires large amount of manual operation and it is a very time-consuming task in the animation production.

Cloth pattern is the most common cloth design representation in the fashion industry. This thesis presents an pattern based cloth modelling method for animation artist which improves efficiency and reduce tediousness of current pattern based cloth modelling techniques by automating the pattern adjusting processes requires deep tailoring knowledges. This chapter firstly presents a general overview of the cloth production procedure in fashion industry. In order to make cloth that fits the target character, the measurements of character is crucial, therefore, a detailed introduction of research achievements in anthropometry study is also presented. After that, the state of the art virtual clothing techniques in both fashion industry and computer animation are introduced. Because the measuring method presented in this thesis utilises geodesic for length measurements extraction and uses genetic algorithm to adjust cloth patterns automatically, therefore, a brief review on geodesic algorithm and genetic algorithm finally is presented.

2.1 Patternmaking in Fashion Design

The process of constructing a complete cloth consists of several interdependent steps. Each step heavily affects the appearance and fit of the garment. Within these steps, patternmaking settles among the earliest few steps in the construction of a complete cloth. It is a highly skilled craft that has evolved over the centuries.

In ancient Roman period, producing textile materials was a laborious process. Fabric was weaved using primitive looms entirely by hand. Therefore, fabric was a very expensive commodity and it was an important symbol for the social status of the wearer. In terms of cloth structure at that time, cloth was mainly made by a set of uncut, rectangular shaped fabric pieces in order to minimize waste (Vout 1996).

In the 15th century, the seminal art of patternmaking began. The fab-

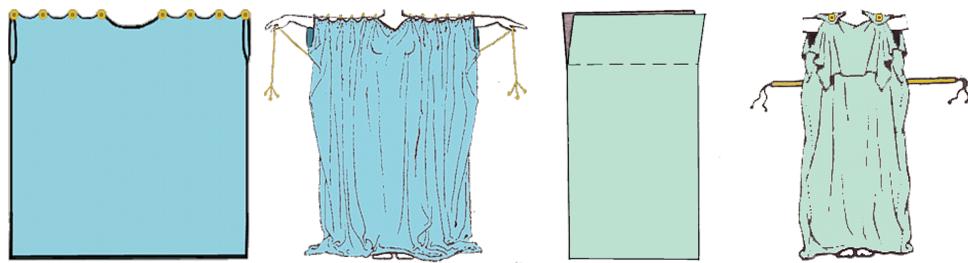


Figure 2.1: The peplos(left) and the chiton(right) was the common cloth wear by woman in ancient Roman (McManus 2003).

ric was carefully trimmed to fit the contour of body (MacDonald 2009). The foundation of the modern fashion design was built since then. Before the industrial revolution, patternmaking is a highly respected art. During that time, tailors choicely worked with the measurements taken from their costumer to handmade cloth patterns. Only the high society are able to afford tailor made cloth. During the industrial revolution, the standardization of cloth pattern has become essential to the popularization of massive produced cloth. However, initial attempts to create standardized cloth pattern resulted in a little detailed cloth that is poorly fitted. After large amount of experimentations and standardization of sizing regulation, patternmaking has successfully transferred from customization to standardization(MacDonald 2009). During that time, cloth patterns are made into one size and tailor had to grade cloth pattern to the size that wearer was needed. Because pattern-grading requires lots of tailoring expertise and experiences, cloth pattern was not widely accepted in home sewing. At the end of 19th century, Butterick introduced revolutionary graded cloth patterns(Butterick 1994) to the world. The effects of this idea were significant as it opens up the market of home sewing to the general public. Butterick's pattern not only made dressmaking much easier, but also pushed fashion from high class into general public all over the world. During the WWII, with the help of the development of anthropometry, sizing chart was introduced based on anthropometric data. The use of sizing chart

in graded patterns production have made it fit the general public better than before. Nowaday, after nearly a century, graded patterns has been widely accepted by today's fashion industry(Noke 1987; Rosen 2004).

Based on different target customers, today's patternmaking techniques can be classified into two general categorises, patternmaking for massive production(Nugent 2008; Shoben & Ward 1987; Stecker 1996; Staff 2007), and patternmaking for custom tailoring(Margolis 1964; Cabrera & Meyers 1983; Browne 2011). Patternmeking for massive production mainly focus at the ease of pattern distribution and standardization. This type of patternmaking techniques grade patterns into different predetermined size in order to fit the most of customers. In custom tailoring, patterns are distributed as an one size design concept, the final cloth requires experienced tailor to fine tune the size and shape of cloth patterns based on measurements taken from customer to create a fit cloth.

2.2 Anthropometry

Fit is a essential requirement for clothing that directly determines the functionality of a cloth. In order to achieve fit, measurements of the wearer's body need to be acquired. Anthropometry is a branch of human science that studies human body size, shape, mobility, flexibility and strength (Gupta & Zakaria 2014). Human body dimension, as ours personality, is largely varied among the population. Many user-centred applications require understanding of the variability of human body dimension. Especially for garment industry, as cloth is an object that its functionality is determined by its coverage and sealability. Both the coverage and the sealability need to be ensured by obtaining wearer's body measurements. This section debriefs the research achievements in anthropometry and introduces several methods for anthropometry data acquisitions.

Human physical stature was the first topic which was studied systematically in anthropometry. Its history can be traced back to 18th century (Tanner 1981). However, the recognition of human body proportion is far earlier than 18th century. In ancient Egypt, a modular grid was often used for the preparation of human figure painting by tomb painters (Pheasant & Haslegrave 2006). This modular grid consists of 18 units from the crown of skull to feet. Figure 2.2 demonstrates three ancient Egyptian figures that have 18:11 relationship between the height of hairline to the navel (Robins & Fowler 1994). The separation of modular grid provides a consistent guideline upon which a figure's proportions could be based on.

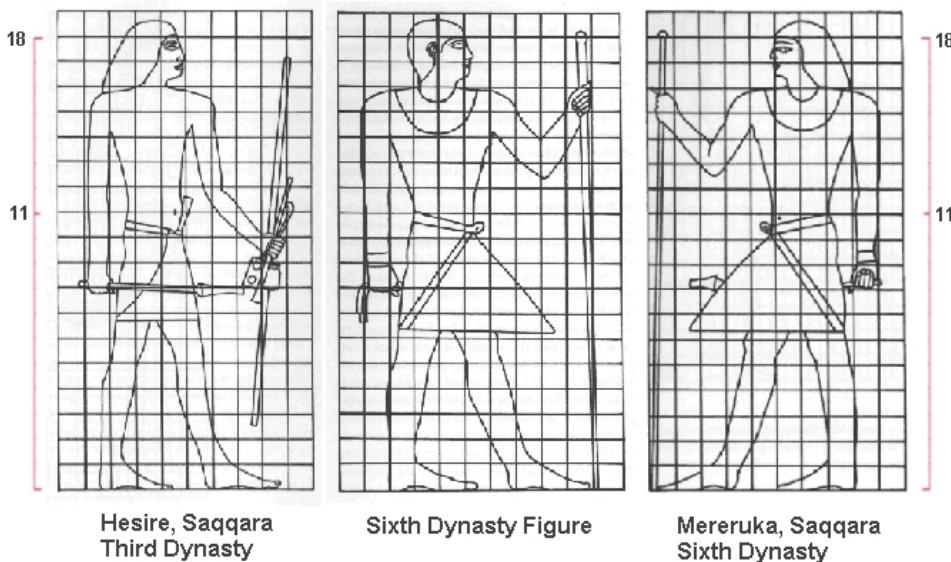


Figure 2.2: *Body proportion in ancient Egypt*

This modular system which was initially used as a drawing standard is still remaining in use today. The most detailed system of human proportions that today's anthropometry researches built on is from the Roman architectural theorist Vitruvius in 15 B.C(Selin 2008). His theory of human proportions is well known as the “well-shaped man” (Arnheim 1955), in which the height of the human stature is equal to the arm span. Vitruvius also employs this human proportion as a fundamental principle in his building design as

he claims that “No temple can have a rational composition without symmetry and proportion, that is, if it has not an exact calculation of members like a well-shaped man” (Pollio et al. 1914; Frings 2002).

The most recognized visualization of Vitruvius’s human proportion is the drawing created by Leonardo da Vinci (Stemp 2006). This piece is accompanied by nodes that are based on the theory of human body proportion developed by Vitruvius. It depicts a male figure in two superimposed postures that the arms and legs of a male person are circumscribed by a circle and a square respectively. After this piece was created, the theory of human proportion had become bound up with the “Golden ratio”. “Golden ratio” refers to the umbilicus divides of the stature of a male person in standing posture in golden section. Such that the ratio of the greater part of the stature to the whole body is equal to that of the lesser part to the greater part (Stemp 2006).

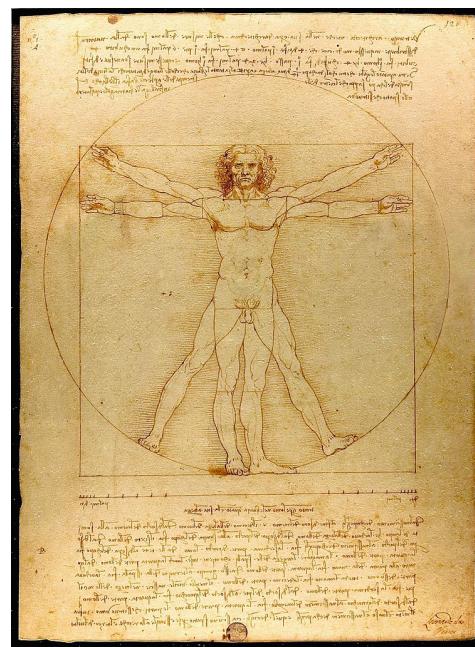


Figure 2.3: *The Vitruvian Man created by Leonardo da Vinci in 1490* (Stemp 2006)

However, the study carried out by Vitruvius was based on Roman pop-

ulation in 15 B.C. In the past two hundred years, anthropometry has shown that span exceeds height in 59-78% of normal adult white men (Schott 1992). Studies on anthropometry was not systematically carried out until 1870's. A Belgian mathematician who named Quetelet published a statistical analysis of the chest sizes of 5000 Scottish soldiers (Quetelet 2011). This was the birth of the science of anthropometry. After that, in a very long time, anthropometric data was mainly used for taxonomic or physiological studies. The development of this science accelerated during WWII which was powered by aircraft industry due to the need of designing better aircraft cockpits.

The measurements involved in modern anthropometry varies from field to field. In general, six measurements are used to describe a human stature, these six measurements are shown in Table 2.1.

Height	Point-to-point vertical measurement
Breadth	Point-to-point horizontal measurement running across the body or segment
Depth	Point-to-point horizontal measurement running fore-and-aft along the body
Curvature	Point-to-point measurement following a contour
Circumference	Closed measurement that follows a body contour
Reach	Point-to-point measurement following the long axis of the arm or leg

Table 2.1: *The basic anthropometry measurements*

In order to extract measurements correctly, subject is required to stay in a predetermined posture. Figure 2.4 demonstrates standard posture for extracting anthropometric measurements.

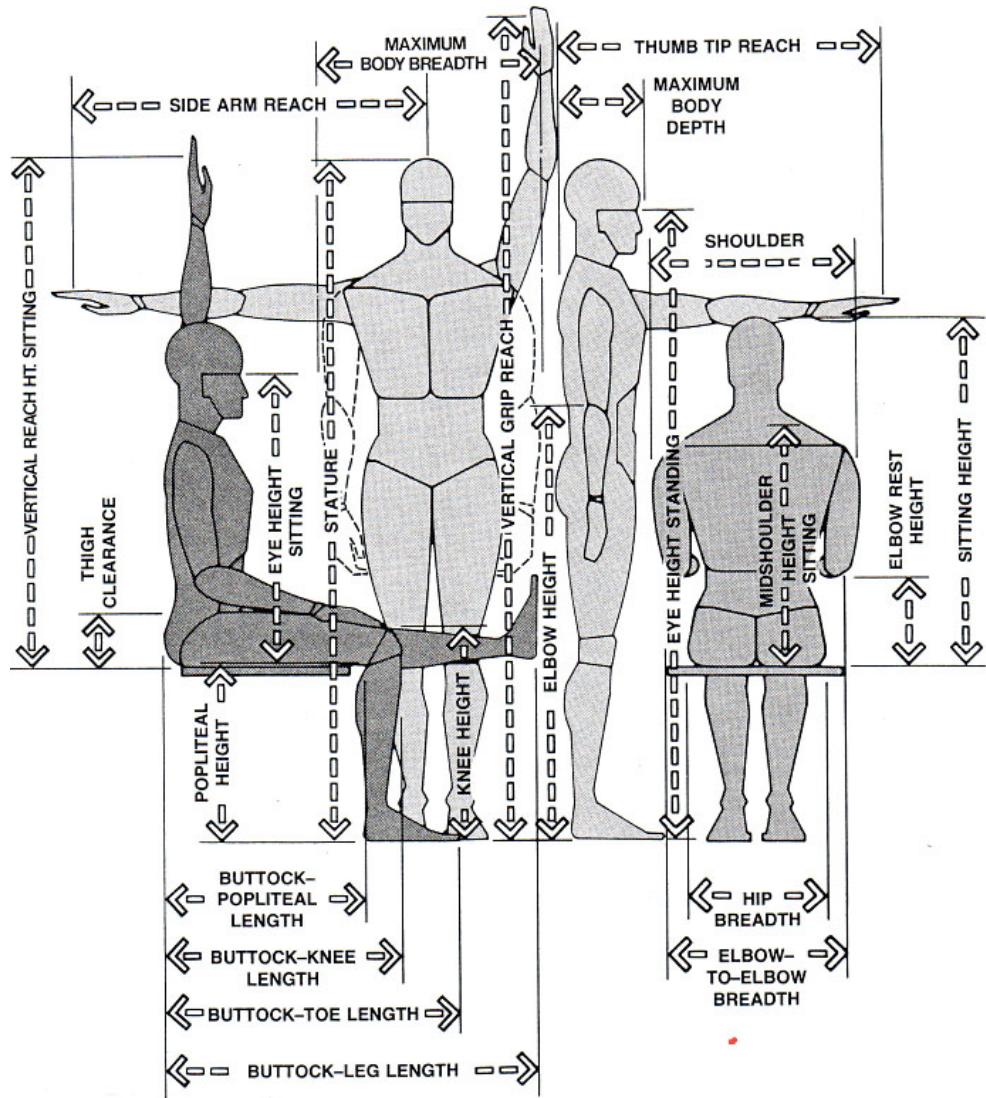


Figure 2.4: Several standard posture for anthropometry measurement extraction (Pheasant & Haslegrave 2006)

Driven by the industrialization of early 19th century, development of standardization reached to an unpretentiously speed. In order to produce cloth for general public in a more cost effective manner, grading system was introduced into fashion industry in order to massively produce cloth in different sizes. Pattern grading is a standard method of applying increases and decreases to cloth patterns to make the cloth larger or smaller in order to produce clothes in a range of predetermined sizes (Schofield & LaBat 2005).

Generally, there are three steps involved in defining a grading system (Schofield & LaBat 2005): Firstly the general population are divide into several categories based on body shapes and proportions with similar characteristics. Then a body measurement is selected as the primary size interval for each category, finally an interval of the primary size is chosen to define the remaining body measurements for each category. Gupta & Zakaria (2014) indicates that, one of the greatest challenges for fashion industry is to produce cloth that fits customer properly. Therefore, for massive cloth production, anthropometry is crucial to the design of a grading system(Norton et al. 1996; Samaras et al. 2007; Mehta 2009).

2.3 Virtual Clothing in Computer Graphics

Virtual clothing is the generic terms of simulating clothing and clothes in a virtual environment. It reproduces both visual features and physical behaviours of textile objects in computer simulated virtual reality (Volino & Thalmann 2000). In general, today's virtual clothing methods can be classified into three types based on the core technique that used for constructing cloth shape or driving its deformation. They are geometrical based virtual clothing methods, physical based virtual clothing methods and hybrid virtual clothing methods.

2.3.1 Geometrical Based Virtual Clothing Methods

Geometrical virtual clothing methods can be traced back to 1986. Weil (1986) presented a method to generate a hanging cloth that utilises geometrical modelling techniques. In this method, cloth was constructed by a grid consists of vertices. Then the shape of cloth was generated from catenary curves between its hanging points, this is demonstrated in Figure 2.5. This technique creates

an underlying shape out of several hanging points. Then it passes through each set of these points and maps a catenary curve to the set. Finally it takes the lowest out of each overlapping set and uses it for rendering. By using this method, the stationary hanging cloth can be generated efficiently, however, limited by the shape of catenary curve, this method can only generates hanging cloth, it cannot generate more complex shape of the cloth.

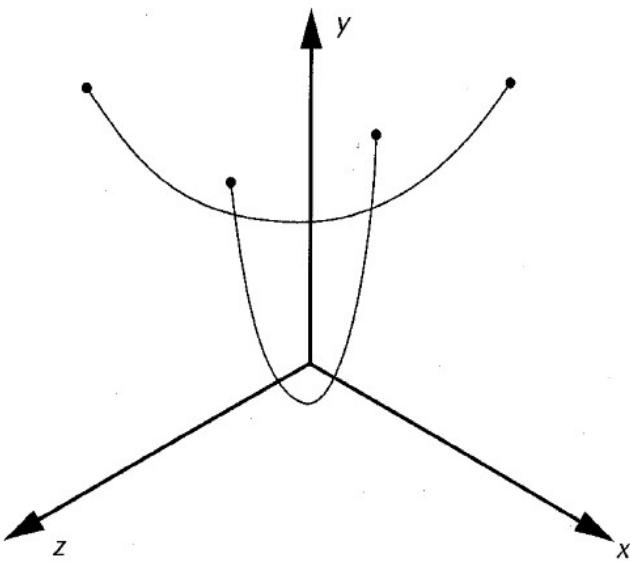


Figure 2.5: Catenary curves for simulating hanging cloth(Weil 1986)

T. Agui & Nakajima (1990) introduced a geometrical method for modelling sleeve on a bent arm. The sleeve is represented by a cylindrical surface consists of a group of circular curves. Wrinkles are created by the consequence of differences in curvature between the inner and outer part of the sleeve. This method only focused upon simulating bent sleeve and can only be implemented in the stationary cloth simulation.

Hinds & McCartney (1990) presented a method for interactive garments designing based on mannequins represented by bicubic B-Spline surface. The garment is represented by a group of 3D panels defined by its edges, and these

are created as a surface offset from the body with various distances. later on, Hinds et al. (1991) presented a method to translate 3D pattern into 2D patterns by using the method presented by Calladine (1986).

Miller et al. (1991) introduced a geometrically deformed model for extracting a topologically closed geometric model from a volume data set. This method introduced a simple geometry as an initial object which is already topologically closed such as a sphere or a cube. Then based on a set of constraints, this simple object is expanded to fit the object within a volume. At the same time, the deformed object maintains its closed and non-self-intersect property. The major advantage of this method is that the sampled data are aggregated by placing geometrical relationship on the model. This allows an initial convex model to be transformed into a concave object. Since the computational cost is associated with the number of constrains that are required during the deformation, the level of detail can be easily controlled. Later on, Thomas Stumpf (2008) extended this method into cloth modelling. According to the experiment reported in this paper, their method reached linear time complexity in terms of number of the mesh vertices. However, since physical property of the cloth is related to the size of the clusters, the topology of cloth mesh can affects the physical property of cloth.

Decaudin et al. (2006) presented a geometrical cloth modelling method which warps developable surfaces around the character body in a natural manner to create visually realistic clothes. By flattening developable surfaces, it can also provides sewing patterns from 3D cloth model. This method utilises a sketch based interface to generate 3D surfaces around character body. Then, seam lines are added manually directly on those 3D surfaces. Each surface panel is enclosed by seams while all the panels are kept assembled along the seam line. Wrinkles are categorized into two types, diamond wrinkle which caused by axial compression and aligned wrinkle which caused by twisting, these two types of wrinkle are demonstrated in Figure 2.6. Wrinkles appeared

on cloth are generated by combining these two types of wrinkle together. Due to the computational complexity of handling developable surface in implicit form, combination of these two types of wrinkle results a limited number of wrinkle variation on a cloth. However, this paper has mentioned that this problem can be solved by adding more wrinkle types, but significant amount of computation is required to achieve such an improvement.

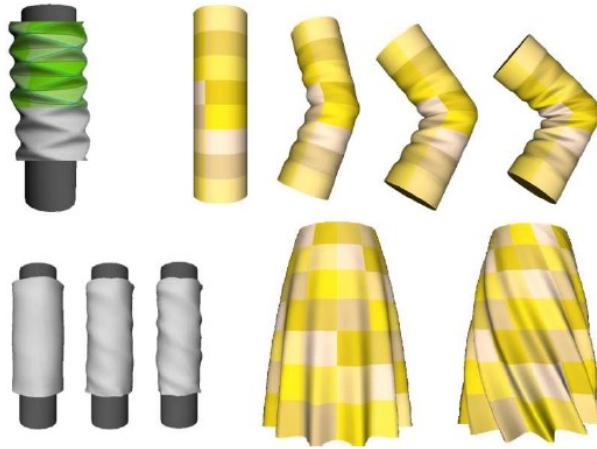


Figure 2.6: Two wrinkle pattern presented in (Decaudin et al. 2006). First row demonstrates the diamond wrinkle pattern and its derivative. The second row is the aligned wrinkle and its derivative.

Wang et al. (2003) introduced a feature based geometrical cloth modelling method for constructing 3D cloth from 2D sketches based on the pre-defined human body features. The human body features are defined based on the profiles of different body parts according to the tailoring rules in fashion industry. For each type of cloth such as suit or skirt, 3D cloth templates are pre-constructed. Then, given a human body, 3D cloth is constructed via a three steps process, firstly, based on those predefined human body features, 3D cloth template is roughly fitted onto a human body. Then according to the 2D sketch which are drawn by user, the profiles of 3D cloth can be specified. Finally, by interpolates the profiles on 3D cloth template, a smooth surface

is constructed via a variational subdivision scheme. For fashion designers, cloth patterns can be extracted by cutting the 3D cloth into parts and flatten them into 2D patterns. However, this method can only models certain type of cloth which shapes follows closely over body profiles. In addition, it can only generate simple cloth mesh which requires further processing to complete a detailed cloth.

Igarashi & Hughes (2002) proposed an interactive method for putting and manipulating clothes on a 3D model. This method constructs 3D cloth from 2D cloth patterns. Patterns are putted on 3D model by establishing proportional correspondence of free-form marks on both 3D character model and 2D patterns drawn by users. Then clothes on 3D character can be manipulated with surface dragging along the body surface. This techniques can only models simple style cloth without foldings such as collar and cloth-cloth collision is ignored due to the computational costs involved in calculating surface constrain of 3D cloth mesh.

Wang et al. (2009) proposed an interactive techniques for designing 3D cloth directly on a 3D mannequin model by using constrained contour curves and style curves. In this paper, contour curves such as silhouette curves and cross section curves are used for defining the general shape of clothes, whilst style curves such as seem lines, notch lines and dart lines are used for generating detailed 2D cloth patterns on cloth surfaces. Contour curves are extracted based on the shapes of human body and the boundaries of sub-meshes of certain style cloth. By editing contour curves, the general shape of cloth can be modified. Style lines are generated by projecting user's strokes onto 3D cloth. Patterns then can be generated by flattening mesh pieces defined by style lines. This method provides a convenient and intuitive approach to design complex cloth on a 3D mannequin. However, editing contour curves and style curves requires knowledge in fashion design and patternmaking which makes it difficult for animation artists.

Meng et al. (2012) introduced a flexible shape control technique for resizing 3D garments automatically while preserve the shape of user-defined features on clothes. 3D clothes are modelled on a reference human body by using any type of cloth modelling techniques. The automatic cloth resizing techniques presented in this paper consists of three steps. Firstly, 3D cloth is transferred from the reference human body to the target human body by using distance constrain between the vertices on cloth and vertices on human body model. Secondly, B-spline curves that are created by user are used to define the shape of features on clothes and a gradient based optimization method is used to match the shape of corresponding feature curve between the cloth on reference human model and the cloth on target human model. Finally, interpolation between feature curves on cloth for target human body are performed to form a smooth shape transaction. Because feature curves are defined by user on each cloth modelled for target human bodies, inconsistent user input can be heavily affected the preservation of cloth features.

Brouet et al. (2012) proposed an automatic method for cloth transferring between characters with different body shapes. This method fits a cloth onto a different character by adjusting vertices on 3D cloth and patterns are extracted after the 3D cloth is fitted to a new character. 3D cloth adjusting process is performed by combining the proportional scaling method with a constrained gradient based optimization process. The constrains are defined by associating the user selected point on the cloth mesh with a pair of points on the relevant bone and character skin. The distances between the selected vertices on cloth mesh and their associated point pairs are used as parameters of the optimization. Optimization is performed by moving each vertex of the cloth mesh on its tangent space to minimize the distance changes. After 3D cloth is fitted onto the new character, the cloth patterns are extracted by using surface flattening method. Because the constrains of optimization is defined by user and the selected reference points are varies on the different character model, the convergence of the optimization can be heavily affected by the

user inputs. This method requires cross-parametrization between source and target models, the fit of the cloth can also be affected by the accuracy of the parametrization. Moreover, extracting the resized 2D cloth patterns requires extra computation and the distortion of the pattern shape can also be introduced by the surface flattening process.

Turquin et al. (2007a) introduced a sketch based interface for modelling 3D cloth on virtual characters. The goal of this paper is to provide an intuitive method to models cloth on a 3D character based on the stocks drawn by users. Users draw sketches to define the shape of cloth, wrinkle details and how the cloth is worn by virtual character. Distances between stocks and mannequin are used to create a distance field which further to be used for cloth mesh generation. This interface is designed to target to non-experienced users. However, because the information from user input stroke cannot define complex cloth structure, it can only models simple style single layer clothes.

Umetani et al. (2011) proposed an interactive tool for cloth design that enables bidirectional editing between 2D patterns and 3D cloth. In order to maintain synchronization between 2D and 3D prospective, the topology of 2D cloth pattern and its corresponding 3D cloth piece is identical and updated simultaneously by user input through a real-time physics-based simulation method which allows real-time feedback in response to changes in 2D cloth patterns. However in order to maintain real-time responsiveness, cloth self-collision is ignored. Moreover this method can only synchronize 2D pattern with a 3D cloth on a static 3D mannequin it cannot simulating cloth dynamic behaviour subject to motion of 3D character.

Yu et al. (2012) proposed a sketch based interactive cloth modelling approach for dressing 3D virtual character. 3D clothes are modelled by sketching cloth contours around the character body. Key body features such as extreme points or girth of human body are used as constraints during the construction of clothes. This method provides an intuit way to model 3D clothes

for artist. However, the key body features are defined based on shape similarity of human body, therefore, it is not suitable for modelling outfits for animation characters due to their various body shapes and proportions.

Geometrical based virtual clothing methods provide an effective approach to construct and edit 3D cloth shapes. However, cloth is a soft object which its bending resistance is much lower than its stretch and compression resistance, many of its material features can only be reproduced when the external forces are applied. Geometrical based virtual clothing methods are able to drive the deformation of cloth, however, due to the computational complexity of mimicking detailed dynamic behaviours of cloth, physical models are usually utilised to produce its dynamic behaviours.

2.3.2 Physical Based Virtual Clothing Methods

For physical based virtual clothing methods, cloth is usually represented by a set of particles that are inter-connected to each other by springs. This representation replicates the behaviour of soft flexible object such as fabric. In general, there are two types of model involved in physical based cloth modelling method, energy-based method(Terzopoulos et al. 1987) and the force-based method(Volino & Magnenat-Thalmann 2005). Energy-based model calculates the total energy of cloth by using a set of energy equations. Those equations determine the shape of cloth by moving the particles in order to achieve a minimum energy state. This type of method is widely used in static simulations. Because energy model is based on the kinematic theory in which the shapes are compositions of geometrically or algebraically defined primitives. the Force-based techniques use differential equations to represent the force among each particle and perform a numerical integration to solve differential equation to locate the position of each particle at every time step. This type of methods is normally used in dynamic cloth simulation.

Terzopoulos et al. (1987) introduced a method utilises physically-based model to construct the shape of a cloth object. Their method uses elasticity theory to describe the shape and motion of the deformable materials. It also has the ability to interact with other physically-based models. The simulation model introduced in this paper is based on the simplifications of elasticity theory to deformable curves, surfaces, and solid objects. It has the ability to generate static shapes by simulating their physical properties such as tension and rigidity. Moreover, by bringing the physical properties such as mass and damping in to the physical simulation, the dynamic behaviour of objects can be simulated. “*The simulation involves numerically solving the partial differential equation, which govern the evolving shape of the deformable object and its motion through space.*” (Terzopoulos et al. 1987). However, it is very time consuming to solve the energy equation for a complex object, limited by the computational power, this method can only simulate the interaction between simple shaped objects.

Volino & Magnenat-Thalmann (2005) introduced a general mechanical model for cloth simulation. Mass-spring system cannot represents the anisotropic nonlinear mechanical behaviour which often appears on fabric. This model utilises an accurate particle system for dynamic simulation instead of mass-spring system. A triangle face of cloth mesh is described by three 2D coordinates correspond to three mechanical properties, weft, warp elongation and shear. These coordinates describes the location of each triangle’s vertices on the weft-warp coordinate system that defined by the directions U and V with an arbitrary origin. This is shown in Figure 2.7. This particle system is able to simulate the anisotropic nonlinear mechanical behaviour of a cloth object by using polynomial spline approximations of strain-stress curves. Furthermore, the aforementioned three mechanical properties are physically measured from real cloth piece using Kawabata Evaluation System (Kawabata et al. 1999) and SiroFAST method (De Boos et al. 1997).

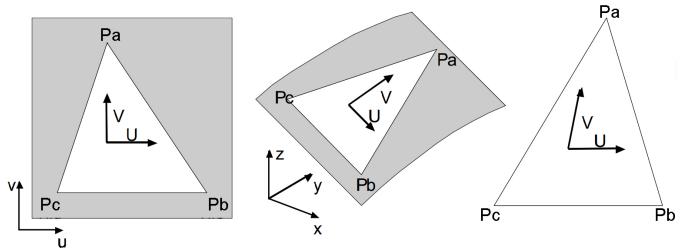


Figure 2.7: A triangle face of cloth surface defined on 2D cloth surface(left). The deformed triangle in 3D space(middle). The deformation of weft-warp coordinate system of this triangle face(right).

Volino et al. (2009) proposed a simulation model for large deformations of textile. It can generates the nonlinear tensile behaviour of textile with accuracy and robustness and the process of simulation remains simple. Differing from the majority of existing cloth simulation systems, this model uses accurate strain-stress curves to represent weft, warp and shear tensile behaviour. This model also includes elasticity and viscosity which makes it suitable for dynamic motion simulation. Most importantly, the computation involved in this method is not more expensive than mass-spring model but offers a significantly more accurate results.

Chen & Tang (2010) introduced a method for simulating inextensible cloth in a collision-free condition subjected to a conservative force (e.g., the gravity). This paper points out that for a piece of cloth, its stretch resistance and compression resistance are many times larger than its bending resistance. Therefore, the simulated cloth can be considered as an inextensible object. However, traditional physically-based cloth algorithm cannot preform correctly if some of the stiffness coefficients are set close to infinite. Thus they provide a method such that the physical-based simulation process is transferred into a deformation process of an initial developable mesh surface to a final mesh surface. Their method deforms cloth mesh by using energy minimization. Gauss-Newton iteration is used for the minimization of energy

function. However, this method can only handles conservative force, it cannot handles non-conservative force such as friction.

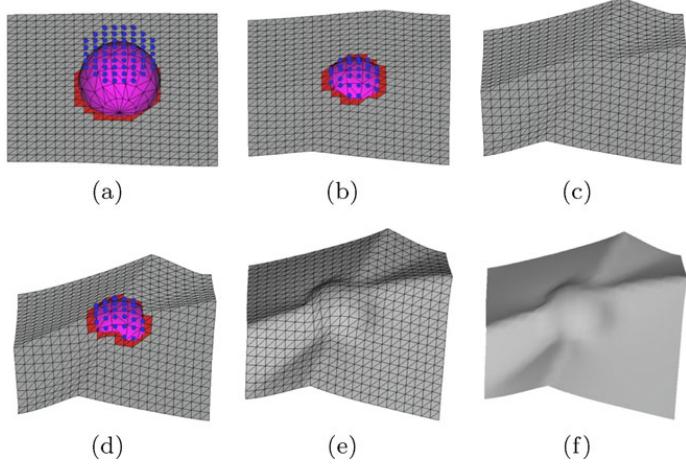


Figure 2.8: blue points are the dynamic anchor points, (a), (b), and (d) are three intermediate states in sequence. The number of the dynamic anchor points are reduced gradually; (c) and (e) are the two sequent intermediate states where the potential energy of the mesh is lowered gradually. (f) is the final state where the potential energy reach to the minimum (Müller & Chentanez 2010)

Physical based virtual clothing method calculates inter-force among each vertex. The number of vertices on cloth mesh directly influence the detail and performance of the simulation. Therefore, in order to produce fine detail results, the resolution of the cloth mesh must stays high which leads to a very heavy computation. Moreover, the parameters of differential equation used for representing the cloth object is difficult to obtain. Thus it has been mainly used in off-line high accuracy simulation such as garment industry and off-line film production. In game industry, limited by the heavy computation of physical based virtual clothing methods, the detail of cloth is largely constrained.

2.3.3 Hybrid Virtual Clothing Methods

Hybrid virtual clothing method combines both physical simulation method and other types of deformation methods together such as data-driven method or geometrical method in order to compensate the shortcomings of each technique to produce a fine detailed result while minimizing the computation.

In order to reduce the computational complexity of physical cloth simulation methods, Rudomin (1990) presented a method which uses geometrical approximation to reduce the computation time of physical based virtual clothing method introduced by Terzopoulos et al. (1987). Later on, Kunii & Gotoda (1990); Tsopelas (1991) proposed a series of hybrid virtual clothing methods that map geometrically modelled fine wrinkle details onto a physical simulated cloth mesh in order to avoid heavy computation for simulating high resolution mesh. Hadap et al. (1999) introduced a texture based wrinkle modelling method for cloth simulation. This method generates deformation detail based on the bump map that is created by user on a physical simulated coarse cloth object. However, because there can be a limited number of pre-defined shape details or bump maps created by user, the variety of the deformation is consequently limited.

Data-driven method is widely used in combination with physical based simulation method to improve the efficiency of cloth simulation. Cutler et al. (2005) introduced a kinematic method for generating wrinkles on cloth for CG characters. Winkles are created by artists using geometry sculpture software and are stored into a database. The final cloth deformation details are generated by matching the similar stretch distribution of current cloth to the referencing posture of a character. However, the similarity between reference posture and coarse physical simulated cloth only exists on the tight fit cloth, the wrinkle database can not be used to generate shape detail on the loose fit cloth.

Popa et al. (2009) introduced a data-driven method for generating fine detailed folds on captured cloth model. In this method, the shape and the position of wrinkles are captured from video footage based on the wrinkle's distinguishing shape characteristics, and wrinkles are created by using stretch-minimizing deformation, which produces believable wrinkle shapes. However, the details of captured wrinkles are limited by the resolution of cloth mesh therefore, significant amount of detail can be lost during the capturing process.

Feng et al. (2010) introduced a hybrid method which can provides high-quality dynamic folds and wrinkle for cloth simulation while still maintaining real-time ability. This method captures the relationship between two different resolutions of mesh by using data-driven model. Then this relationship is used to transforming lower quality deformation on lower resolution mesh through a mid-scale deformation transformation onto the higher resolution mesh with fine detailed wrinkles added on. In animation production pipeline which they developed, during each time step, it starts with the physical simulation for low-resolution mesh. After that, the deformation transformation and collision detection is executed. In this step, two stages of nonlinear mapping are used to generate the deformation of high-resolution mesh. At the first stage, the deformation for mesh with the lowest resolution is mapped onto a mid-resolution object which uses proxy bone to represents a local region on the high-resolution cloth mesh, and then the deformation of proxy bones are mapped onto the high resolution to create final deformation. Collision detection is also a critical and time-consuming process during cloth simulation. In this paper, they proposed a method of using proxy bones originally for the mid-scale cloth deformation to represents high-resolution cloth object to interact with character model. The bounding volume is attached to every proxy bone which represents a small area on cloth object. When character moves, instead of calculating the collision between each mesh triangles of the cloth and body model, collisions between each proxy bone are evaluated. Although

the accuracy of collision handing is slightly reduced, the computation time for collision test are reduced significantly. Although the efficiency has been largely improved, like other data-driven method, it requires a time consuming pre-simulation to acquire training data.

The hybrid method combines different modelling and simulation techniques into one package to solve the problems of virtual cloth construction. In general, other type of virtual clothing methods are used to overcome the shortcomings of the expensive computation of physical virtual clothing method in order to provide a more effective and efficient solution. Because other type of techniques such as geometrical cloth modelling method or data-driven cloth simulation method usually compromises the physical simulation accuracy, it is often used in the area which focuses on visual realism more than physical accuracy, such as computer animation.

2.4 Geodesics

Human-being, as a species, shares similar body structure and proportions. Based on human body sizes and proportions, the population can be categorized in to several groups where within each group, individuals share similar body characteristic. This similarity enables fashion designers to design a cloth based on one standard body template and then convert into different sizes. It also builds up the foundation of cloth sizing chart in fashion industry. Known what sizing category we belong to can help us find cloth that fits us in local cloth shops conveniently.

However, animation characters do not have this luxury. Because they are the products of creative imagination of animation artists. Therefore, the body shapes and proportions of animation characters vary greatly and they can be largely different from a real human. Moreover, for animation characters, the standard posture used for modelling differs from person to person. In

order to model clothes that fit animation characters, the measurements need to be extract first. Due to the inconsistent posture of characters, the measuring method used in anthropomorphic data acquisition can not be used.

Figure 2.9 illustrates the character design in file “Ratatouille”.

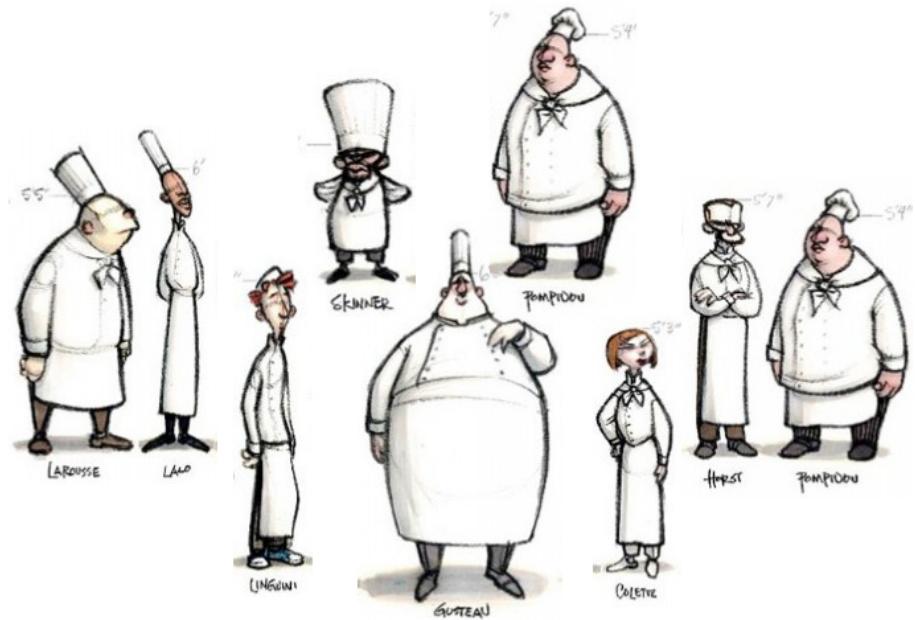


Figure 2.9: Character designs in film “Ratatouille” (Brad Bird 2007).

In real life, when tailoring clothes, few measurement data need to be extracted from customers. Tape ruler is usually used as the major measuring tool during this process. During measuring, tailor holds two ends of a tape ruler and place them onto a body part whilst applying tension to the ruler to force it follows the profile of the body part where its length or girth is the desired measurement. Figure 2.10 illustrates the process of taking measurements in Italian fashion industry(Browne 2011).

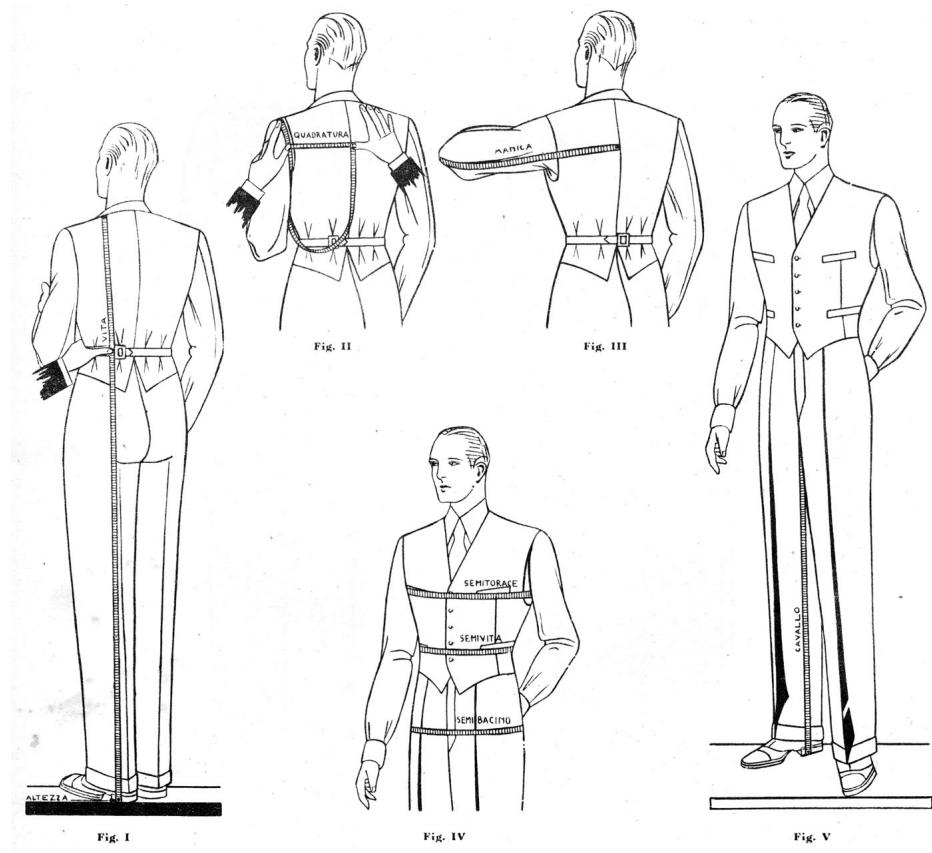


Figure 2.10: Taking of measurements in fashion industry (Brad Bird 2007).

For tape measuring method, locations which are used to apply two ends of a tape ruler need to be defined, the posture of measured subject is not strictly required. Therefore, tape measuring method can cope with small different postures of subjects. Tape ruler is a soft flexible strap, when tension is applied, based on Hooke's law, the tape ruler stays at a minimum energy states and the ruler, which follows the profile of customer body is at its shortest.

In order to produce made-to-measure clothes for animation characters, the body measurements of characters need to be extracted first. Geodesic path is the shortest path between two points on a curved surface. In order word, on a character model, a geodesic path between two points shares similar features as a tape ruler when it is placed between same two points and tension is ap-

plied to it. Based on this observation, this thesis utilises geodesic to mimic the tape measuring process in tailoring for length measurements extraction from different characters in different postures. Geodesic computation is a common operation in computer aided design, machine learning, medical image analysis and computer animation. Throughout years of research, many algorithms have been developed for various applications, such as surface-based brain flattening (Bartesaghi & Sapiro 2001; Wandell et al. 2000), mesh refining (Peyré & Cohen 2006a), mesh segmentation (Katz & Tal 2003) and terrain navigation (Aleksandrov et al. 2005).

Calculating geodesic usually involves traversing every node in a graph in order to find shortest path between source and destination, therefore, reducing computational complexity is a main goal for geodesic related researches. Dijkstra (1959) was the first to address “single source all destination shortest path” problem on a directed non-negative weighted graph. Early works of geodesic computation mainly focused on convex polyhedrons. Sharir & Schorr (1984) extend Dijkstra’s(Dijkstra 1959) algorithm into three dimension space, with a time complexity $O(n^3\log n)$. Later on, an exact solution of geodesics on a convex polyhedral surface was developed by Schreiber & Sharir (2006), with a reduced complexity of $O(n\log n)$.

For non-convex polyhedrons, a challenging issue is that, a geodesic path may go through hyperbolic vertices on a polyhedron. Based on the definition of geodesic proposed by Mitchell et al. (1987a), a geodesic path on a planner forms a straight line. When the faces surround to a hyperbolic vertex is flattened, one edge will appear twice in two different edge sequences since these two edge sequences are all locally shortest around this hyperbolic vertex but not necessarily shortest in other local area, the computational result may not be correct.

In order to handle this problem, Mitchell et al. (1987b) (MMP algorithm) extended Dijkstras algorithm (Dijkstra 1959) and introduced the tech-

nique of “continuous Dijkstra” which propagates distance information from source vertex outward in a Dijkstra-like manner. The “continuous Dijkstra” propagation performs as follows, each edge is partitioned into several sections called “window”, each window carries distance information through out the propagation. MMP algorithm costs $O(n^2 \log n)$ time to solve the “Single Source All Destinations” problem. Surazhsky et al. (2005) implemented MMP algorithm and furthermore, extend the original MMP algorithm (Mitchell et al. 1987b) to a bounded error approximation algorithm noted as MMP approximate algorithm. The time complexity of the MMP approximate algorithm has been improved to $O(n(\log n))$ (Surazhsky et al. 2005).

Another popular approximation algorithm is the fast marching method (FMM) (Kimmel & Sethian 1998). This method involves solving a discrete version of the Eikonal equation over a regular grid, with the cost of $O(n \log n)$. However, the uniformity of the grid highly affects the approximation error. Therefore, performing FMM on an irregular and skinny triangulated grid will leads to significant high error. Bose et al. (2011) points out that the approximation error of FMM is actually unbounded.

Chen & Han (1990)(CH algorithm) provided an exact solution to geodesic problem based on a key observation of “one angle one split” principle with time complexity of $O(n^2)$. This method consists of two parts. In the first part, the shortest path from a given source to each vertex on mesh is computed and a set of windows is defined on each edge. Each window contains informations about the shortest path from the given source point to points on the current window. In the second part, windows are used to compute decomposition of the surface and the shortest path to any destination point on polyhedral surface can be calculated. However special case needs to be dealt within the circumstance of a geodesic passes through a hyperbolic vertex because the planer unfolding to this type of vertex can be self-overlapped.

Xin & Wang (2009)(Improved CH algorithm) discovers that in CH al-

gorithm, many windows are unnecessary for the propagation. Therefore, their method improved the efficiency of algorithm by introducing less windows during propagation. Although the asymptotic time complexity is $O(n^2(\log))$, according to the numerical experiments, the Improved CH algorithm show great advantage over CH algorithm and MMP algorithm in terms of computational time costs.

Calculating geodesics is a very time consuming process, the reviewed researches are mainly focus at improving the efficiency of the geodesic computation while maintaining the computational accuracy. However, because high resolution character model has become more and more popular in computer animation, the current algorithms still consume large amount time to calculate geodesics on a high resolution model(MMP and ICH requires more than one hour to calculate geodesics on a 800k vertices polygon model). Moreover, although technique such as fast marching method(Kimmel & Sethian 1998) is proposed to reduce the time complexity of the computation, its accuracy varies among the different character models with different topologies.

2.5 Genetic Algorithm

Modelling cloth from cloth patterns for a character requires each pattern to be adjusted individually. During pattern adjustment, several criteria need to be satisfied in order to fit a cloth to a character and preserve the design of cloth. In this thesis, pattern adjustment is considered as a multiple objectives optimization process. For optimization algorithms, genetic algorithm advances itself by its simplicity and efficiency.

In nature, individuals that suit the environment best survive. The ability to adapt to the changing environment is crucial for the survival of individuals. Genetic algorithm are inspired by such mechanisms of evolution and natural selection in real world (Srinivas & Patnaik 1994b). The pioneering

book of Holland (1992) demonstrates wide range of problems that can be solved by using evolutionary process in a highly parallel manner. The process which utilises the concept of evolution is called genetic algorithm (Koza 1995). Nowadays, after decades of research, genetic algorithm has become a practical and robust optimization method that can solve wide range of problems in real world.

In general, the genetic algorithm transforms a set of individuals named population into a new generation by using Darwin's principle of reproduction and survival of the fittest (Holland 1992). Each individual in the population represents a possible solution of the given problem and every individual is associated with an fitness value that evaluates its performance to the given problem. During the process of transformation, genetic operations such as crossover(mate), mutation and selection are performed in order to explore new area on the cost surface. For every individual that is newly generated, a fitness value is assigned to it. This entire process is performed iteratively till the best solution is found or the evolution limitation has been reached.

There are two types of optimization based on the number of objectives of the given problem, single-objective optimization and multi-objective optimization. For single-objective optimization, finding a single best solution is the purpose of the optimization. This method has been studied intensively for decades (Srinivas & Patnaik 1994b; Koza 1995). However, many real-world problems can not be described by a single objective, moreover, in many cases, it requires simultaneous optimization for multiple competing or conflicting objectives. Therefore, for multi-objective optimization, the best solution usually does not exist, instead, a group of "good" solutions are generated to form "pareto front" (Pareto 1906). Multi-objective genetic algorithm is designed specifically for such type of problems.

Schaffer (1985) introduced vector evaluated genetic algorithm(VEGA) for solving multi-objective optimization problem. This algorithm is an ex-

tension of simple genetic algorithm(SGA) (Schaffer 1985). This algorithm modifies the selection operator of SGA so that at every generation, a sub-population are selected based on each objective in turn. Let K denotes the number of individuals in each sub-population and N be the number of individuals in whole population. Within each sub-population, N/K number of individuals are selected and shuffled together to form a new generation. Due to the linear combination of all objectives during the evolution process, this algorithm cannot form pareto front on concave cost surface.

Ishibuchi & Murata (1996) proposed a weighted-sum approach for multi-objective optimization. The randomly generated weight is assigned to each objective for each individual. Crossover operator selects individuals from dominated set according to the weights of individual to generate offspring. This method provides an approach that is highly effective for generating a strongly non-dominated solution. However, the setting for weights can be difficult because each weight directly determines the scale of the objective. Inappropriate setting on weights can leads to failure of the formation of pareto front. Moreover, this method cannot generates pareto optimal solutions for non-convex searching space.

Horn et al. (1994, 1993) proposed a method that based on the pareto domination tournament selection and equivalence class sharing(NPGA) (Horn et al. 1994). During selection, a random number of individuals are selected into a comparison set, then two random individuals are compared against the comparison set for domination, If one is dominated by the comparison set and the other is not, the dominating individual is selected for crossover. If neither or both individuals are dominated by the comparison set, the equivalence class sharing is performed. During this process, the individual that has the least number of neighbours with in the niche radius (Shir & Bäck 2006) is considered “better fits”. This method only performs pareto selection to a part of the population, therefore, high efficiency can be achieved and pareto front

can be well kept throughout large amount of generations. However, because the selection is only performed on a part of the population, the quality of the convergence is highly depended on the choice of niche radius value and the number of individuals in the comparison set.

Zitzler (1999); Zitzler & Thiele (1999) proposed a method(SPEA) that combines the concept of elitism and non domination. For each generation, an external population which consists of a set of non-dominated individuals that are selected since the initial population is kept. The number of the dominated solutions determines the fitness value for each individual of current population. In general, if an individual dominates more in terms of objectives over other individuals, the higher fitness value will be assigned to the dominating individual. The external population is then updated after fitness value has been assigned to entire population. If an individual is not dominated by both current population and external population, it is added into the external population and all the individuals that are dominated by the added one are removed from external population. However, because individuals that are dominated by same external population have identical fitness values. When external population only has one individual, this algorithm becomes a random search algorithm.

Knowles & Corne (1999) introduced the pareto archived evolution strategy(PAES). This method starts with a random solution of a multi-objective optimization problem. Then, this solution is mutated by a normally distributed probability function that has zero mean and constant mutation strength. The generated offspring is compared with the original solution and the one with better fitness value is used for mutation of next generation. A set of solution that contains good solution for each mutation are kept. Every time when mutation is performed, offspring are compared against its parent and all the other solutions in this set for dominations. The PAES has provided a direct method for controlling the diversity of pareto optimal, therefore, the prema-

ture convergence is less likely to occur and a higher probability to achieve real pareto front.

Deb (1999); Srinivas & Deb (1994); Deb (2001) introduced non-dominated sorting genetic algorithm(NSGA) based on the work of Goldberg & Deb (1991) which is able to handles any number of objectives. This method ranks individuals based on the non-domination level and the fitness of each individual is determined by the non-domination level of itself. This method can maintain the stability and uniformity of the non-dominated individuals throughout the reproduction. Later, Deb et al. (2002) improved efficiency of this method and achieved a fast and elitist multi-objective evolutionary algorithm based on non-domination sorting method. This improved version is named as NSGA-II. By applying crowding distance into the sorting method, this improved version shows great advantage in terms of efficiency to either SPEA or PAES.

NSGA-II(Deb et al. 2002) is one of the most effective and efficient multi-objective optimization method that has been widely used in many fields. Although it can handles any number of the objectives, recent research(Ishibuchi et al. 2008) indicates that, when the number of the objectives increases, more individuals have become non-dominated, therefore more evolutionary iteration is needed to achieve convergence. In face, based on the experiments, NSGA-II performs best when the number of the objectives is around three(Ishibuchi et al. 2008). This thesis defines three objectives for the cloth pattern optimization, therefore, NSGA-II outstands itself in the pattern adjusting problem in this thesis.

2.6 Summary

In this chapter, a brief history cloth making and anthropometry are presented first, then three types of virtual clothing methods in computer graphic has

been reviewed.

The geometrical virtual clothing method generates cloth for virtual character by using geometric modelling method, the physical properties of the cloth object is not in its consideration. Therefore, it is used as a modelling tool for generating static cloth geometry and it needs a considerable amount of manual inputs to create a cloth.

The physical virtual clothing method describes cloth by using dynamic model. The shape of cloth is determined by forces or energies of each vertex on cloth mesh. This method not only be able to model static shapes of cloth but also be able to simulate its dynamic behaviours. Physical virtual clothing method has been widely used in fashion industry and animation production. However, because the level of details is determined by the number of basic element in dynamic model used for representing cloth, the only way to increase the level of detail is to increase the resolution of the mesh or particles. This increases the computational cost.

In the area where visual appearance is more important than physical simulation accuracy such as films or games, hybrid virtual clothing method is applied in order to improve the efficiency of virtual clothing while maintaining high visual realism. Hybrid virtual clothing methods combine different types of modelling or simulation method together in order to compensate the shortcomings of each method to be applied on its own and provides a highly efficiency approach for its application. However, hybrid method also has its own drawbacks, because it uses some techniques such as data-driven deformation method to simulate wrinkles instead of producing them by solving physical equation. The variety of fine detailed wrinkle is often limited by the non-physical deformation method.

The methods reviewed in this chapter are mainly focus at reproducing either static shape or dynamic behaviour of cloth for a particular character. Only a few of them addressed problem of modelling cloth for different char-

acter with different body shapes and proportions. In order to transfer a cloth from one character to another, large amount of repetitive manual operation is still required. This duplication of effort can hardly be eliminated even when the same cloth design is fit to different characters. Most of current methods tie the character with their cloth together, and they can not be separated. Especially in today's virtual character applications, the number of the different characters that required in the virtual environment has been increased rapidly. The low efficiency caused by the duplication of effort in current virtual clothing methods has become a significant obstacle. Therefore, dressing different characters efficiently is still considered as a challenging task.

In order to model cloth from cloth patterns, measurements of the character model is crucial for determining the size of each pattern. This thesis utilises geodesic path to measure the length of desired body parts in a similar manner to tape measuring process in fashion industry. Moreover, by using geodesic distance instead of euclidean distance also results less variation for different character postures. However, computing geodesic by current geodesic algorithms on a high resolution model is still a very time consuming process and can heavily affects the flow of animation production pipeline. In order to improve the efficiency of geodesic computation whilst maintaining high accuracy, two geodesic algorithms are proposed in this thesis respectively for accurate and approximate geodesic computation on triangulated manifolds. The most important contribution is that the proposed approximation algorithm can reach linear time complexity with a bounded error on triangulated manifolds. The time consumed for solving the geodesic path between multiple pairs of source and destination has been largely reduced, at the meantime the accuracy of the solution is maintained.

In the real world, in order to produce fit cloth, the size and shape of each cloth pattern are adjusted based on the measurements of the customer. This process requires deep knowledge in tailoring. Current pattern based

cloth modelling method models cloth by mimic this process in a computer simulated environment. In practice, adjusting cloth patterns for a particular character still requires many manual operation. Because few animation artists possess such deep knowledge in tailoring, therefore, this method is rarely used in the production of film and games. The automatic cloth pattern adjustment method presented in this thesis considers the process of adjusting cloth patterns as an optimization problem. There are three major objectives in the optimization that need to be considered, the measurements of the character, the integrity of the seam-line on each pattern and the preservation of the shape of each pattern.

For current gradient based optimization methods, preventing algorithms from converging to local minima is always a difficult problem that many researchers are working on. Among these optimization methods, genetic algorithm outstands itself by its simplicity and efficiency. By using such an approach, the cloth pattern adjusting method presented in this thesis can automatically generates the best combination of the shapes and sizes of each pattern. Most importantly, By automating the process of pattern adjustment, the tailoring knowledge that required by current pattern based cloth modelling method is on longer needed. Therefore, animation artists can model varies clothes based on the large amount of existing cloth design patterns in today's fashion industry. Moreover, by automating the process of creating fit cloth for a character, the efficiency of generating cloth for different characters with different body shapes and proportions can be largely improved.

Chapter 3

Character Measurements Extraction

In cloth production, customer needs to be measured first in order to determine the size of cloth. In anthropometric data acquisition, two types of measurements are associated with cloth making, length measurements and circumference(girth) measurements. Both measurements are obtained by calculating the length of tape ruler between two datum points on skin or surrounding a body part. Table 3.1 lists out several major measurements that are associated with cloth making.

	Name	Measuring Method
Circumference	Bust Girth	The horizontal girth around the bust point
	Chest Girth	The horizontal girth passed over the shoulder blades, under the axillae, and across the chest
	Waist Girth	The horizontal girth go through front waist point and back waist point
	Middle Hip Girth	The horizontal girth around the abdomen girth point
	Hip Girth	The horizontal girth around the hip point
	Neck girth	The horizontal girth go through the neck shoulder point
	Cuff Girth	The girth around the wrist point
Length	Height	The distance from the back neck point to the heel point
	Back Length	The distance from the back neck point to the back waist point
	Sleeve Length	The distance from the neck point to the wrist point
	Arm Hole Length	The distance from the front axilla point go through the shoulder point to the back axilla point
	Sleeve Top	The shortest distance from the shoulder point to the line which go through two axilla point on the flattened sleeve pattern
	Waist length	The distance between the waist line and the hip line
	Crotch Depth	The distance from the centre of the front waist line to through crotch point to the centre of the back waist line
	Inside-Leg Length	The distance from the crotch point to the inside ankle point

Table 3.1: *The definitions of the measurements and their associated datum points (Armstrong 2000; EN:13402 2001)*

In order to extract measurements listed in Table 3.1, datum points need to be defined on character body. Figure 3.1 illustrates the datum points for general measuring on a female figure based on the tailoring rule (Xiong 2008).

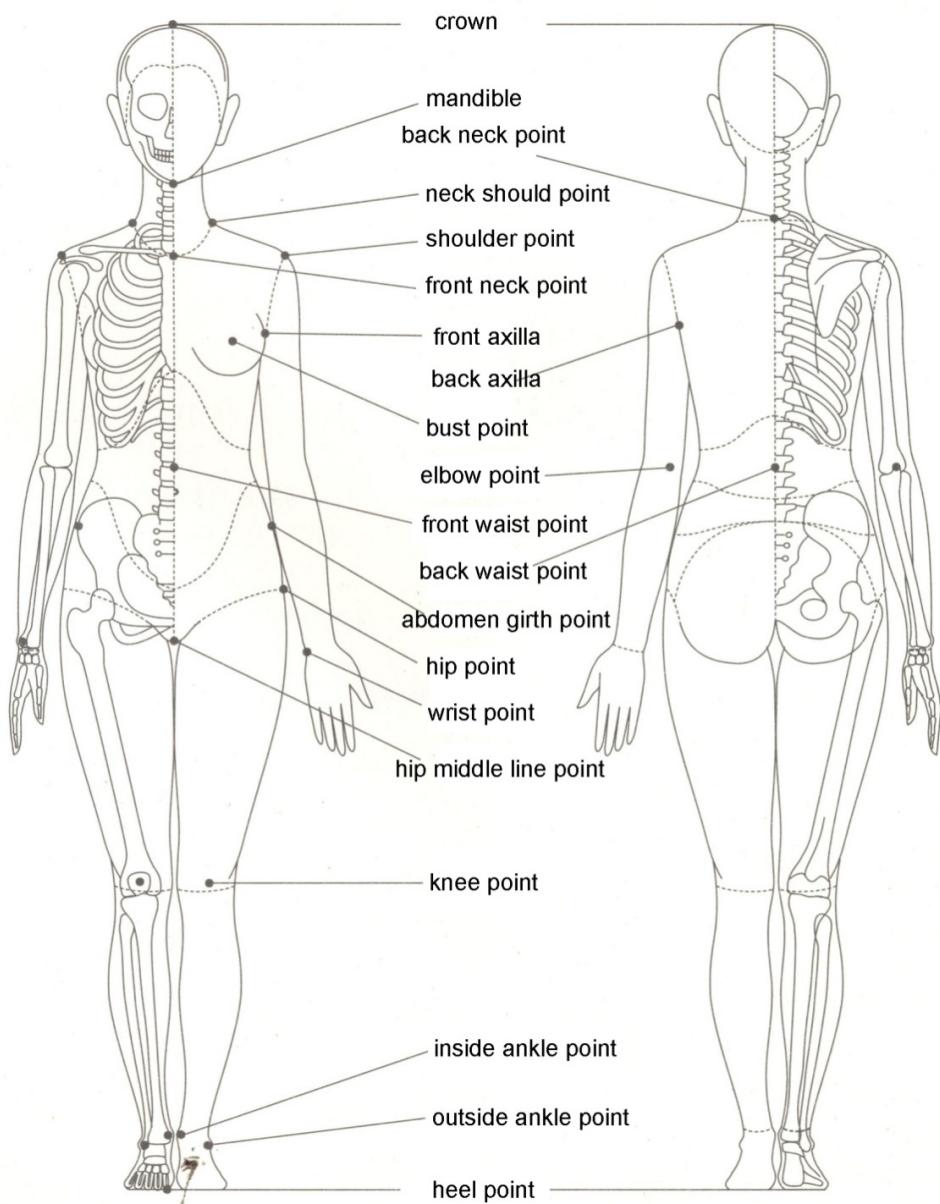


Figure 3.1: Datum points on a female body(Xiong 2008)

When extracting circumference from a subject, tape ruler is applied to the cross-section of a body part. After tension is applied to the ruler, it forms a convex hull of the cross-section of that body part. Therefore, in this thesis, the circumference is measured by applying convex hull algorithm(Graham 1972; De Berg et al. 2008) to the cross-section of a body part, the arc length of the convex hull is used as circumference of the body part. Figure 3.2 demonstrates two circumference measurements on a character.

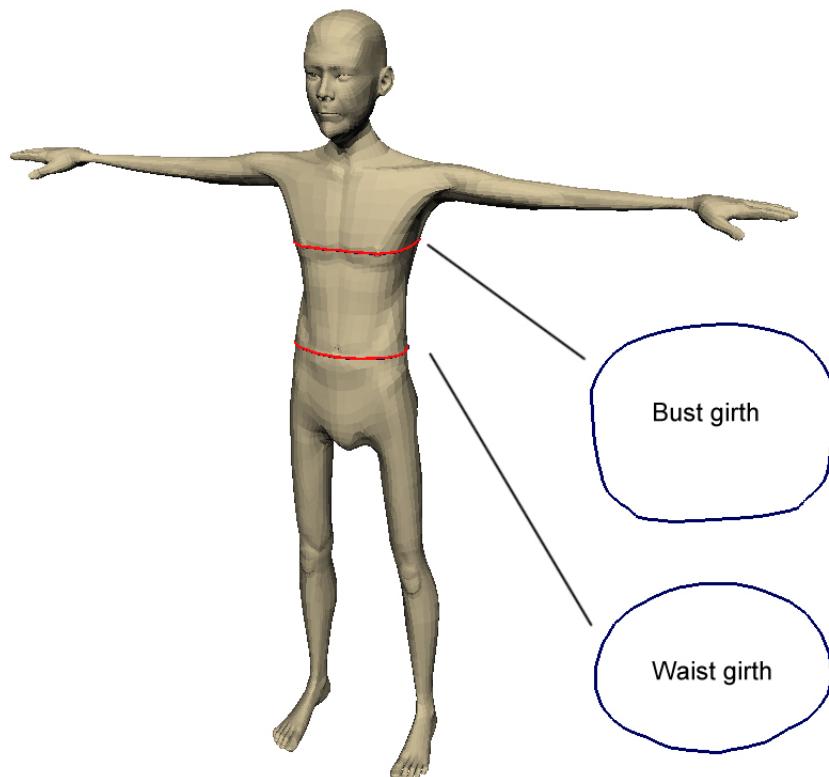


Figure 3.2: Two circumference measurements on character, two blue closed curves indicate the convex hull of its corresponding measurements

For extracting length measurements, subjects are usually required to stay in a predefined posture through out the measuring process. For example, when measuring arm length, subjects are required to stretch straight their arm in order to apply ruler from shoulder point to wrist point. The correctness of

the posture significantly affects the accuracy of measurements. When arm is bent, the distance between shoulder point and wrist point will be shortened. In computer animation, “T-Pose” is the standard posture for character modelling because the body and limbs are stretched straight so the space between different body parts is maximized. This provides many convenience for rigging or texturing. Unfortunately, the concept of “T-pose” are ambiguous because different studios have different definition of “T-Pose”.

The changes of posture might need different measuring method to cope with, for example, when measuring a character in a bent arm posture, the length of arm is acquired by adding the length of upper arm and lower arm. Another example is the acquisition of character height, for a character in a standing straight posture, the height can be obtained by calculating the Euclidean distance from top of head to the bottom of heel. However, when character bows or sits, the aforementioned measuring method no longer suitable for the circumstance. The height need to be measured separately from head, neck, torso and length of leg. With different posture, the datum point used for measuring are also differs.

In tailoring, when measuring customer for making a cloth, one end of a tape ruler is held to one of two datum points and tension is applied to the tape ruler so that it follows the profile of the body as closely as possible when it reaches the second datum point. At this point, due to the applied tension on tape ruler, the length of ruler reaches shortest following the body profile between these two datum points. Geodesic path is the shortest path between two points on a curved surface. Therefore, in this thesis, geodesic is used for measuring length of body parts of characters. Using geodesic to extract length measurement can also copes with different character postures. For example, Figure 3.3 demonstrates different measurement caused by different posture. The red cone indicates the datum points on the shoulder and wrist of the character. The green curve indicates the geodesic generated by the

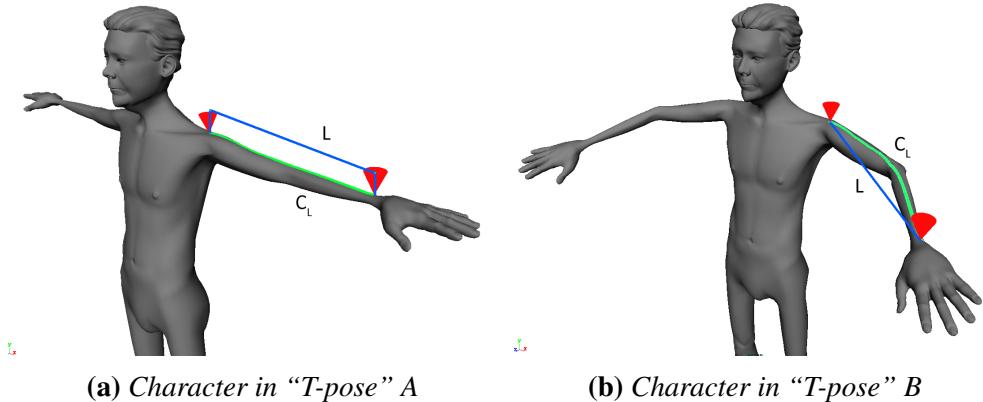


Figure 3.3: Measurements in different postures

method presented in this chapter between two datum points and blue line is the straight line connecting two datum points. L denotes the length of the blue line(the Euclidean distance between two datum points), C_L denote the arc length of geodesic. The character is in two most common “T-pose”. In Figure 3.3, when the character in “T-pose” A, $L = 51.708$ and $C_L = 51.075$, when character in “T-pose” B, $L = 45.867$ and $C_L = 51.494$. This shows the geodesic distance between two datum points are more consistence in different posture than Euclidean distance when measuring length of the body part of character. Therefore in this thesis, the length measurement are acquired by measuring the geodesic between two datum points on the body of the character in order to cope with different postures of the character.

In order to calculate geodesic, a graph with positive edge weight is required. The discrete polyhedral surface, as the most common shape representation in computer graphic, has this connectivity. Therefore, in the past few decades, many algorithms have been developed to compute the geodesic on discrete polyhedral surfaces. However, current geodesic computation on a high resolution polygon model is still a very time consuming process. In many applications, such as surface flattening(Zigelman et al. 2002) and remeshing(Peyré & Cohen 2006b), the efficiency of geodesic algorithm has become the bottle neck of its implementations.

This chapter presents a novel discrete geodesic computation scheme to improve the efficiency of geodesic computation on a discrete polyhedral surface. It consists of two algorithms for accurate geodesic path computation and high efficiency approximate geodesic computation. Among these algorithms, the approximate algorithm for polyhedral surface achieves linear time complexity whilst maintain a small bounded error.

3.1 Introduction

For different shape representation, geodesic has different definitions. Firstly, for parametric surface, geodesic refers to a curve connecting two points on surface, such that the geodesic curvature at any point on the curve is zero (Polthier & Schmies 2006).

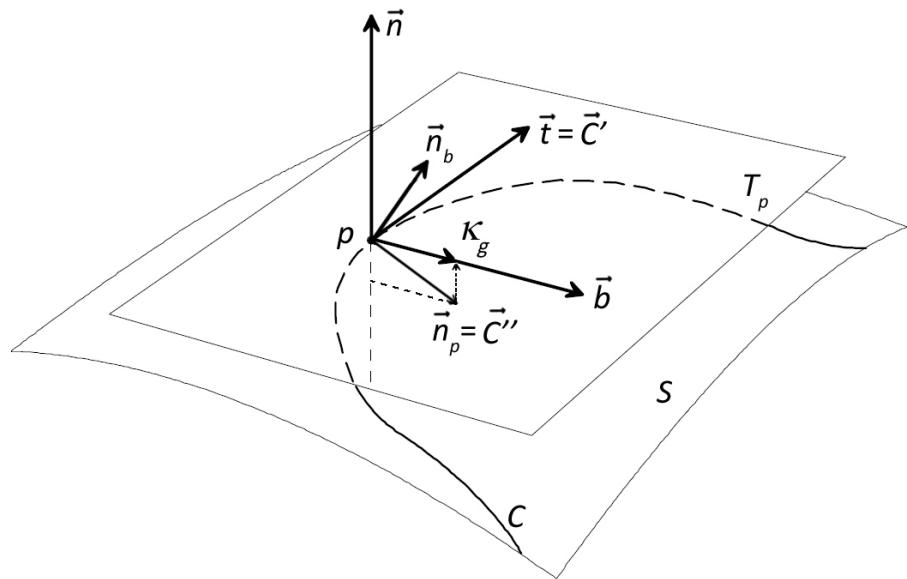


Figure 3.4: Illustration of geodesic curvature

Figure 3.4 demonstrates geodesic curvature at a point on a curve, where

S is a two-dimensional smooth surface. C is a curve on S , p denotes a point on C which has arc length s to C . Tp is the tangent plane at point $p \in S$, \vec{n} is the surface normal at p , \vec{C}'_s denote first order derivative of C and its the tangent vector at point p . \vec{b} is a vector perpendicular to \vec{n} and \vec{C}'_s . \vec{C}''_s denotes the second order derivative of C at p that has arc length s to C which is the geodesic curvature vector of C at p . \vec{n}_b is the binormal unit vector that is perpendicular to \vec{C}'_s and \vec{C}''_s . κ_g is the length of the projection of \vec{C}''_s on \vec{b} .

Definition 1 (Polthier & Schmies 2006) Let S be a two-dimensional parametric surface, a curve C is a geodesic if one of the equivalent properties holds:

1. C is locally shortest curve(C has shortest arc length).
2. \vec{C}''_s is parallel to the \vec{n} .
3. C has vanishing geodesic curvature $\kappa_g = 0$.

Secondly, in order to define geodesic on convex polyhedral surface, the vertices of the polyhedral surface need to be categorized first. This can be done based on the total vertex angle.

Definition 2 (Polthier & Schmies 2006) Let S be a polyhedral surface and a vertex $v \in S$. F_v denotes the one-ring neighbour faces of v , which can be written as $F_v = f_1, \dots, f_i, \dots, f_m$, θ_i is the interior angle of f_i at v . Then the total vertex angle $\theta(v)$ is given by,

$$\theta(v) = \sum_{i=1}^m \theta_i(v)$$

All vertices of a polyhedral surface can be categorized based on the sign of the “vertex angle excess”, which can be calculated by $2\pi - \theta(v)$. Figure 3.5 demonstrates three types of vertices on polyhedral surface.

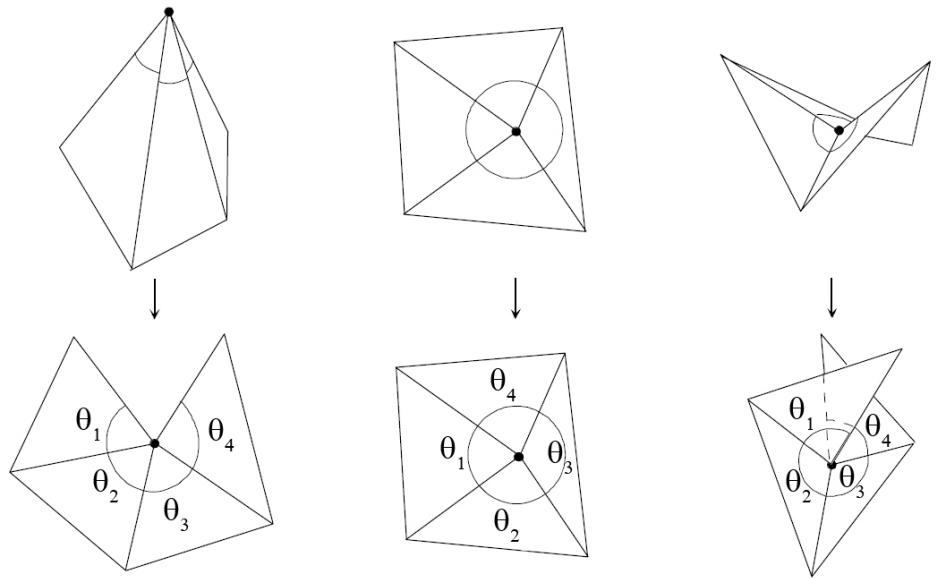


Figure 3.5: Spherical Vertex(left), where $2\pi - \sum\theta > 0$; Euclidean Vertex(middle), where $2\pi - \sum\theta = 0$; Hyperbolic Vertex(right), where $2\pi - \sum\theta < 0$.

Therefore a geodesic on polyhedral surface can be defined as follow,

Definition 3 Let S be a two-dimensional polygon surface, A piecewise curve C is a geodesic if one of the equivalent properties holds(Polthier & Schmies 2006): “A geodesic path P goes through an alternating sequence of hyperbolic vertices and (possibly empty) edge sequences such that the unfolded image of the path along any edge sequence is a straight line segment and the angle of the path passing through a vertex is greater than or equal to π . No edge can appear more than once in an edge sequence”(Mitchell et al. 1987a).

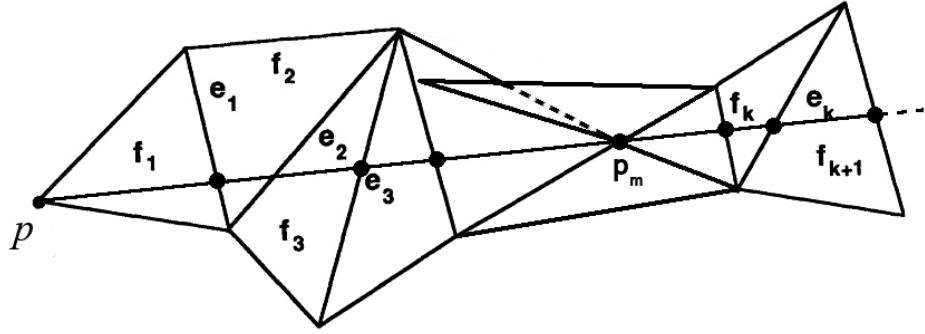


Figure 3.6: Geodesic on a polyhedral surface, where $f_1 \cdots f_k$ is the planar unfolding of S . $e_1 \cdots e_k$ is the edge sequence E the geodesic goes through, p_m is a hyperbolic vertex on the polygon surface.

Figure 3.6 illustrates a geodesic starts at point p and goes through a sequence edge $e_1 \dots e_k$. After each all the that a geodesic goes through are flattened, its path forms a straight line.

According to the theory of Chopp (1993), given an initial piecewise curve that passes a series of vertices on the polyhedral surface, after defining the tangent plane at every point on this piecewise curve, by moving each point on the piecewise curve iteratively for a small step at direction of \vec{b} which is indicated in Figure 3.4, the geodesic curvature κ_g can be eliminated gradually. When this iteration converges, which means the geodesic curvature at every point on this piecewise curve has been eliminated, this piecewise curve becomes a geodesic on the polyhedral surface.

In the following sections, a novel geodesic curvature flow based geodesic computation scheme is introduced in detail. Two algorithms have been developed to solve different problems,

1. Accurate geodesics computation on polyhedral surface
2. Linear time complexity approximate geodesics computation with a bounded error on polyhedral surface

3.2 Geodesic Curvature Flow

Based on geodesic curvature defined in Definition 1, an iterative scheme is proposed for computing geodesics over meshes. This scheme employs iterative regression to eliminate geodesic curvature, when the process converges, the curve becomes a geodesic.

Clairaut (1731); Serret (1851) introduced Frenet-Serret formulas to describe the kinematic properties of a particle moving on a continuous and differentiable curve in three-dimensional Euclidean space \mathbb{R}^3 . In other words, this formulas describes the derivatives among the tangent vector, normal vector and binormal vector of a point on a continuous and differentiable curve. Frenet-Serret formulas is defined as follow,

Let C be a curve in Euclidean space. This curve C can be parametrized by its arc length s . A point on curve C that has arc length s can be denoted by $C(s)$. Then the Frenet-Serret frame is defined by three vectors,

The first order derivative of C , also known as the tangent unit vector \vec{t} is defined as:

$$\vec{C}'_s = \vec{t} = \frac{dC(s)}{ds} \quad (3.1)$$

The second order derivative of C , also known as the principle normal unit vector \vec{n}_p of the curve is defined as:

$$\vec{C}''_s = \vec{n}_p = \frac{\frac{d\vec{t}}{ds}}{\left\| \frac{d\vec{t}}{ds} \right\|} \quad (3.2)$$

The binormal unit vector of C is defined as:

$$\vec{n}_b = \vec{n}_p \times \vec{t} \quad (3.3)$$

Note that \vec{n}_b' , \vec{t} and \vec{n}_p are perpendicular to each other, the plane defined by \vec{t} and \vec{n}_p is the osculating plane at point $C(s)$.

In order to calculate the gradient for the iteration, firstly, a natural coordinate system need to be defined by the matrix form of Frenet-Serret formulas (Kreyszig 1991),

$$\begin{bmatrix} \frac{d\vec{t}}{ds} \\ \frac{d\vec{n}_p}{ds} \\ \frac{d\vec{n}_b}{ds} \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix} \begin{bmatrix} \vec{t} \\ \vec{n}_p \\ \vec{n}_b \end{bmatrix} \quad (3.4)$$

where $\tau = \langle -\vec{n}, \vec{n}_b' \rangle$ denotes the torsion which measures the turnaround of the binormal vector \vec{n}_b' , “ \langle, \rangle ” is the dot production of two vectors, $\kappa = 1/r$ is the curvature where r is the radius of the osculating circle. s is the arc length of the curve. Equation 3.4 can be written into following form,

$$\begin{cases} \frac{d\vec{t}}{ds} = \kappa \vec{n}_p \\ \frac{d\vec{n}_p}{ds} = -\kappa \vec{t} + \tau \vec{n}_b \\ \frac{d\vec{n}_b}{ds} = -\tau \vec{n}_p \end{cases} \quad (3.5)$$

Based on Definition 1, let \vec{n} denotes standard unit normal of surface S , κ_p denotes geodesic curvature and τ denotes geodesic torsion. The Frenet-Serret formula can be defined on a tangent plane T_p by,

Equation 3.5 can be written as,

$$\begin{cases} \frac{d\vec{t}}{ds} = \kappa_n \vec{n} + \kappa_g \vec{b} \\ \frac{d\vec{b}}{ds} = -\kappa_g \vec{t} + \tau_g \vec{n} \\ \frac{d\vec{n}}{ds} = -\tau \vec{n} - \kappa_g \vec{t} \end{cases} \quad (3.6)$$

where $\kappa_n = \langle \vec{C}_s'', \vec{n} \rangle$ is the normal curvature at $C(s)$, $\kappa_g = \langle \vec{C}_s'', \vec{b} \rangle$ is the geodesic curvature at $C(s)$. $\tau_g = \tau - \frac{d\theta}{ds}$ is the geodesic torsion, θ is the

angle between n and C'' . C_s'' denote the second order derivative to arc length s . Now, geodesic curvature flow can be defined as follow,

Definition 4 (*Chopp 1993*), Let $S \subset \mathbb{R}^3$ be a two-dimensional manifold embedded in three-dimensional space. $C(s)$ is a curve on S that is parametrised by arc length s and it is moving with speed $F(\kappa_g)$ in the direction perpendicular to $C(s)$, κ_g is the geodesic curvature of $C(s)$ on S , \vec{n} denotes the normal of S and it is continuous on S . Therefore, at every point on $C(s)$, based on the Frenet-serret frame, a natural coordinate system can be given by the vectors \vec{C}'_s , \vec{t}_s and \vec{C}''_s . C'_s is the first order derivative of $C(s)$, Thus, for any point $C(s)$, the speed can be defined by geodesic curvature is,

$$\kappa_g = \langle (\vec{n} \times C'_s), C''_s \rangle \quad (3.7)$$

therefore, the velocity of the geodesic curvature flow $C(s, t)$ can be given by,

$$k_g \vec{b} = C''_s - \langle \vec{C}''_s, \vec{n} \rangle \vec{n} \quad (3.8)$$

where $\langle \vec{C}''_s, \vec{n} \rangle \vec{n}$ is the projection of \vec{C}''_s on \vec{n} , note that t is a time variable of the flow in which $t = 0$ states the initial status of flow $C(s, t)$.

This flow is also known as the Euclidean curve shortening flow (Salden et al. 1999). Spira & Kimmel (2002); Wu & Tai (2010) proved that the flow of Equation 3.8 can minimizes geodesic curvature $C''_{s,t}$ point-wise on $C(s, t)$ by moving $C(s, t)$ in gradient direction.

Polthier & Schmies (2006) proved that, on polyhedral surface, if a discrete geodesic does not pass any hyperbolic vertex, this geodesic is also a shortest geodesic. In this thesis, the concept of vanishing geodesic curvature is exploited from smooth surface geodesic computation to the discrete surface geodesic computation. In computer graphic, a polyhedral surface can be considered as the inscribed polygon approximation of a smooth surface.

Therefore a curve on smooth surface can also be approximated by a piecewise curve consists of a set of fixed number of sample points. However, there is no parametric form for the piecewise curve $C(s)$ nor for the polyhedron surface. Especially for high resolution model, which usually has millions of vertices, creating parametric approximation surface for such a polyhedron requires a large amount of computational resources. In order to avoid this excess calculation, the second order derivative of $C(s)$ is directly defined on the piecewise curve by $\widetilde{p_{i-1}p_ip_{i+1}}$ as shown in Figure 3.7, where C denotes a curve on smooth surface S and the $C(s)$ is the parametrized form of C by its arc length. p_{i-1}, p_i and p_{i+1} are three sample points on C . Thus $\widetilde{p_{i-1}p_ip_{i+1}}$ denotes a section of the piecewise approximation of smooth curve C . In this case, at p_i , the second order derivative of C_s at p_i can be estimated by Equation 3.9,

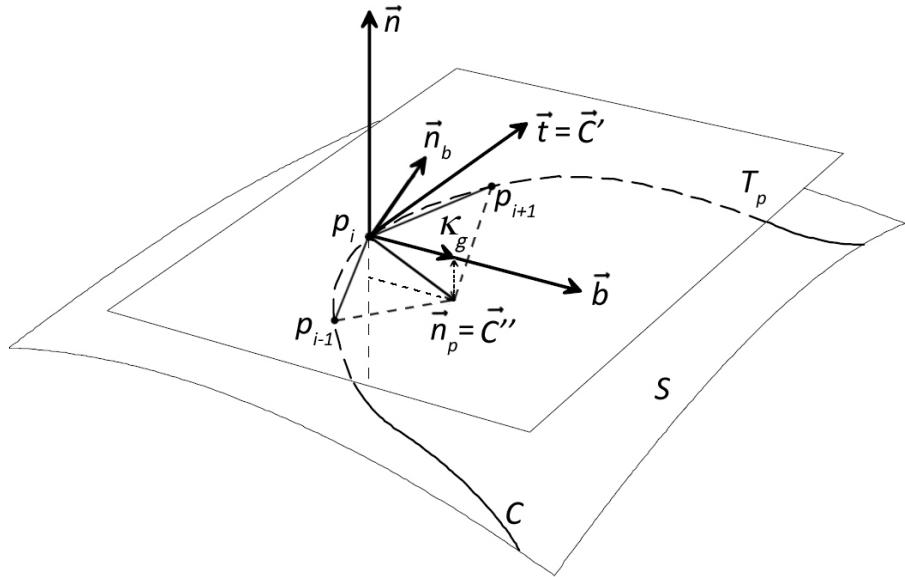


Figure 3.7: Geodesic curvature on a piecewise curve

$$\begin{aligned}
\vec{C}_s'' &\approx \overrightarrow{p_ip_{i+1}} + \overrightarrow{p_ip_{i-1}} \\
&\approx (p_{i+1} - p_i) + (p_{i-1} - p_i) \\
&\approx p_{i-1} - 2p_i + p_{i+1}
\end{aligned} \tag{3.9}$$

In order to define the natural coordinate system on a vertex of a polyhedral surface, the tangent plane of S at p_i needs to be confirmed. Assume that the polyhedral surface is a discrete approximation of a smooth surface S , given a point p_i on $C(s)$, the tangent plane of the polyhedron at vertex p_i can be defined by the fitting plane of the one-ring neighbour vertices of p_i . According to Definition 1, it can be said that, if the projection of \vec{C}_s'' onto this tangent plane at p_i vanishes, the geodesic curvature will also vanish at p_i accordingly. Furthermore, if geodesic curvature at all the point on the piecewise approximation curve of $C(s)$ vanishes, $C(s, t)$ reaches a stable states, and becomes a geodesic.

Let P denotes all the points on the piecewise approximation curve of $C(s)$, the geodesic curvature of $C(s)$ can be written as $\vec{C}_s'' = KP$.

$$K = \begin{bmatrix} 0 & 0 & 0 & \cdots & & & & & & & & 0 \\ 0 & 0 & 0 & \cdots & & & & & & & & 0 \\ 0 & 0 & 0 & \cdots & & & & & & & & 0 \\ 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & \cdots & & & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & \cdots & & 0 \\ 0 & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & \cdots & 0 \\ & & & \ddots & & & \ddots & & & & \ddots & \\ 0 & \cdots & & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & 0 \\ 0 & \cdots & & & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 \\ 0 & \cdots & & & & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 \\ 0 & \cdots & & & & & & & & & 0 & 0 & 0 \\ 0 & \cdots & & & & & & & & & 0 & 0 & 0 \\ 0 & \cdots & & & & & & & & & 0 & 0 & 0 \end{bmatrix}_{3m \times 3m}$$

and

$$P = [p_{1x}, p_{1y}, p_{1z}, \dots, p_{mx}, y_{my}, p_{mz}]^T$$

where K is a $3m \times 3m$ coefficient matrix evolved from the coefficients in Equation 3.9, m denotes the number of sample points on piecewise curve. Note that, elements of the first and the last three rows of K are set to 0. This is because by using Equation 3.9 to approximate \vec{C}_s'' on a piece-wise curve $C(s)$, three points on $C(s)$ is required. However, on the first and the last point on $C(s)$, only one sample exist within its one-ring neighbour. Moreover, during geodesic curvature flow calculation, the position of source point and destination point should not be moved. Therefore, in K , the first and last three rows which correspond to the calculation of first and last point on $C(s)$ are set to 0 explicitly to avoid the movement on both source and destination point of $C(s)$.

Next, constructing a matrix \vec{n} which is a block diagonal matrix consists of normal vectors of the tangent plane at every point in P .

$$\vec{n} = \begin{bmatrix} n_{1x} \\ n_{1y} \\ n_{1z} \\ n_{2x} \\ n_{2y} \\ n_{2z} \\ \vdots \\ n_{mx} \\ n_{my} \\ n_{mz} \end{bmatrix}_{3m \times m}$$

Now, geodesic curvature flow $C(s, t)$ can be rewritten as,

$$C(s, t) = k_g \vec{b} = KP - \vec{n} \vec{n}^T KP \quad (3.10)$$

Let P_t denotes all the sample points on $C(s)$ at t time, μ denotes a iterative step length. when t evolves into $t + 1$, all the points in P_t move towards the direction of \vec{b} . Therefore, the updated curve denoted as P_{t+1} can be expressed as,

$$P_{t+1} = P_t + \mu(KP_t - \vec{n} \vec{n}^T KP_t) \quad (3.11)$$

where μ is an iterative step size which heavily affects the stability of curve updating. In order to maintain the stability, μ is usually set to a small value, in our experiment, $\mu = 0.0001$. However, setting μ into a small value requires extensive increment on number of iterations required for eliminating geodesic curvature.

3.3 Geodesic on Mesh

3.3.1 Tangent Space Constraint

In order to calculate geodesics on a polyhedral surface using Equation 3.11, the normal vector needs to be estimated at every iteration for all the points on polyhedral surface corresponding to every sample point till Equation 3.11 converged.

Because Equation 3.11 updates points on $C(s)$ in the direction that is on tangent plane of surface S at the initial position of p . If S is a non-flat surface, moving a point on tangent plane will cause this point to deviate from S . In order to solve this problem, a constrain in the tangent space needs to be applied to the movement of sample points. This can be implemented as follow.

Firstly, let P denotes all the sample points on curve C , all the normal vectors of one-ring neighbour faces of the closest vertex to p_i on S is stored in to a matrix N ,

$$N = \begin{bmatrix} n_{1_x} & n_{1_y} & n_{1_z} \\ & \vdots & \\ n_{m_x} & n_{m_y} & n_{m_z} \end{bmatrix}_{m \times 3}$$

Applying PCA (Jolliffe 2002) to N , then the tangent space can be defined by the eigenvector which associated with the largest eigenvalue. After that, for each $p_i \in P$, let $p_{i,t}$ denotes the current position of p_i and $p_{i,t+1}$ denotes the p_i after it moves to the direction of its geodesic curvature vector. This update of location of p_i can be expressed by,

$$p_{i,t+1} = p_{i,t} + \mu \vec{n}_i \vec{n}_i^T (\overline{p_{i,t}} - p_{i,t}) \quad (3.12)$$

where, μ is an iterative step size and $\overline{p_{i,t}}$ denotes the projection of p_i on

tangent plane at the closest vertex to p_i on S . Every time when p is updated by Equation 3.12, it is always projected back to tangent plane of the closest vertex to it. In essential, the tangent space constraint of Equation 3.12 always moves a point within the tangent space and therefore it tends to preserve the characterization of the underlying surfaces. After that, a steady solution can be formed by combining Equation 3.12 with Equation 3.11.

$$P_{t+1} = P_t + \mu(\vec{n}\vec{n}^T\bar{P}_t - \vec{n}\vec{n}^TP_t + KP_t - \vec{n}\vec{n}^TKP_t) \quad (3.13)$$

where, \bar{P}_t is a column vector consisting of the coordinates of all the projections of the sampling point of curve C on S .

3.3.2 Implicit Euler Scheme

In order to minimize κ_g by using Equation 3.13, the explicit method (Euler method) is a straightforward solution (Butcher 2008). However, because Euler method is a first-order method, its local error is proportional to the square of the step size, and its global error is proportional to the step size (Butcher 1987). In order to maintain the stability of the solution, integration must be proceed with a very small iterative step size. Moreover, this iteration performed for every geodesic on a mesh, therefore results in a very time-consuming process.

The basic idea of the implicit Euler method is to employ the implicit differencing, i.e, the right-hand side of Equation 3.13 is evaluated at the new location of $t + 1$. This is called as the backward Euler scheme (Enns & McGuire 2000). Although the error of the backward Euler scheme is the same as explicit Euler method, the method could maintain its stability with any step size (Hairer 2010). By increasing the step size of the iteration, the implicit Euler method could achieve much higher efficiency (Butcher 2008).

In order to apply backward Euler scheme to Equation 3.13, firstly, move all parts which contains P_t on the right hand side of Equation 3.13 to the left hand side. Equation 3.13 then can be written as,

$$P_{t+1} + P_t(\mu(\vec{n}\vec{n}^T - K + \vec{n}\vec{n}^T K)) = P_t + \mu\vec{n}\vec{n}^T\overline{P}_t \quad (3.14)$$

New, let the P_t in parts which was moved from right hand side of Equation 3.13 are evaluated by P_{t+1} . Then applying the backward scheme to Equation 3.13 yields,

$$(I + \mu(\vec{n}\vec{n}^T - K + \vec{n}\vec{n}^T K))P_{t+1} = P_t + \mu\vec{n}\vec{n}^T\overline{P}_t \quad (3.15)$$

where I is an identity matrix, μ denotes the step size. The resulting geodesic curvature flow of Equation 3.15 remains stable even at $\mu \rightarrow \infty$ (Hundsdorfer & Verwer 2003). Moreover, Equation 3.15 is the linear equation form of $AX = B$ where $A = I + \mu(\vec{n}\vec{n}^T - K + \vec{n}\vec{n}^T K)$, and $B = P_t + \mu\vec{n}\vec{n}^T\overline{P}_t$. This equation can be solved efficiently by LU factorization with full pivoting presented by Trefethen & Bau (1997).

3.3.3 Least Squares Scheme

Although the implicit Euler scheme provides an efficient method for solving Equation 3.15. The experiment shows that the geodesics result from Equation 3.15 still requires a few iterations in order to achieve convergence. Because during the convergence of Equation 3.11 , all the sample points on geodesic curve are updated within the tangent space except two endpoints p_1 and p_m of curve $C(t)$. Therefore, Equation 3.11 can be written in a least squares form

in which two endpoints p_1 and p_m are the constraints (Jiang 1998),

$$(K - \vec{n}\vec{n}^T K) p = [p_{1_x}, p_{1_y}, p_{1_z}, 0, \dots, 0, p_{m_x}, p_{m_y}, p_{m_z}]^T \quad (3.16)$$

where,

$$K = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & & 0 \\ 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & \cdots & 0 \\ & & \ddots & & & \ddots & & & \ddots & \\ 0 & \cdots & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & 0 \\ 0 & \cdots & & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 \\ 0 & \cdots & & & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 \\ 0 & \cdots & & & & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \cdots & & & & & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \cdots & & & & & & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\vec{n} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ n_{2x} \\ n_{2y} \\ n_{2z} \\ \ddots \\ n_{m-1x} \\ n_{m-1y} \\ n_{m-1z} \\ 0 \\ 0 \\ 0 \end{bmatrix}_{3m \times m}$$

where K is a $3m \times 3m$ size matrix, \vec{n} denotes the normal of the tangent plane. Essentially, Equation 3.16 is the 1^{st} order approximation to the geodesic curvature flow.

Here, define a matrix A which contains all the geodesic curvature of sample points on curve $C(s)$.

$$A = K - \vec{n}\vec{n}^T K \quad (3.17)$$

Then a linear system can be constructed from Equation 3.16,

$$AP = [p_{1x}, p_{1y}, p_{1z}, 0, \dots, 0, p_{m_x}, p_{m_y}, p_{m_z}]^T \quad (3.18)$$

where, m is the number of sample points on piecewise curve C . Next, define a matrix N^* where $N^* = \vec{n}\vec{n}^T$ and a matrix B , where $B^* = \vec{n}\vec{n}^T \bar{P}$. Combining Equation 3.18 with tangent plane constrain defined by Equation

3.12, the Equation 3.18 can be written as,

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,3m} \\ A_{1,0} & A_{1,1} & & \\ \vdots & \ddots & & \vdots \\ A_{3m,0} & \cdots & A_{3m,3m} \\ N_{0,0}^* & N_{0,1}^* & \cdots & N_{0,3m}^* \\ N_{1,0}^* & N_{1,1}^* & & \\ \vdots & \ddots & & \vdots \\ N_{3m,0}^* & \cdots & N_{3m,3m}^* \end{bmatrix} P = \begin{bmatrix} p_{1_x} \\ p_{1_y} \\ p_{1_z} \\ 0 \\ \vdots \\ 0 \\ p_{m_x} \\ p_{m_y} \\ p_{m_z} \\ B_{0,0}^* \\ \vdots \\ B_{3m,0}^* \end{bmatrix} \quad (3.19)$$

where, $A_{i,j}$, $N_{i,j}^*$ and $B_{i,j}$ denote the corresponding element in row i and column j in matrix A , N^* and B accordingly. \bar{P} is a column vector consists of all the projections of p_i on tangent plane defined by \vec{n} and the closest vertex on S to p_i . m denotes the number of sample point on curve. The coefficient matrix on the left hand side of the system has a size of $(2 * 3m) \times 3m$ and the one on the right hand side of the system has the size of $(2 * 3m) \times 1$.

Note that Equation 3.19 cannot computes a geodesic curve from scratch but requires the initial location of the sample points for computing \bar{p} . The vertices near a geodesic path are usually taken as the initial sample points on the path. Such initial sample points are used for computing \bar{p} and \vec{n} in Equation 3.19, which is then solved to obtain the final locations. According to this scheme, the final location of a sample point is still constrained within the tangent space defined by its initial location. Note that the solution of Equation 3.19 cannot guarantees the updated sample points lying on the polyhedral

surface since the tangent space and actual polyhedral surface space are two different surface. the actual polyhedral surface is reckoned as inscribed polygon of the surface that is represented by tangent space. In the case that require the geodesic path on polyhedral surface, a projection method was developed to project the geodesic path from smooth surface to its approximate mesh.

3.3.4 Geodesic Path Projection

Although sample point updated by Equation 3.13 only moves within tangent space, the actual mesh is not flat. Therefore, all updated points tend to deviate from mesh. This is showed in Figure 3.8,

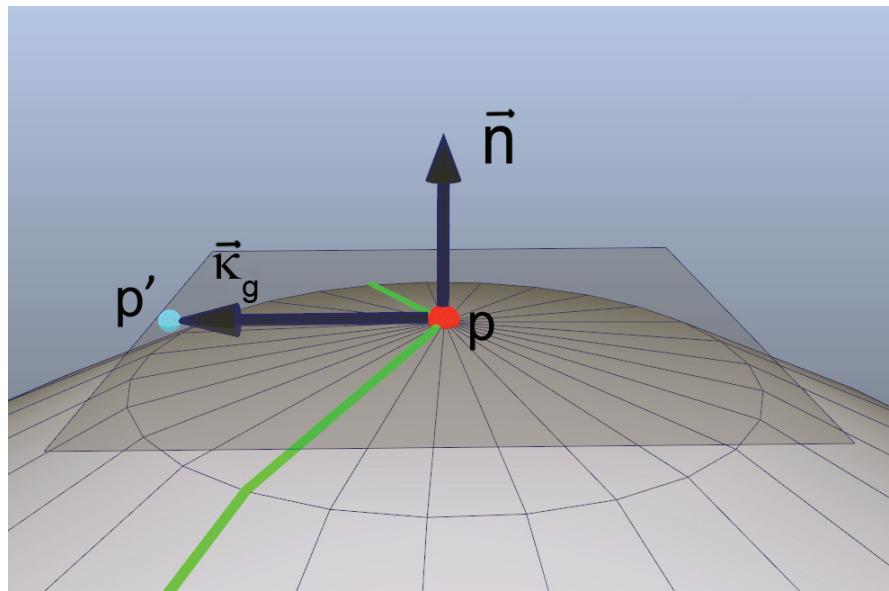


Figure 3.8: Derived point caused by updating its location within tangent space, green piecewise curve is the curve before it is updated by Equation 3.11, gray plane is the tangent plane at point p (red point), the point p' (light blue point) is the updated point.)

In order to calculate geodesics on the polyhedral surface, the offset geodesic curve needs to be projected back onto polyhedral surface. Polthier & Schmies (2006) defines geodesic as when at any point on the curve, geodesic curvature vanishes. On polyhedral surface, if a curve fits this defi-

nition and does not pass any spherical vertex on the polyhedral surface, it is both straightest and locally shortest geodesic. This is shown in Figure 3.9.

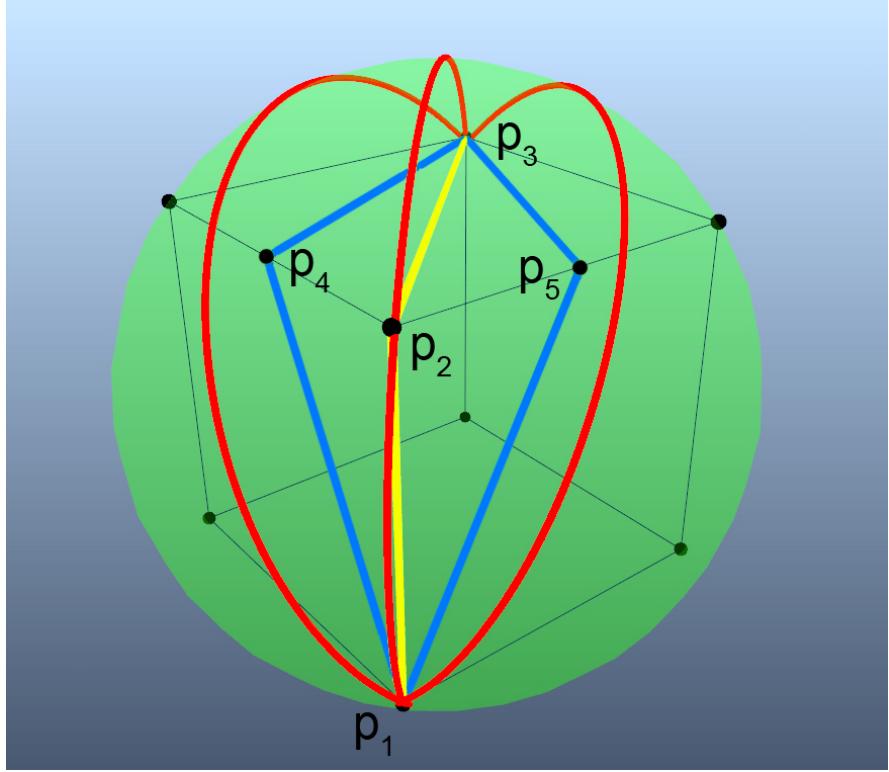


Figure 3.9: Geodesic on smooth surface and its projection on its inscribed polygon mesh, the green sphere contains its inscribed cubic, the red curves is the geodesic on sphere (great circle arc), p_1, p_2 and p_3 are points on sphere, p_4 and p_5 are the centre point on its edge.

In Figure 3.9 both the yellow line $\widehat{p_1p_2p_3}$ and the blue lines $\widehat{p_1p_4p_3}$, $\widehat{p_1p_5p_3}$ are projections of their corresponding red geodesic path on cube. Path $\widehat{p_1p_2p_3}$ that goes through the spherical vertex p_2 is the straightest but not the shortest geodesic path on cube. However, the blue lines that pass p_4 or p_5 is both the straightest and shortest path on cube according to the definition presented in (Polthier & Schmies 2006). Note that $\widehat{p_1p_4p_3}$ and $\widehat{p_1p_5p_3}$ are projections of the great circle on sphere in the direction of the curvature of their corresponding great circle.

Here, in order to project floating geodesics onto mesh, firstly, the direction of projection needs to be defined. According to the definition of geodesic

curvature, geodesic curvature vector $\vec{\kappa}_g$ of a curve C at point p is the projected vector of the curvature vector \vec{C}'' at p onto the tangent plane T_p at point p on surface S . This is shown in Figure 3.4. Therefore, the curvature κ at point p can be written as:

$$\vec{\kappa} = \vec{\kappa}_g + \vec{\kappa}_n$$

where $\vec{\kappa}_g$ is the component of \vec{C}'' along \vec{b} , $\vec{\kappa}_n$ is the component of \vec{C}'' along \vec{n} . \vec{n} is the normal vector of point p on S . Let the magnitude of vector $\vec{\kappa}_g$ be denoted by κ_g , then $\vec{\kappa}_g = \kappa_g \vec{b}$. the scalar κ_g is called geodesic curvature of C at p . Let κ be the magnitude of vector \vec{C}'' . Then the curvature \vec{C}'' of curve C at p can be expressed by geodesic curvature κ_g at p by:

$$\kappa_g = \kappa \cos \theta$$

where θ is the angle between osculating plane of C at point p and tangent plane T_p . Therefore, if p moves along the direction of \vec{C}'' , value of θ will not be changes and osculating plane of C at point p stays the same. Hence geodesic curvature κ_g will remain identical. Therefore, for each sample point on C the projection direction is its C'' . However, since there is no guarantee that every successive sample point can be projected into the successive faces on polygon, actually, in our experiments, in most cases, the successive sample point of current sample point will falls onto the face that outside of one-ring neighbour face of the face contains current projected point. Here, by using osculating plane of C at p , we have developed a method connecting all the projected point of p . The procedure is described as follows.

Firstly, select p_0 as the starting point that always lies on a vertex of mesh. With two successive sample points p_1 and p_2 on the floating geodesic, these three points form a plane which cuts mesh S . This plane, $\Delta p_0 p_1 p_2$, is viewed as the approximation of the osculating plane at p_1 , this is shown in Figure 3.10.

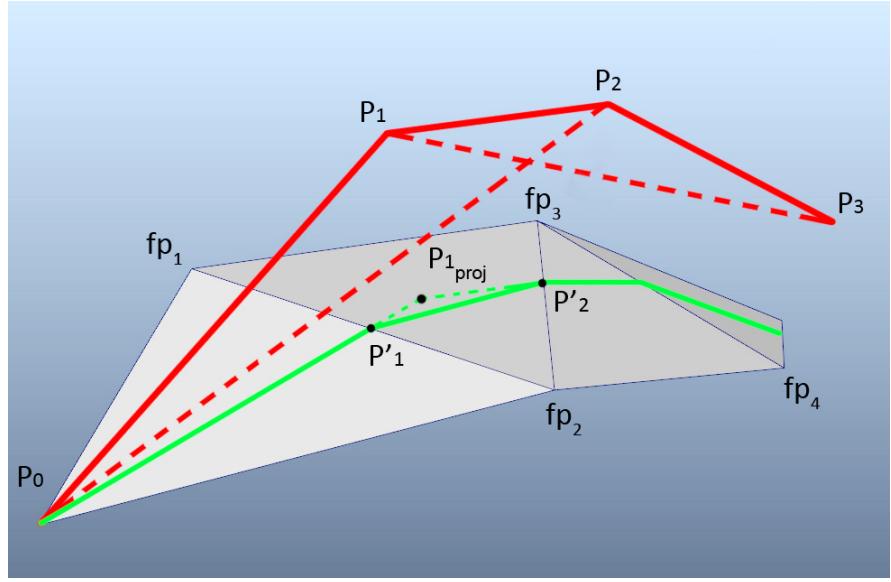


Figure 3.10: Projection of a geodesic path.

In Figure 3.10, red line indicate the piecewise approximation of a geodesic. p_0, p_1, p_2, p_3 are the sample points on geodesic path $\widetilde{p_0p_3}$. $\Delta p_0p_1p_2$ is the approximation of the osculating plane at p_1 . fp_1 fp_4 are the vertices on polyhedral surface. P_{1proj} is the projection of p_1 . p_0 which denotes the source point of geodesic path, p'_1 and p'_2 are the point on projected geodesic path which depicted in green line.

Let \vec{N}_1 denotes the normal vector of osculating plane $\Delta p_0p_1p_2$ and \vec{N}_2 denotes the normal vector of face $\Delta p_0fp_1fp_2$. Firstly, starts from the source point p_0 , the rectifying plane of the projected path is approximated by $\Delta p_0fp_1fp_2$. As a result, the tangent vector of projected geodesic at p_0 can be calculated by Equation 3.20.

$$\vec{v} = \vec{N}_1 \times \vec{N}_2 \quad (3.20)$$

where \vec{v} is equivalent to $\vec{p_0p'_1}$. Following this direction, \vec{v} intersects with an edge within the one-ring neighbour faces of p_0 . p'_1 denotes the intersection on edge $\overline{fp_1fp_2}$. Then, p'_1 is selected as the next starting point. An impor-

tant property of manifold mesh is that, only one or two faces are incident to an edge. Therefore, the next projected face can be easily determined as the adjacent face $\Delta fp_1fp_2fp_3$ to current face $\Delta p_0fp_1fp_2$ through edge $\overline{fp_1fp_2}$.

If the projection of p_1 does not falls into $\Delta fp_1fp_2fp_3$, the next intersecting point on face $\Delta fp_1fp_2fp_3$ can be determined by Equation 3.20. If the projection of p_1 falls into the face $\Delta fp_1fp_2fp_3$ as shown in Figure 3.10, \vec{N}_1 is updated by the normal vector of approximated osculating plane $\Delta p_1p_2p_3$ at point p_2 . Starting from projection p_{1proj} of p_1 , perform Equation 3.20, the next intersection of \vec{v} and $\overline{fp_2fp_3}$ can be determined as p'_2 shown in Figure 3.10. The projection method is summarized in Algorithm 1

Algorithm 1 Geodesic Projections on Mesh

```

1: procedure PROJECT GEODESIC PATH ON MESH(A mesh  $S$ , and a floating geodesic path  $\widetilde{p_0p_n}$ )
2:   for  $i = 1, i < m, i++$  do ( $m$  denote the number of sample points on a floating geodesic)
3:     Select the successive vertices of  $p_i$  and build osculating plane  $\triangle p_{i-1}p_ip_{i+1}$ 
4:      $\vec{N}_1 \leftarrow$  normal vector of  $\triangle p_{i-1}p_ip_{i+1}$ 
5:      $F_{end} \leftarrow$  Face contains projection of  $p_i$  on  $S$ 
6:     select  $p_{i-1}$  as the starting point  $p_s$ 
7:     while  $F' \neq F_{end}$  do
8:       for  $F' \in$  adjacent faces of  $p_s$  do
9:          $\vec{N}_2 \leftarrow$  normal vector of current face
10:         $\vec{t} \leftarrow p_s + \vec{N}_1 \times \vec{N}_2$ 
11:        Compute the intersection point  $p'$  that an edge of current face  $F'$  intersects with  $\vec{t}$ 
12:        if  $p'$  is within the edge then
13:          Add  $p'$  into projection path
14:          Update  $p_s$  by  $p'$ 
15:          Update  $F'$  by the adjacent face of  $F'$ 
16:        end if
17:      end for
18:    end while
19:    Update  $p_s$  by  $p_i$ 
20:  end for
21: end procedure

```

3.3.5 “Continuous Dijkstra” Propagation

As described in Equation 3.11, this algorithm needs an initial path to calculate a updated geodesic path. “continuous Dijkstra” strategy presented by Dijkstra (1959) is used to generate this initial path. When performing “continuous Dijkstra” strategy, all points in the propagation boundary are kept sorted by the distance back to the source point in an incremental order in a priority queue. At each step, starting from the first element in the priority queue, the boundary is propagated outward. New points are inserted into the queue at the location where the order of queue remains increment.

However, keeping the priority queue involves “comparison sorting algorithm” which is a searching algorithm which costs $\log(n)$ time to n factorial possible orderings of a data set (Cormen et al. 2001). This process requires significant amount of time to complete especially when performing on a high resolution model. Because Equation 3.11 only requires an initial path that approximates to the shortest path. Vertices in the propagation boundary do not need to be kept in such an order. This algorithm is presented in detail below.

Starting from the source p_s , firstly, the 1st ring vertices of p_s are pushed into an array denoted as “wavefront” W that is equivalent to the propagation boundary. Each edge that connects p_s and a point in W is the geodesic path between them. This is due to within a triangle, the shortest path between two points is the edge that connects these two points. Next, the length of every edge is stored along with the path information to vertices in W . Since this algorithm does not employ priority queue,in order to maintain isometric propagation, a propagation radius limitation r_{max} is introduced to constrain the propagation at every step. At this moment, r_{max} is updated as the length of the longest edge that connects p_s and W .

After the first W is formed by one-ring neighbour vertices of p_s , the first vertex $p_i \in W$ is selected and its one-ring neighbour vertices exclud-

ing vertices that have been visited are stored into an array N' . With a vertex $pn_i \in N'$, the next step is to determine its parent node that pn_i is connected in order to form its initial path from p_s to pn_i . The algorithm presented in this chapter calculates geodesics by eliminating geodesic curvature at each sampling point on curves. When performing propagation outwards, each unvisited vertex needs to be connected to a visited vertex to form a new initial path for geodesic curvature flow calculation. For each visited vertex, the path from source vertex to this vertex is already a straightest geodesic path resulted by solving Equation 3.19. For each unvisited vertex, in order to form an initial path that is close to its “ideal” straightest geodesic path, the section from this unvisited vertex to its adjacent visited vertex should follows the direction of geodesic path from source to this visited vertex. Therefore, each unvisited vertex always selects a geodesic path which can form the straightest initial path within its one-ring-visited vertices. This process is illustrated in Figure 3.11.

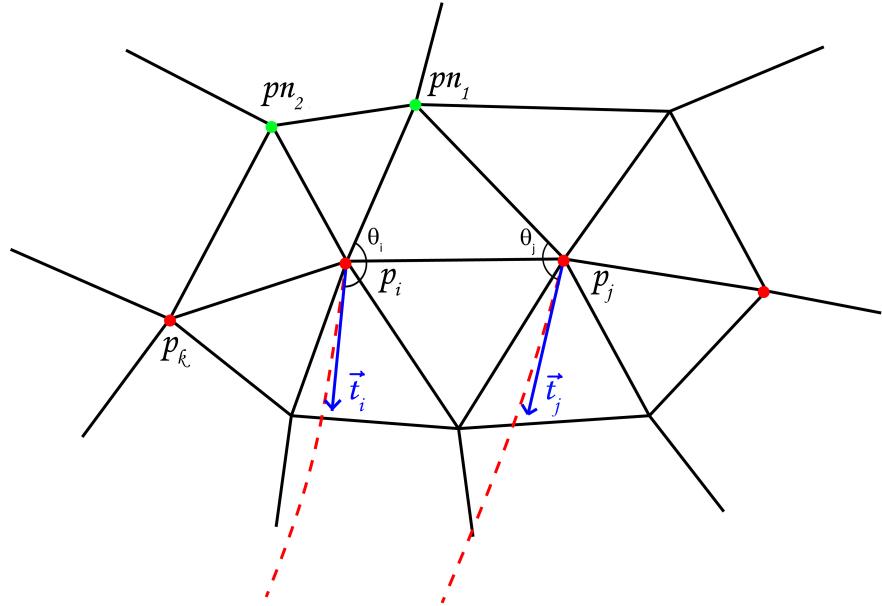


Figure 3.11: Calculating geodesic path for a unvisited vertex, red point p_i, p_j and p_k are visited vertex in wavefront W , red dash line indicates geodesic path from source to p_i and p_j , vector \vec{t}_i and \vec{t}_j is the tangent vector at p_i and p_j respectively, pn_1 and pn_2 are one-ring neighbour of p_i that has not been visited.

Where, pn_1 and pn_2 is in the one-ring neighbour N' of p_i that have not been visited. For a random vertex $pn_1 \in N'$, the visited vertices in one-ring neighbour of pn_i are selected which in the example demonstrated in Figure 3.11, p_i and p_j are selected. In the next step, for all the visited vertices that connect to pn_1 , the angle θ between \vec{t}_i and the edge that connects pn_1 and current selected visited vertex is calculated. \vec{t}_i denotes the tangent vector at current visited vertex on its geodesic path. The parent node of pn_1 is then selected from the visited vertex with the largest θ .

In Figure 3.11, the included angle θ_i between $\overline{p_ipn_1}$ and \vec{t}_i is greater than the included angle θ_j between $\overline{p_jpn_1}$ and \vec{v}_j . Therefore, it can be observed that, comparing two curves of edge $\overline{p_ipn_1}$ plus geodesic path at p_i and edge $\overline{p_jpn_1}$ plus geodesic path at p_j , the curve of edge $\overline{p_ipn_1}$ plus geodesic path at p_i are straighter than the curve of edge $\overline{p_ipn_1}$ plus geodesic path at p_i . Thus, vertex pn_1 is pushed into the end of the geodesic path at vertex p_i as the initial path for pn_1 . For every new point that is pushed to the end of its parent point's geodesic path, it contains the entire information of the parent vertex. As a result, the algorithm presented here does not require “backtracking” that are necessary for MMP, CH and FMM algorithms in order to retrieve the geodesic path. The aforementioned method propagates the geodesic information from the interior ring to the external one. However, without the help of priority queue, propagating the geodesic information from the vertex with smaller geodesic distances to the vertex with larger distances can not be guaranteed. Therefore, in order to maintain the isotropy of the propagation, a propagation radius limit r_{max} is employed. After pn_1 has selected p_i as its parent node, the estimate geodesic distance d_{est} from source to pn_1 can be approximated by the length of edge $\overline{pn_1p_i}$ plus geodesic distance from source to p_i . Then Equation 3.11 is performed to calculate the geodesic path from source to pn_1 only if d_{est} is smaller than the current r_{max} . After that, p_i is stored into an array W' as the “new wavefront”. After all the vertices in W has been popped out, W is replaced by W' .

However, if d_{est} is larger than the current r_{max} , then pn_1 is stored into an array W_{next} as the “next wavefront” for next propagation. For every unvisited vertex that connects to the visited vertex in W , only the largest d_{est} is kept as d_{maxEst} . When both W and W' are empty, the W is replaced by W_{next} and r_{max} is replaced by d_{maxEst} as the new propagation limit. By using this method, within this limitation r_{max} , the presented method still propagates information from the interior rings to external ones. The r_{max} forms an isometric line on the polyhedral surface that constrains the propagation at every step. Because r_{max} is updated every time when W is replaced by W_{next} , as a result, in between the updates of r_{max} , the length of all the geodesic paths are in between the r_{max} and d_{maxEst} . Therefore, the causality is guaranteed by r_{max} . This propagation algorithm is summarized in Algorithm 2.

When solving Equation 3.11, the updated points on geodesic path always move away from the vertices on initial path. Therefore, in Algorithm 2, in order to calculate the projections of the sample point p on floating geodesic path, examining every single face on the mesh is a very time consuming process. In order to maintain the efficiency of the presented algorithm, it is important to keep track on the corresponding vertex on the mesh where p is updated from. Based on the observation from our experiments, for every unvisited vertex, the straightest path is selected as the parent and the updated sample point on geodesic path never move out of their original one-ring neighbourhoods. However, after the “wavefront” propagates for more than once, this assumption does not stand any more. Thus one-ring neighbourhood of every sample point needs to be redetermined after the initial path is updated by Equation 3.11. This is achieved by the following procedure. Firstly, starting from one-ring neighbour vertices of a sample point q , the nearest vertex q_i is selected. Then within one-ring neighbours of q_i , the nearest vertex q_j to q is selected as the “new” initial vertex of point q . This updating process is able to ensure the projection of the updated sample point q on floating geodesic path always falls into the one-ring neighbour faces that is adjacent to q_j .

Algorithm 2 Accurate Geodesics Algorithm

```
1: procedure INITIALIZATION(A mesh  $S$ , and a source  $p_s$ )
2:   for  $p_i \in S$  do
3:      $p_i.geoDist \leftarrow \infty$ 
4:      $p_i.geoPath \leftarrow \emptyset$ 
5:   end for
6: end procedure

7: procedure CALCULATE ONE RING NEIGHBOUR OF  $p_s$ (  $W$  ,  $r_{max}$ )
8:   for  $p_i \in W$  do
9:      $d_{est} \leftarrow Dist(p_i, p_s)$ 
10:     $p_i.geoDist \leftarrow d_{est}$ 
11:     $p_i.geoPath \leftarrow p_s + p_i$ 
12:    if  $currentDist > r_{max}$  then
13:       $r_{max} \leftarrow d_{est}$ 
14:    end if
15:   end for
16: end procedure

17: function GETPARENTNODE( $W, v$ )
18:   for  $v' \in oneRingNeighbourOf(v)$  do
19:      $\theta = 0$ 
20:     if  $v' \in W$  then
21:        $\vec{v}_1 \leftarrow v - v'$ 
22:        $\vec{v}_2 \leftarrow getTangent(v)$             $\triangleright$  Return the tangent vector of
          geodesic path from source to  $v'$  at point  $v'$ 
23:       if  $\theta < angleBetween(\vec{v}_1, \vec{v}_2)$  then
24:          $\theta \leftarrow angleBetween(\vec{v}_1, \vec{v}_2)$ 
25:          $parentNode \leftarrow v'$ 
26:       end if
27:     end if
28:   end for
29:   return  $parentNode$ 
30: end function
```

Algorithm 2 accurate geodesics algorithm (continued)

```
31: procedure CONTINUES DIJKSTRA PROPAGATION( Calculate geodesic
   for every  $p_i \in S$ )
32:   while  $W \neq \emptyset$  do
33:     for  $p_i \in W$  do
34:       for  $pc_j \in \text{oneRingNeighbourOf}(p_i)$  do
35:         if  $pc_j \in W$  OR  $pc_j.\text{geoDist} \neq \infty$  then
36:           continue
37:         else
38:            $\text{parentNode} \leftarrow \text{GETPARENTNODE}(W, pc_j)$ 
39:            $d_{est} \leftarrow \text{distanceBetween}(\text{parentNode}, pc_j) +$ 
                $\text{parentNode}.\text{geoDist}$ 
40:           if  $d_{est} < r_{max}$  then
41:             Perform Equation 3.11 on  $\text{initialPath}$  re-
               sults geodesic path from source to  $pc_j$ 
42:             Perform Algorithm 1 to project floating path
                $pc_j.\text{geoPath}$  onto  $S$ 
43:             Update the neighbourhoods for the samples
               point on  $pc_j.\text{geoPath}$  separately
44:              $W' \leftarrow pc_j$ 
45:           else
46:             Update  $d_{maxEst}$  if  $d_{est} > d_{maxEst}$ 
47:              $W_{next} \leftarrow pc_j$ 
48:           end if
49:         end if
50:       end for
51:     end for
52:      $W \leftarrow W'$ 
53:      $W' \leftarrow \emptyset$ 
54:     if  $W = \emptyset$  then
55:        $W \leftarrow W_{next}$ 
56:        $W_{next} \leftarrow \emptyset$ 
57:        $r_{max} \leftarrow d_{maxEst}$ 
58:     end if
59:   end while
60: end procedure
```

During the projection, the osculating plane is formed by three successive sample points on a geodesic. This osculating plane is used to cut the related faces which are one-ring neighbours of these three sample points. The intersection of these two planes is the projection of the geodesic on the mesh. Note that with a manifold mesh, an edge can only be connected with one or two faces, therefore, with a face that has been determined to intersect with the osculating plane, its adjacent face can easily be determined. As a result, calculating the intersections between the cutting plane and all the related edges costs a constant time.

A challenging issue of using Algorithm 2 to calculate geodesic is to solve large sparse linear system of Equation 3.18. Moreover, the size of matrix in Equation 3.18 is determined by the number of sample points on a path. Therefore, followed by the propagation of “wavefront”, the size of matrix in Equation 3.18 increases. This increment on the matrix size of Equation 3.18 results in very expensive computational cost for calculating geodesics on a mesh. In fact, solving Equation 3.18 is the most time consuming process in the entire geodesic computation.

In order to improve the efficiency of Algorithm 2, an approximate algorithm is derived from it. In this algorithm, a small-size window is introduced to Equation 3.11 to avoid solving large sparse linear system. When an initial estimation of a geodesic path is obtained, as shown in Figure 3.11, a fixed-size window that covers the last w sample points is employed. Then Equation 3.11 is applied to w to calculate a local geodesic patch. Let $\widetilde{p'_0 p'_{n-1}}$ denotes the geodesic path from source to the parent of p' parent. Thus, only the section $\widetilde{p'_{n-w} p'_n}$ is updated to form the approximated geodesic path. This algorithm is summarized in Algorithm 3.

In Algorithm 3, window size w is a constant number, therefore, the matrix size in the linear system of Equation 3.11 is also constant throughout the propagation. Because for each unvisited vertex on mesh, the time used for

Algorithm 3 Approximate Geodesics Algorithm

```
1: procedure INITIALIZATION(A mesh  $S$ , and a source  $p_s$ )
2:   for  $p_i \in S$  do
3:      $p_i.geoDist \leftarrow \infty$ 
4:      $p_i.geoPath \leftarrow \emptyset$ 
5:   end for
6: end procedure

7: procedure CALCULATE ONE RING NEIGHBOUR OF  $p_s(W, r_{max})$ 
8:   for  $p_i \in W$  do
9:      $d_{est} \leftarrow Dist(p_i, p_s)$ 
10:     $p_i.geoDist \leftarrow currentDist$ 
11:     $p_i.geoPath \leftarrow p_s + p_i$ 
12:    if  $currentDist > r_{max}$  then
13:       $r_{max} \leftarrow d_{est}$ 
14:    end if
15:   end for
16: end procedure

17: function GETPARENTNODE( $W, v_{unvisited}$ ) ( $W$  denotes propagation boundary)
18:   for  $v' \in oneRingNeighbourOf(v_{unvisited})$  do
19:      $\theta = 0$ 
20:     if  $v' \in W$  then
21:        $\vec{v}_1 \leftarrow v_{unvisited} - v'$ 
22:        $\vec{v}_2 \leftarrow getTangent(v_{unvisited})$   $\triangleright$  Return the tangent vector of geodesic path from source to  $v'$  at point  $v'$ 
23:       if  $\theta < angleBetween(\vec{v}_1, \vec{v}_2)$  then
24:          $\theta \leftarrow angleBetween(\vec{v}_1, \vec{v}_2)$ 
25:          $parentNode \leftarrow v'$ 
26:       end if
27:     end if
28:   end for
29:   return  $parentNode$ 
30: end function
```

Algorithm 3 Approximate Geodesics Algorithm (continued)

31: **procedure** CONTINUES DIJKSTRA PROPAGATION(Calculate geodesic
for every $p_i \in S$)
32: **while** $W \neq \emptyset$ **do**
33: **for** $p_i \in W$ **do**
34: **for** $pc_i \in \text{oneRingNeighbourOf}(p_i)$ **do**
35: **if** $pc_j \in W$ **OR** $pc_j.\text{geoDist} \neq \infty$ **then**
36: **continue**
37: **else**
38: $\text{parentNode} \leftarrow \text{GETPARENTNODE}(W, pc_j)$
39: $d_{\text{est}} \leftarrow = \text{distanceBetween}(\text{parentNode}, pc_j) +$
 $\text{parentNode}.\text{geoDist}$
40: **if** $d_{\text{est}} < r_{\max}$ **then**
41: **Perform** Equation 3.11 on the $\widetilde{p_{n-w}p_n}$ part of
 initialPath results geodesic path from p_{n-w}
 to pc_j
42: **Combine** $\widetilde{p_0p_{n-w-1}}$ with $\widetilde{p_{n-w}pc_j}$ to form the
 approximate geodesic from source to pc_j
43: **Perform** Algorithm 1 to project floating path
 $pc_j.\text{geoPath}$ onto S
44: **Update** the neighbourhoods for the samples
 point on $pc_j.\text{geoPath}$ from $n - w^{th}$ to n^{th}
 point separately
45: $W' \leftarrow pc_j$
46: **else**
47: Update d_{\maxEst} if $d_{\text{est}} > d_{\maxEst}$
48: $W_{\text{next}} \leftarrow \text{parentNode}$
49: **end if**
50: **end if**
51: **end for**
52: **end for**
53: $W \leftarrow W'$
54: $W' \leftarrow \emptyset$
55: **if** $W = \emptyset$ **then**
56: $W \leftarrow W_{\text{next}}$
57: $W_{\text{next}} \leftarrow \emptyset$
58: $r_{\max} \leftarrow d_{\maxEst}$
59: **end if**
60: **end while**
61: **end procedure**

solving Equation 3.11 is constant and the time used for calculating geodesic is also same. Therefore, by introducing the concept of window to each geodesic update, Algorithm 3 is able to achieve linear time complexity. However, when window size is fixed, only a section of the initial path is updated by Equation 3.11 which means the geodesic curvature at points fall into this section is eliminated, the rest sample point on initial path are not updated. Combining the updated section with the rest initial path resulting a path which is not a true geodesic path but an approximation of the “true” geodesic path. Moreover, larger the w is set, closer the resulting path is to the true geodesic path and higher the accuracy of approximation is but longer the computation for solving Equation 3.11 is. Hence, in the experiment section of this chapter, an experiment for using different window sizes on a model is designed to illustrate the relationship between window size and computational cost as well as the relationship between window size and approximation accuracy.

In real world, when using tape ruler to measure the length of a body part, it follows the profile of body in a convex-hull like manner that it only makes contact with subject body on the convex part. This is demonstrated in Figure 3.12.



Figure 3.12: Tape measuring on human body

The shape of upper leg is a convex shape, therefore, the tape ruler follows its profile till knee cap. Because the knee cap is much higher than the surface of lower leg, the lower section of leg between ankle to knee cap is a concave shape, therefore, tape ruler from knee to ankle forms a straight line.

However, geodesic path only travels on curved surface no matter the shape of surface is concave or convex, therefore, a difference exists in the measurement data acquired by tape ruler measuring and the one acquired by geodesic path measuring. This difference is become larger on muscular character whose body surface is more uneven than a skinny character.

In order to improve the measuring quality of the presented geodesic path measuring method, a path correction procedure is designed to straighten the section of a geodesic path where it travels through a concave part of character body. Firstly, all the vertices normal of character model are set outward first. Because at each sample point on a geodesic path, the geodesic curvature is eliminated by Equation 3.19, the normal vector at each sample point collinear to the normal vector of surface. Based on the direction of the normal vector at a sample point on a geodesic path and its corresponding surface normal vector. A geodesic path can be segmented into several sections. This is shown in Figure 3.13.

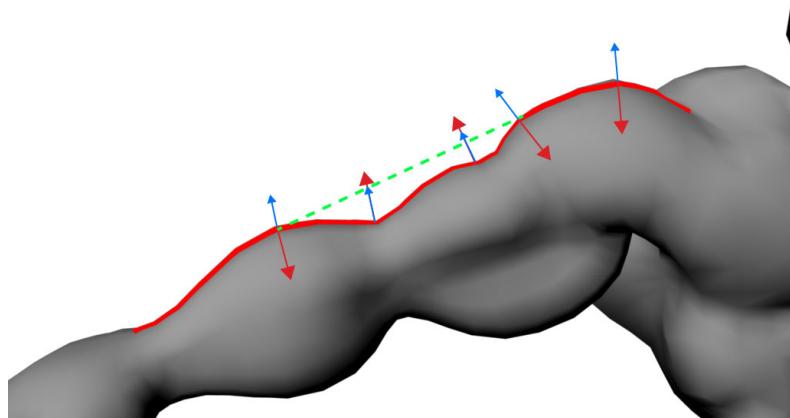


Figure 3.13: Curve normal and vertex normal on a geodesic path on the arm of character A

In, Figure 3.13 red path is the geodesic path used for measuring arm length. Red arrows demonstrate curve normal at sample points on path, blue arrows indicate vertex normal on the character model. Green dashline indicates the path where tape ruler would went through based on our observation. Because red path is a geodesic path, at each sample point, its curve normal collinear with the vertex normal at corresponding location on mesh. Note that, on the convex part of mesh, the curve normal and vertex normal of mesh are in opposite direction, whilst on the concave part of mesh, the curve normal and vertex normal of mesh are in the same direction. The presented path shape correction procedure iteratively moves each sampling point which is in the same direction as it corresponding vertex normal to its curve normal direction till all the sample points on a curve either in opposite direction with its vertex normal or has zero curvature. Here, defining concave point as a sample point on a geodesic path whose curve normal is in the same direction with its vertex normal on mesh. The geodesic path shape correction procedure is illustrated in Figure 3.14.

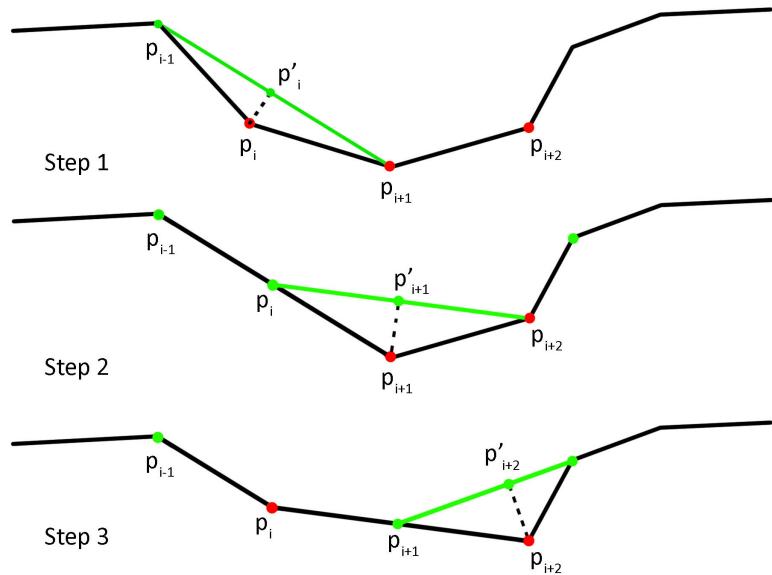


Figure 3.14: Demonstration of path correction process

Where black piece-wise curve is a geodesic path on a character, green

point indicates a non-concave point, red point indicates a concave point. The procedure works in following manner, firstly, select a concave point p_i . Then project p_i onto a line defined by two adjacent points of p_i , where in Figure 3.14 are p_{i-1} and p_{i+1} . Update p_i by its projection p'_i . Select next concave point p_{i+1} , project p_i onto a line defined by p_i and p_{i+2} . Update p_{i+1} by its projection p'_{i+1} . Continue this process till all concave points are become non-concave. Figure 3.15 demonstrates the result of path shape correction process on Character A.

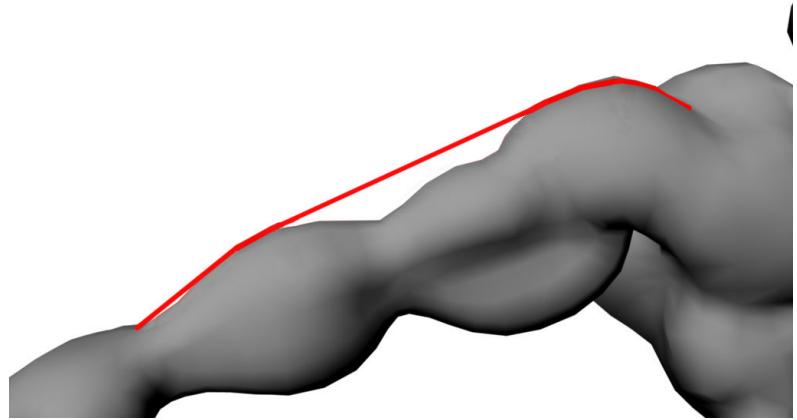


Figure 3.15: A straightened path for measuring arm length.

Moreover, for characters with bent arm where the geodesic path goes through inside of elbow, shown in Figure 3.16, perform this shape correction process will straighten the entire geodesic path. In order to force the corrected path still follows the trend of a bent arm, user can select a section of on a geodesic path where this correction procedure will ignores, such as area around elbow, therefore, at the selected area, the path still follows the shape of character model meanwhile the concave part in the rest of geodesic path will be straightened, Figure 3.17 demonstrates the result on a muscular character.

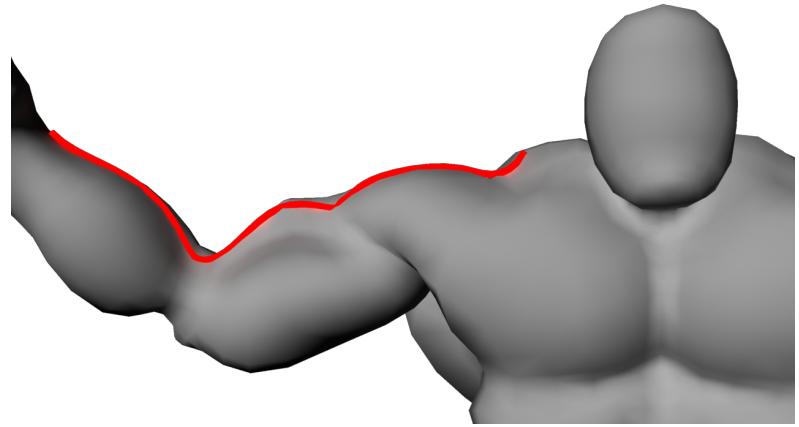


Figure 3.16: Geodesic path on a bent arm of Character A

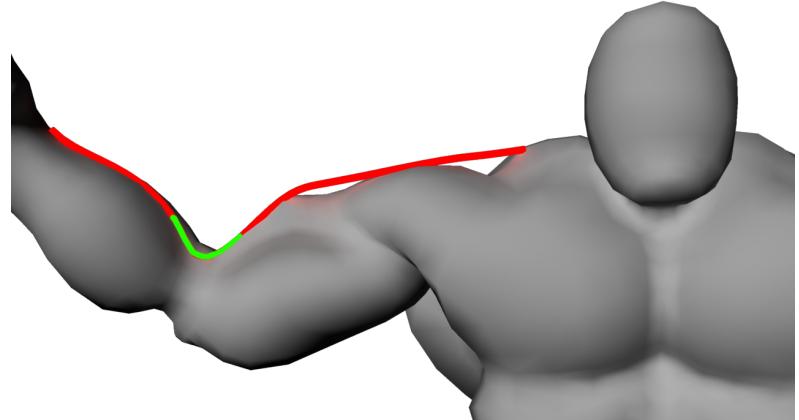


Figure 3.17: Shape corrected path with a section selected by user (green section) to be ignored by shape correction process

3.4 Performance Analysis

The performance of the proposed algorithms is analysed in two categories, the accuracy of the geodesics and the efficiency of the algorithm. Assume the given mesh contains n vertices, e edges and f faces.

3.4.1 Efficiency

The time complexity of Algorithm 2 depends on the number of the sample points on a geodesic. Because the coefficient matrix A on the left hand side of Equation 3.18 is not a square matrix, it is necessary to left-multiply A^T on both side of the Equation 3.18 to form the square matrix for linear system solver. In the implementation of this algorithm, LU solver(Bunch & Hopcroft 1974) is used for solving the linear system. Let m denotes the order of the matrix in Equation 3.18, according to Bunch & Hopcroft (1974); Coppersmith & Winograd (1987), the LU solver has a time complexity of $O(m^{2.379})$. Therefore, for a single source to all destination geodesic computation, the upper bound of the time complexity can be estimated by $O(m^{2.379}n)$, where n denotes the number of vertices on mesh. Additionally, computation of the projection for each three successive sample points of a geodesic onto a mesh costs a constant time, denotes at h , the projection of the entire geodesic costs $O(kh)$ where k denotes the number of sample points on a geodesic path. Consequently, projecting all the geodesics onto mesh costs $O(khm^{2.379}n)$.

For Algorithm 3, due to the fixed size window, the size of matrices in Equation 3.18 is constant. Therefore, solving Equation 3.18 costs a constant time c . The total time complexity can be written as $O(cn)$. Because the window size is w , a new geodesic usually shares a segment with an existing geodesic. The projection of a geodesic path also shares a segment with an existing geodesic projection on a mesh. Projecting one geodesic path therefore costs $O(wh)$, and projecting all the geodesics costs $O(nwh)$, where w and h are constants. This conclusion is important, as it shows Algorithm 3 is a linear-time algorithm.

Among the above presented algorithms, the most noteworthy achievement is the ability of computing geodesic paths in a linear time in Algorithm 3 with a bounded error. This is especially significant for tracing a large number

of geodesic paths on high resolution models with over one million vertices, which is more and more common in various applications due to the technological advancements in high performance hardware and cheap storage. Larger models offer much better resolution and more detailed structural information, making many previously impossible operations possible today.

The aforementioned algorithms is implemented on an Intel Xeon 3.33GHz PC with 24GB RAM running Windows 7 (64-bit) operating system. For time comparison, several popular existing algorithms are used for comparison including MMP (Surazhsky et al. 2005), Improved CH algorithms (Xin & Wang 2009) and FMM (Kimmel & Sethian 1998). In order to compare the time consumption for acquiring geodesic path, “backtracing” is added into MMP, CH/ICH and FMM. The algorithm used for “backtracing” is presented in Surazhsky et al. (2005).

The source codes of MMP and ICH algorithms are available on the author’s project page¹. These algorithms are performed respectively on the Stanford Bunny’s model with 8 different resolutions as shown in Table 3.2. The running times are plotted in Figure 3.18. It can be seen that the Algorithm 3 shows great advantage over others when the resolution of the model increases.

¹MMP at http://code.google.com/p/geodesic/downloads/detail?name=geodesic_cpp_03_02_2008.zip&can=2&q=; Improved CH at <https://sites.google.com/site/xinshiqing/knowledge-share>

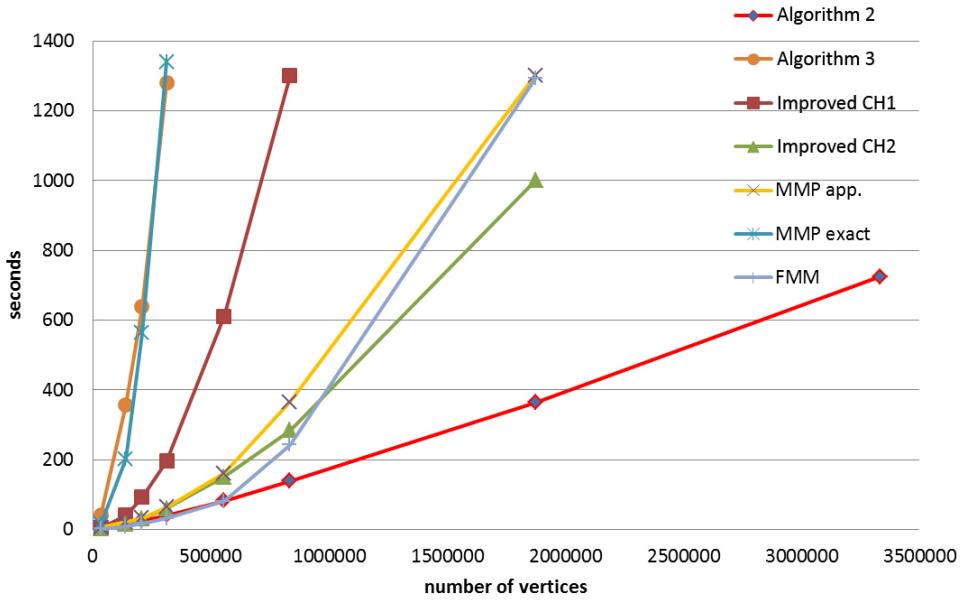


Figure 3.18: The comparison of running time. The time of the ICH and MMP algorithms includes the cost of “backtracing”, for Algorithm 3, $w=30$.

#Vertices	34834	139122	208573	312861
#Faces	69451	277804	416706	625059
#Vertices	556051	833855	1875843	3334537
#Faces	1111216	1666824	3750354	6667296

Table 3.2: The resolution of the bunny models used in the experiment in Figure 3.18

3.4.2 Accuracy

On a non-convex surface, geodesic path between two points is not unique (Thielhelm et al. 2012). In order to set up a ground truth for comparing geodesic calculation accuracy. The experiment is carried out on a sphere. This is due to all great-circles between two points on a sphere are geodesics and geodesic distance between them is the length of a great-circle, this distance can also be called as orthodromic distance. The orthodromic distance

between two points on a sphere can be calculated using Equation 3.21 (Aleksandrov & Zalgaller 1967).

$$d = r \times \arccos(\vec{n}_1 \cdot \vec{n}_2) \quad (3.21)$$

where r denotes the radius of the sphere, \vec{n}_1 and \vec{n}_2 are to normals at two points on the sphere.

Both MMP algorithm(Mitchell et al. 1987b; Surazhsky et al. 2005) and Algorithm 2 are performed on a unit sphere and the length of each geodesic path is compared against the orthodromic distance between the source point and destination point. Figure 3.19 illustrates geodesics from one vertex to all the other vertices on a unit sphere.

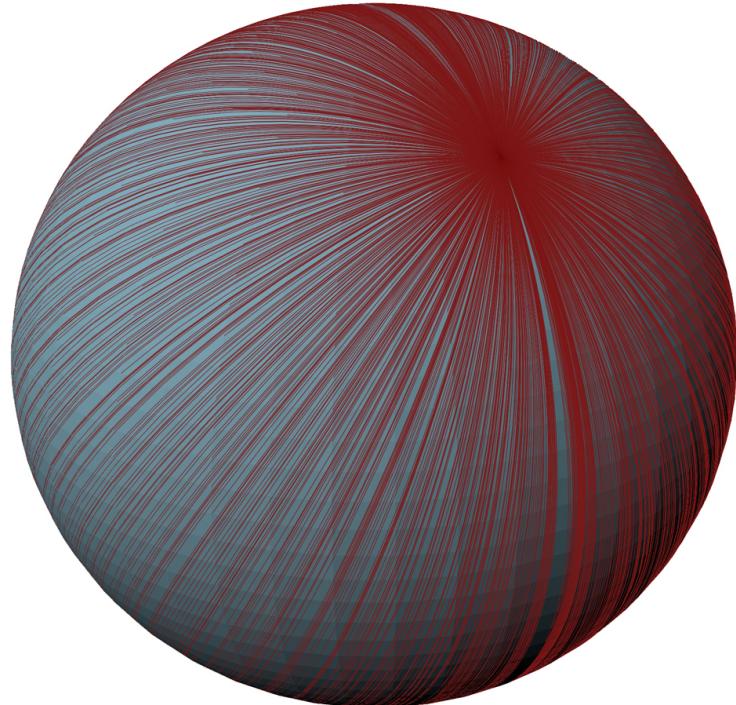


Figure 3.19: Geodesics on a sphere.

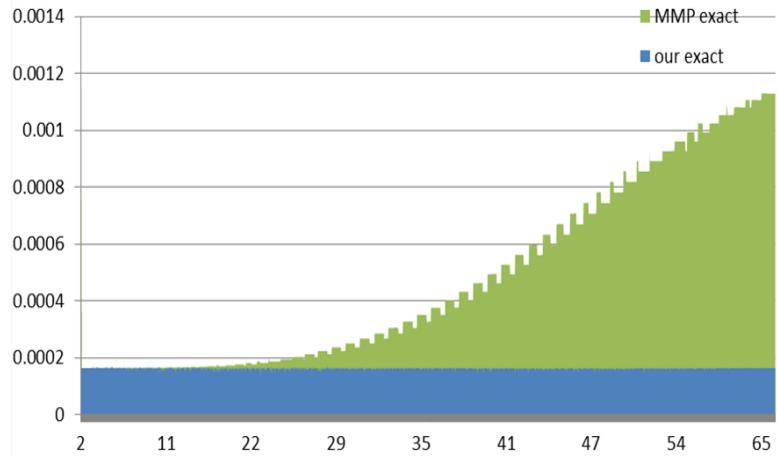


Figure 3.20: Distribution of the relative errors of MMP and Algorithm 2

Figure 3.20 demonstrates the relative error of MMP algorithm(Mitchell et al. 1987b; Surazhsky et al. 2005) and Algorithm 2. Moreover, this experiment is also performed to examine the accuracy of MMP approximate algorithm and Algorithm 3, the result is shown in Figure 3.21.

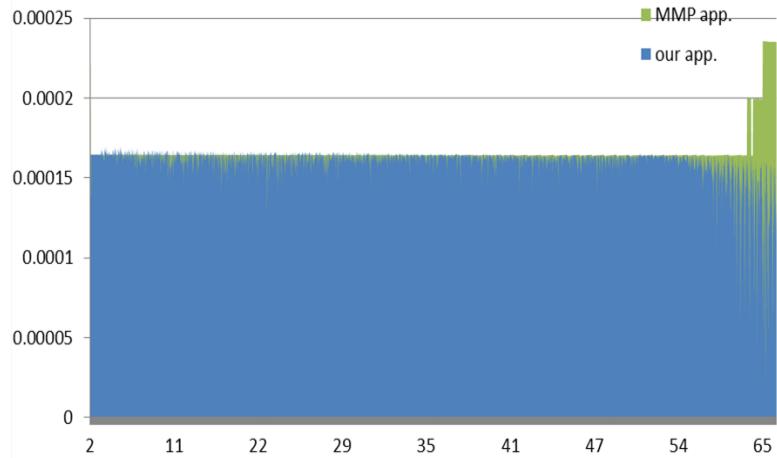


Figure 3.21: Distribution of the relative errors of MMP approximate algorithm and Algorithm 3

The accuracy of Algorithm 3 depends on the following parameters, the edge length e , the window size w , and the number of sample points on each geodesic. Let m denotes the number of sample points on a geodesic. The basic assumption held by Algorithm 3 is that when computing a new geodesic,

there exists one of the previous geodesics that is accurate enough and close to the desired one. As a result, we may crop a small patch at the end of the geodesic estimation by a w -sized window and apply Equation 3.11 to this patch instead of the whole path.

Let C_g denotes a geodesic path on a polyhedral surface. C_g does not pass through any vertex p of the polyhedron unless p is the source point or the destination. It can be observed that the C_g goes across a set of faces. Unfolding this set of faces into a plane, C_g should becomes a straight line linking the source point p_s to the destination point p_d . For a new vertex q , the geodesic $\widetilde{p_s q}$ can be estimated by combining the edge $\overline{p_d q}$ with C_g . Without losing the generality, let a window covers a section of $\widetilde{p_s q}$ from q , w denotes the number of the sample points covered by this window. The error estimation for any geodesic computed by Algorithm 3 is illustrated in Figure 3.22,

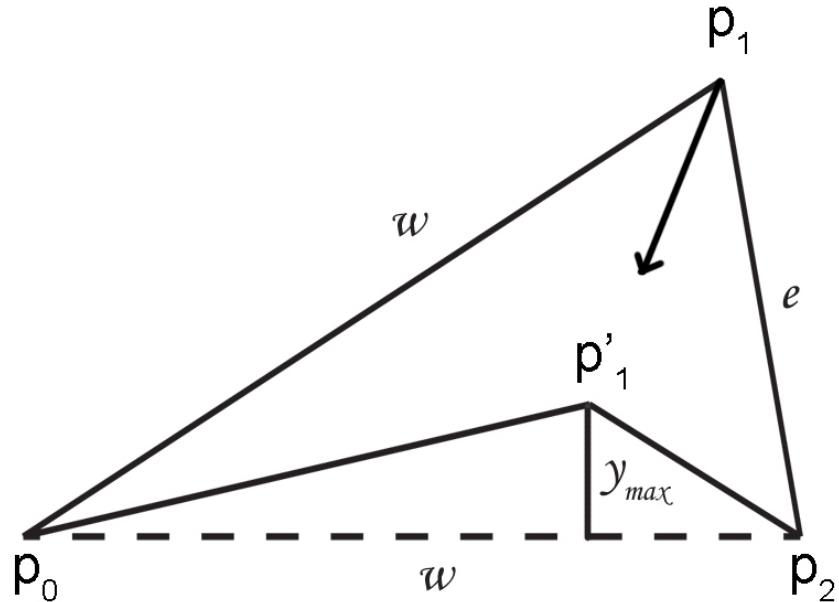


Figure 3.22: Error estimation of Algorithm 3, the $\overline{p_0 p_2}$ is the desired geodesic while $\overline{p_0 p_1}$ plus $\overline{p_1 p_2}$ being the estimation.

Let p_0, p_1, p_2 are three points on a plane as shown in Figure 3.22. The initial path for solving Equation 3.11 is $\widetilde{p_0 p_1 p_2}$ where p_0 is the source point

and p_2 is the destination point. Obviously, the true geodesic from p_0 to p_2 is $\overline{p_0p_2}$. Let $\overline{p_0p_1}$ be an established geodesic path, p_2 is an unvisited vertex. The length of edge $\overline{p_0p_1}$ is always smaller than or equals to the length of $\overline{p_0p_2}$ because the propagation only performs on outward direction. Therefore, in the worst situation, which is demonstrated in Figure 3.22, $\overline{p_0p_2}$ and $\overline{p_0p_1}$ are equal. p'_1 is the updated point calculated by Equation 3.11, Let the distance from p'_1 to edge $\overline{p_0p_2}$ is y_{max} , $\|p_1p_2\| = e$, $\|p_0p_1\| = \|p_0p_2\| = w$, $\|p_0p'_1\| + \|p'_1p_2\| = L$. Therefore, based on Heron's formula (Aleksandrov & Zalgaller 1967), the area of $\triangle p_0p'_1p_2$ can be written as,

$$T_{\triangle p_0p'_1p_2} = \frac{\sqrt{(L^2 - w^2)w^2}}{4} = \frac{wy_{max}}{2} \quad (3.22)$$

therefore,

$$L = \sqrt{4y_{max}^2 + w^2} \quad (3.23)$$

Let err denotes the difference between true geodesic path length and the solution of Equation 3.11, where err can be written as,

$$err = L - \|p_0p_2\| = \sqrt{4y_{max}^2 + w^2} - w \quad (3.24)$$

The average error per-window over the w -sized window can be estimated by,

$$err_{ave} = \frac{\sqrt{4y_{max}^2 + w^2} - w}{w} \quad (3.25)$$

For a geodesic with m sample points, the upper bound of the error is

therefore estimated by,

$$err = m \left(\sqrt{1 + 4(\frac{y_{max}}{w})^2} - 1 \right) < 2m \frac{y_{max}}{w} \sim O(\varepsilon m) \quad (3.26)$$

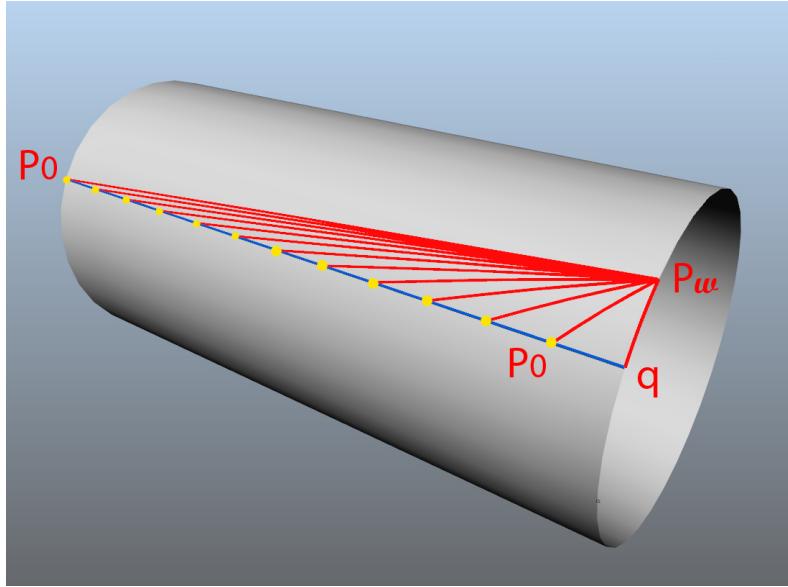


Figure 3.23: Error estimation of the approximate algorithm with different window size

where $\varepsilon = \frac{y_{max}}{w} \ll 1$ and y_{max} denotes the maximum offset distance of sample points to the ground truth. To have an insight into y_{max} , Algorithm 3 is performed on a cylindrical surface, see Figure 3.23. $\overline{p_0q}$ is parallel to the axis of the cylindrical surface, and the true geodesic between p_0 and q is $\overline{p_0q}$. Let the length of the edge between each yellow point is $e = 1$ and the window size w varying from 5 to 100, The numerical results are shown in Figure 3.24a. It can be noted that when $w = 5$, the y_{max} tends to be zero. But, the larger the window, the bigger the y_{max} . This can be explained that on the plane shown in Figure 3.22, the curvature of the geodesic estimation $\widetilde{p_0p_1p_2}$ is becoming smaller when window size w is increased. Equation 3.11 is appropriate to deal with high curvature areas. On the other hand, the window size w is

usually expected to be as large as possible. In Figure 3.23, the p_0 is viewed as the start point of the window. If it is not the source, there must exist a geodesic path from source to p_0 . It is ideal that the desired geodesic from source to p_2 passes through the geodesic from source to p_0 . Thus, it is natural to enlarge the window size as much as possible. Figure 3.24a and Figure 3.24b shows that both the y_{max} and ratio ϵ converge to small values. This means that enlarge w does not reduce the error significantly. The choice of the window size w should take into account the time of solving Equation 3.11 rather than the computational error.

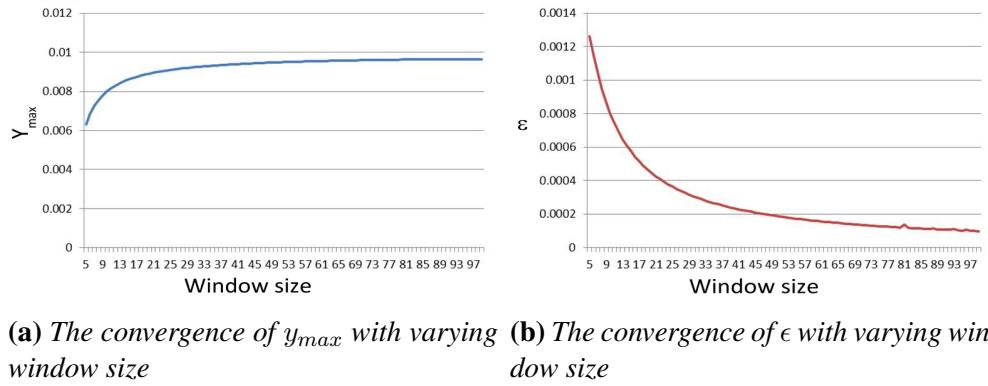
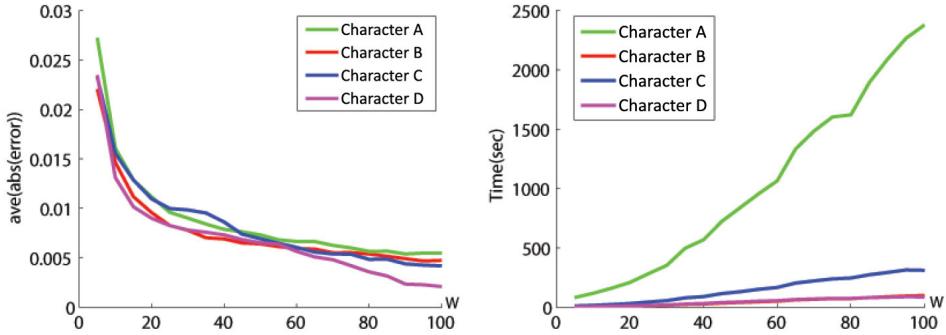


Figure 3.24: The convergenency of error to the window size of the approximate algorithm

In order to investigated how the running time and accuracy of Algorithm 3 vary with the window size w on a more complex model, MMP algorithm is performed on four character models to obtain the ground truth. The window size w is varied between 5 and 100. Experimental results on four character models are reported in Figure 3.25.



(a) Average absolute errors on four models. **(b)** Running times on the same four models.

Figure 3.25: Performance of Algorithm 3 with varying window size.

According to the results, a general trend is that the approximation error decreases with an increasing window size. It drops quickly when the window size is small. However, when the window size reaches 20, the speed of improvement in accuracy starts to slow down as shown in Figure 3.25a. Meanwhile, the running time starts to increase more rapidly, as shown in Figure 3.25b. Therefore, in the character measurements extraction process, the window size w is empirically set to 30.

Further to evaluate the accuracy of Algorithm 3 the MMP approximation and ICH_2 algorithms are also performed on four characters. MMP exact algorithm is performed on these four characters and its solutions are considered as the ground truth for this experiment. The absolute errors is the differences between exact geodesic distances and approximate ones and the relative errors is the ratios of absolute errors over exact distances.

Characters		A #V:127490	B #V:367942	C #V:365476	D #V:356544
		#F:254976	#F:735656	#F:730760	#F:712992
Algorithm 3 (floating)	ave abs	0.00309	0.00098	0.00786	0.005218
	ave rel	0.5054%	0.1911%	1.3454%	0.4852%
Algorithm 3 (projected)	ave abs	0.00401	0.00759	0.00716	0.01095
	ave rel	1.2298%	1.0439%	1.3053%	1.4867%
MMP app.	ave abs	0.00416	0.00991	0.00752	0.00917
	ave rel	1.4183%	1.3665%	1.3283%	1.0886%
ICH_2	ave abs	0.00344	0.003751	0.00795	0.01198
	ave rel	0.7867%	0.5886%	1.3311%	1.5128%

Table 3.3: The average absolute and relative errors of Algorithm 3, MMP app. and ICH_2. For Algorithm 3, $w = 30$.

Note that, for Algorithm 3, two results are kept as one is the offset geodesics and the other is the projections of the offset geodesics onto the mesh. Table 3.3 gives the average absolute and average relative error of these three algorithms.

Moreover, in order to further validate the error estimation for Algorithm 3, MMP exact algorithm is performed on the lowest resolution bunny model in Table 3.2 and length of the geodesics result from MMP exact algorithm is used as the ground truth.

The ratio ε in Equation 3.26 is usually a very small number. Figure 3.26 shows the histogram of the ratio ε over the bunny model, where horizontal axis indicates the value of ε and vertical axis indicates the number of sample points in a geodesic. Figure 3.26 indicates that on the bunny model, for all geodesics, ε remains a small value.

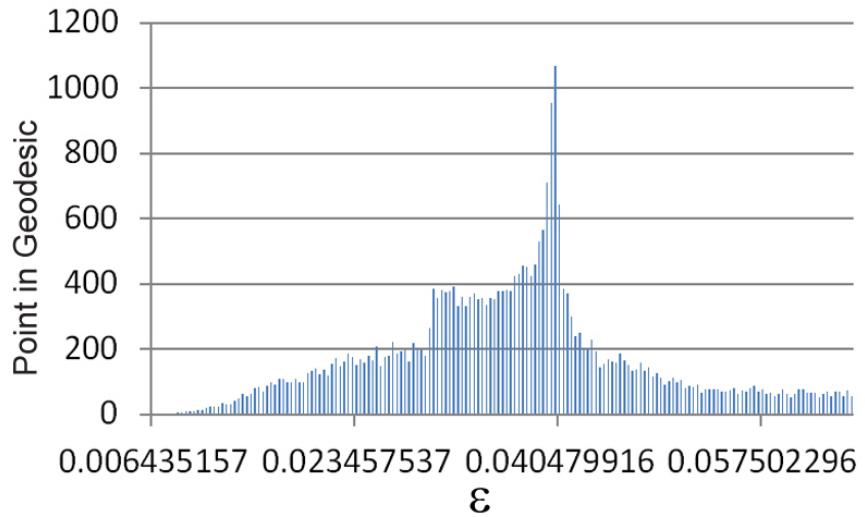


Figure 3.26: Histogram of the ratio ε on the bunny model

Figure 3.27 further shows the distribution of real absolute errors and the estimated ones in the bunny model test. It can be noted that the error estimation of Equation 3.26 can accurately reflects the upper error bound of Algorithm 3

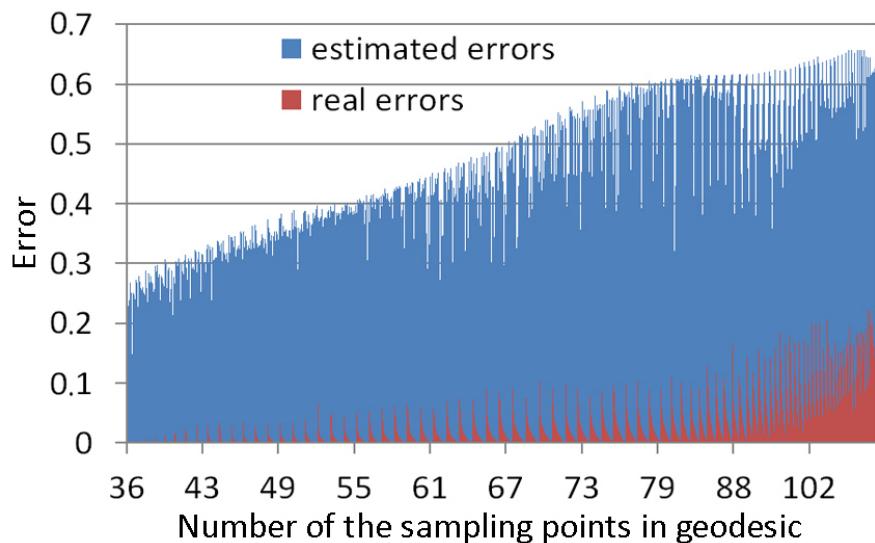


Figure 3.27: The distribution of estimated errors and real errors, the window size of Algorithm 3 is set to 30, the real error is bounded by the estimated error for all the geodesics on the bunny model.

3.5 Results and Conclusions

The proposed measurement method is performed on four different characters in the same hardware environment where experiments in previous section are carried out. For length measurements extraction, Algorithm 3 is utilised for geodesic path calculation. Also, MMP and ICH2 are used in order to compare the measuring efficiency. The experimental results are shown in Table 3.4.

	Character A #V:127490 #F:254976	Character B #V:367942 #F:735656	Character C #V:365476 #F:730760	Character D #V:356544 #F:712992
Algorithm 3	63.59s	194.13s	196.94s	191.55s
MMP	69.78s	479.03s	481.61s	471.10
ICH_2	65.52s	465.19s	466.84s	461.91s

Table 3.4: Measuring time on different characters.

Figure 3.28 to 3.32 visualize resulting measurements on four characters. Figure 3.29 demonstrates length measurements with geodesic path shape correction. In these figures, red curves indicate geodesic path used for extracting length measurements, blue curves indicate convex-hull used for extracting circumference measurements. Table 3.5 to 3.8 lists out resulting measurement data.

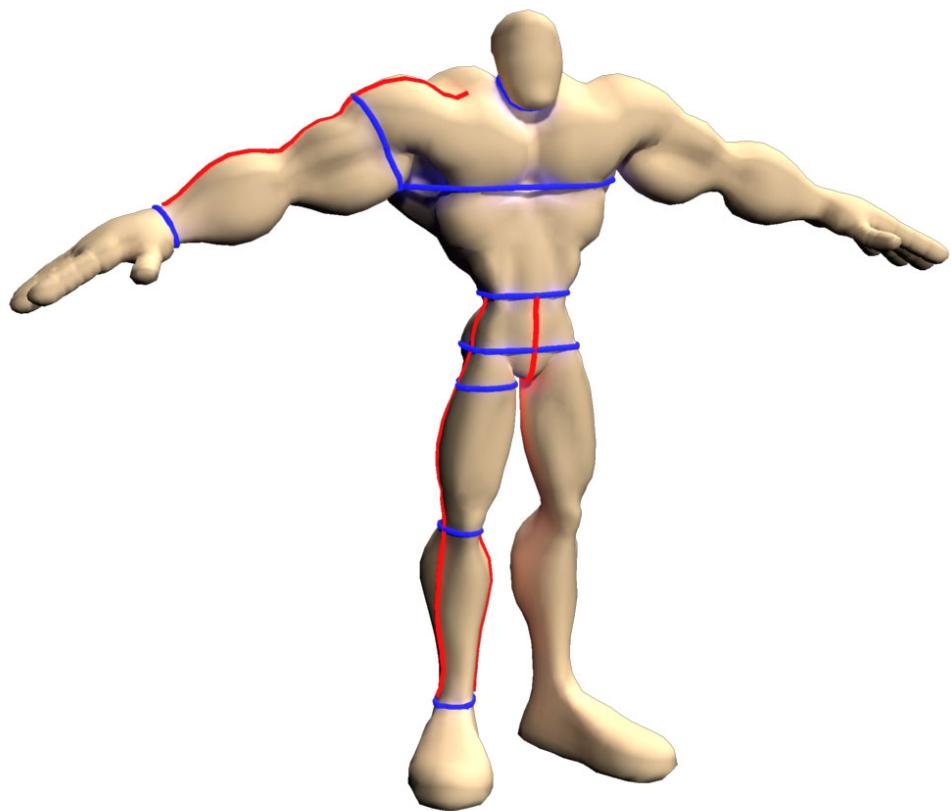


Figure 3.28: Measurements on Character A

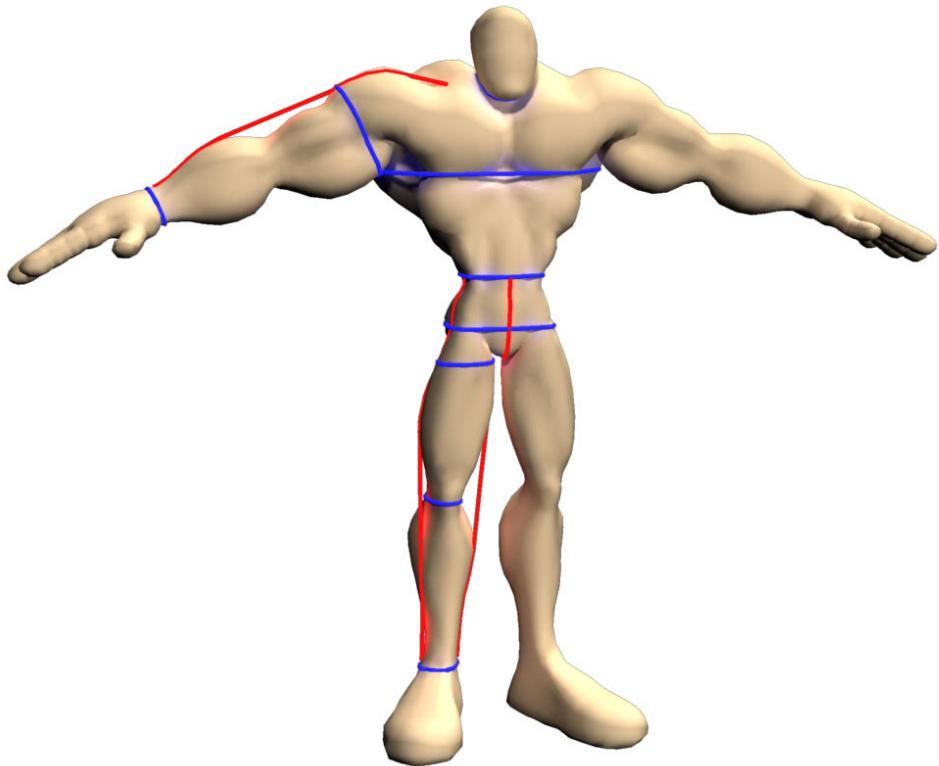


Figure 3.29: Measurements with geodesic shape correction on Character A

Bust Girth	211.9015	Sleeve length	122.8928
Waist Girth	73.8146	Sleeve length (with correction)	120.1646
Hip Girth	83.2698	Arm hole	157.3058
Neck Girth	67.3815	Back length	77.6125
Cuff Girth	59.3813	Outside Leg length	129.1429
Shoulder width	89.9452	Outside Leg length (with correction)	126.2821
Back width	74.8823	Inside Leg length	109.7143
Front width	88.7650	Inside Leg length (with correction)	107.2786
Ankle Girth	47.4285	Thigh Girth	65.5714

Table 3.5: Measurements for characters A.

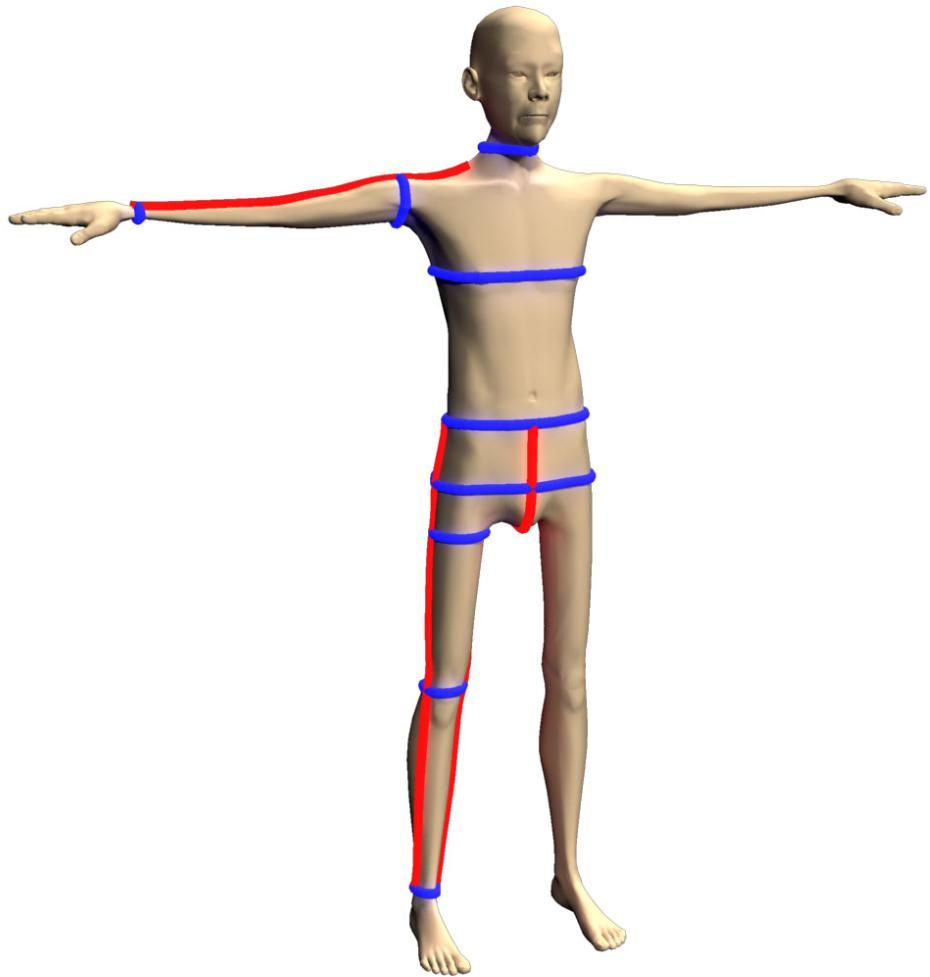


Figure 3.30: Measurements on Character B

Bust Girth	79.0833	Sleeve length	57.3615
Waist Girth	70.2424	Arm hole	29.2281
Hip Girth	87.2813	Back length	44.3503
Neck Girth	43.6637	Outside Leg length	89.1428
Cuff Girth	8.3771	Inside Leg length	74.2857
Shoulder width	40.0348	Thigh Girth	33.7142
Back width	38.6953	Ankle Girth	16.5741
Front width	45.7562		

Table 3.6: Measurements for characters B.

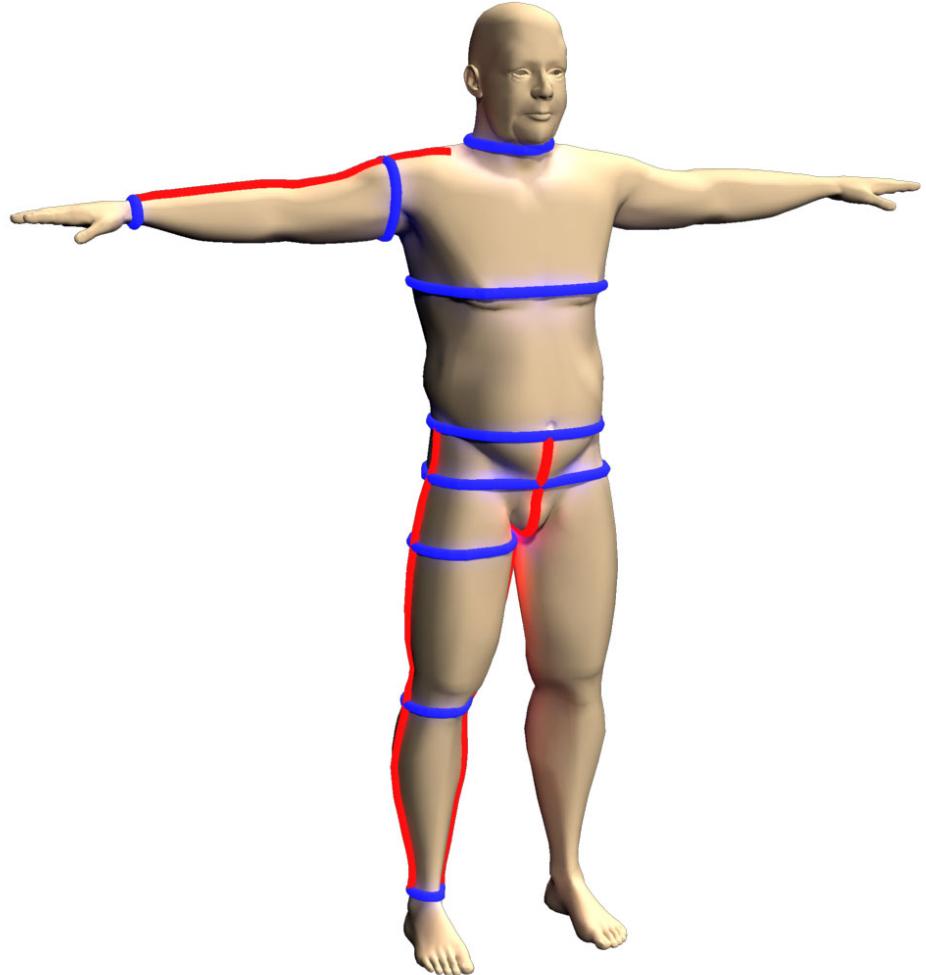


Figure 3.31: Measurements on Character C

Bust Girth	111.9165	Sleeve length	58.3748
Waist Girth	96.6514	Arm hole	42.9157
Hip Girth	107.5671	Back length	45.1839
Neck Girth	68.2416	Outside Leg length	85.0919
Cuff Girth	15.3368	Inside Leg length	71.4285
Shoulder width	51.3362	Thigh Girth	57.1428
Back width	50.3038	Ankle Girth	20.0174
Front width	59.4946		

Table 3.7: Measurements for characters C.

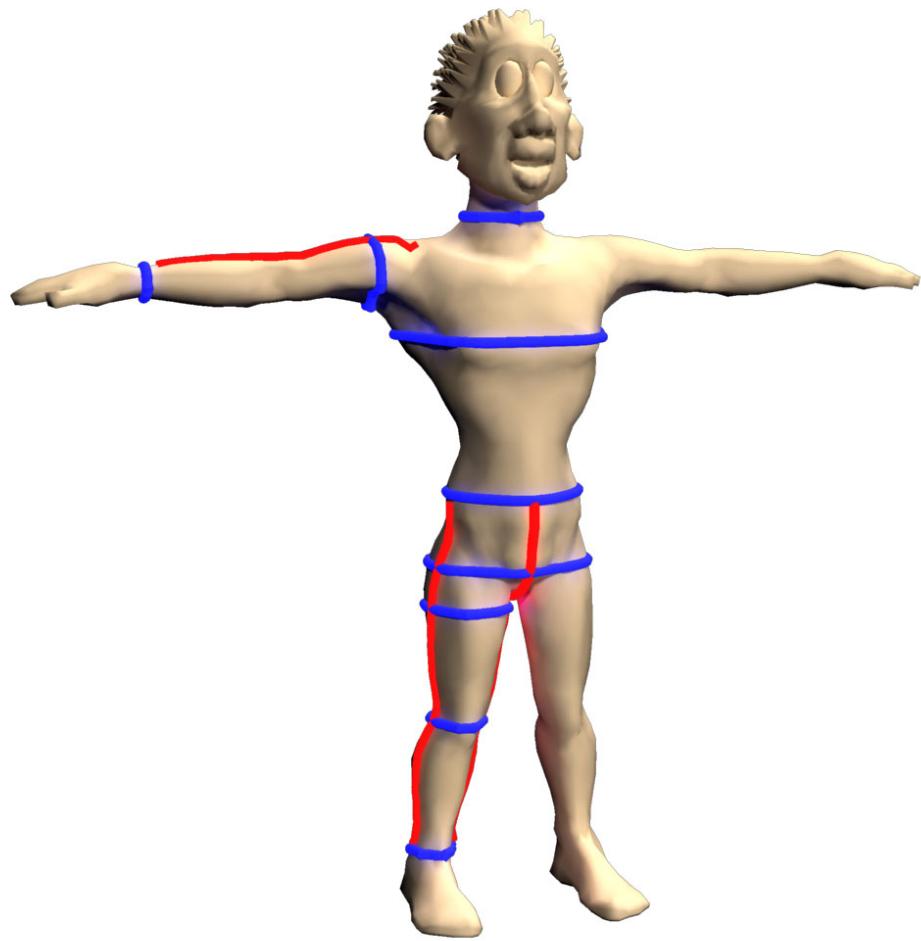


Figure 3.32: Measurements on Character D

Bust Girth	95.8160	Sleeve length	43.0211
Waist Girth	70.7769	Arm hole	37.9510
Hip Girth	73.1573	Back length	36.4173
Neck Girth	59.6871	Outside Leg length	51.4826
Cuff Girth	14.5720	Inside Leg length	41.3055
Shoulder width	34.2857	Thigh Girth	38.5871
Back width	32.2461	Ankle Girth	21.7142
Front width	38.1302		

Table 3.8: Measurements for characters D.

The standard posture used for modelling character differs from person to person. Traditional anthropomorphic data acquisition method requires character to stay in a standard posture in order to extract the correct measurement data. Therefore, when applying traditional anthropomorphic measuring method, different character postures will result different measurements. Hence the cloth that is adjusted based on those measurements will be ill-fitted. On the contrary, geodesics are more close to the circumstance of tape measuring in the real world, it also results less variation when the posture of the character changes.

This chapter proposed a character body measurements extraction method for animation character based on convex-hull and geodesic path computation. Calculating geodesic using current method is a time-consuming task especially on high resolution character models. In order to improve the efficiency of geodesic computation, two algorithms respectively for accurate(Algorithm 2) and high efficiency approximate geodesic computation(Algorithm 3) on triangulated manifolds are proposed in this chapter. For Algorithm 3, due to the fixed window size, it is able to achieve linear time complexity with a bounded error on triangulated manifolds. Numerical comparisons with existing algorithms (i.e. MMP, ICH_1 and ICH_2) have further demonstrated the advantages of Algorithms 3 in terms of both speed and accuracy. Experiments have shown that by integrating Algorithm 3 into the measuring system, the time consumed for measuring a character has been largely reduced.

Chapter 4

Cloth Modelling and Re-targeting

This chapter presents an automatic pattern based cloth modelling method that generates fit cloth for characters. Based on character measurements acquired by the method presented in chapter.3, the shape of cloth patterns are automatically adjusted. This method is able to create fit cloth efficiently for different characters with different body shapes and proportions without tedious manual operation that is required in traditional pattern based cloth modelling methods.

4.1 Introduction

In computer animation, creating assets such as models or scenes for a film is time consuming and costly. Therefore, the reusability of assets is curial to the film production budget. For rigid model such as cars or buildings, once created, they are stored into a asset warehouse and can be used in other films with little tweaking required. Such assets can also be traded among animation studios or animation artists. However, for clothes, their shapes follow the profile of a character, in most cases, they are bespoken for a particular character. Usually, constructing a complete set of outfits for an animation

character takes a well trained animation artist several days to finish. Due to the large difference between character designs among different films, reusing cloth models requires a large amount of manual operation to fit a cloth to another character with different body shapes and proportions.

Cloth is a soft object whose shape is determined by its design, body shape and posture of the wearer. Same cloth dressed by different character will results different cloth shapes. Moreover, in computer animation, characters are usually modelled in different postures, it is difficult to define a unified 3D form for cloth model to be stored, distributed and reused because for every different character, 3D cloth model needs to be heavily modified in order to cope with the new character body shapes and proportions.

In fashion industry, after generations of development, cloth pattern has become an important medium that transfers cloth design into wearable object. Cloth patterns are the basic items that constitutes a functional cloth. In order to implement a cloth design, firstly, the design needs to be broken down into a set of textile pieces, then through the assembling techniques such as stitch and adhesive, pieces are assembled together in a predetermined order to form a complete cloth. The process that breaks down a cloth design into patterns is called “patternmaking” (Armstrong 2000). Through this process, cloth design is transferred from design concept into a set of patterns that can be cut from the raw textile material. Therefore, the same cloth design can be reproduced easily(Hannah 1919). Modern cloth pattern is designed to follow the measurements of standard body template to maximize the generality. Then, by providing the measurements of a specific customer, the shape of each pattern can be adjusted respectively to fit the customer. In massive cloth production, a complex pattern resizing procedure named is developed to adapt the needs for dressing general public. Cloth pattern grading aims at resizing the cloth into several size categories based on the human anthropometric statistic data to fit the most individuals in the general public(Moore et al. 2001).

Cloth pattern in fashion industry has provided a standard to preserve a cloth design, when utilising made-to-measure tailoring technique, it can be resized to fit any customer. The cloth modelling method presented in this chapter adapts this custom tailoring techniques from fashion industry into computation animation. The basic idea is that, clothes are stored in the form of cloth patterns, when modelling cloth for a character, the measurements are acquired for pattern shape adjustment. Each pattern is adjusted based on the corresponding measurements from the character. Finally, cloth pattern are assembled together following a predefined sewing rule to form a complete 3D cloth. .

However, in virtual world, the body shapes and proportions of a animation character are only limited by the imagination of artists, therefore, using size chart developed from real human body to perform proportional scaling to fit cloth to a animation character with exaggerated body shape longer applies. Therefore, for each unique character, their cloth needs to be bespoke from measurements. However, resizing cloth pattern based on measurements requires deep tailoring knowledge which few animation artist process, the difficulty of properly resizing a set of cloth pattern has become bottle neck for the application of pattern based cloth modelling method in computer animation. In order to improve the reusability of clothes in computer animation production, this chapter presents an automatic cloth pattern resizing method to adjust cloth pattern based on the measurements of a character. By automating the cloth pattern resizing process, deep knowledges in tailoring are no longer required in this method. Clothes model can be stored in an unified form (cloth pattern) into asset warehouse, with the presented automatic pattern resizing method, by providing measurements of a new character, cloth pattern can be reused to dress a different character with different body shapes and proportions. Figure 4.1 demonstrates the general workflow of the pattern based cloth modelling method presented in this chapter

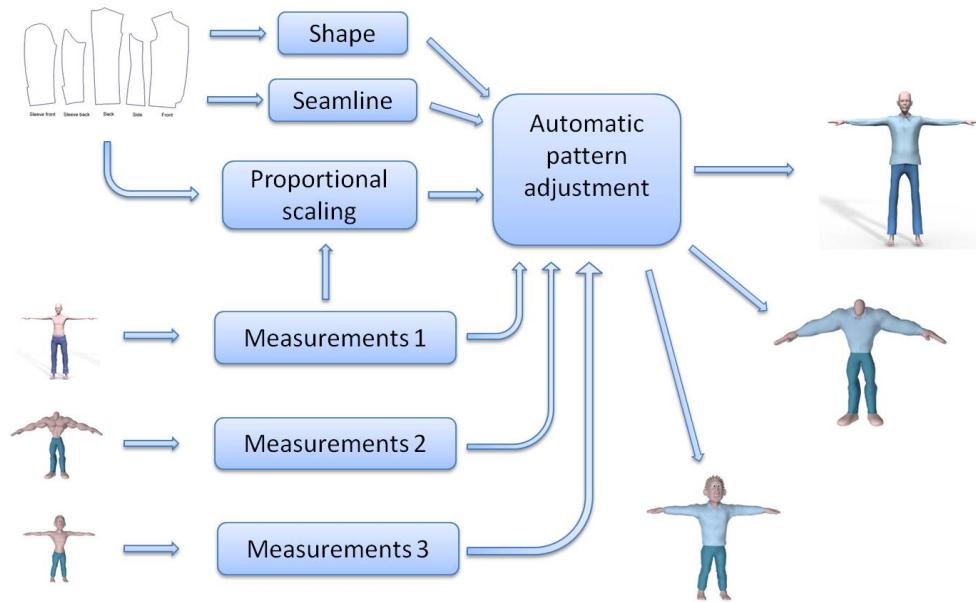


Figure 4.1: Workflow of pattern based cloth modelling method presented in this thesis

4.1.1 Patternmaking

Armstrong (2000) defines cloth patterns as the templates from which parts of a garment are traced onto the fabric before cutting out and assembled. When creating patterns, blocks(also known as slopers) are created first. Blocks are two-dimensional templates that consists of basic design of the a cloth type(Howland 2008), Figure 4.2 demonstrates a block for suit. Blocks are constructed based on the measurements taken from standard body template or the wearer(Armstrong 2000). When measurements are taken from an individual, it provides a good indication of the body dimension that a cloth design is intend to fit. In made-to-measure tailoring, measurements acquisition is particularly important since it directly determines the fit of cloth. Therefore, this process is usually performed by an experienced tailor. In massive cloth production, blocks are usually created by using measurements from a size chart which is based on a particular ethnic or group of the target customers.

For example, US standard clothing size chart(ISO/TR-10652 1991) contains body proportions of the general public of the Americans and European standard clothing size chart(EN:13402 2001) contains the size data of the people who were born in Europe. Based on the blocks, cloth patterns are created by introducing pockets, style line, drapes and other adjustments. Figure 4.3 demonstrate the blocks for a woman's shirt.

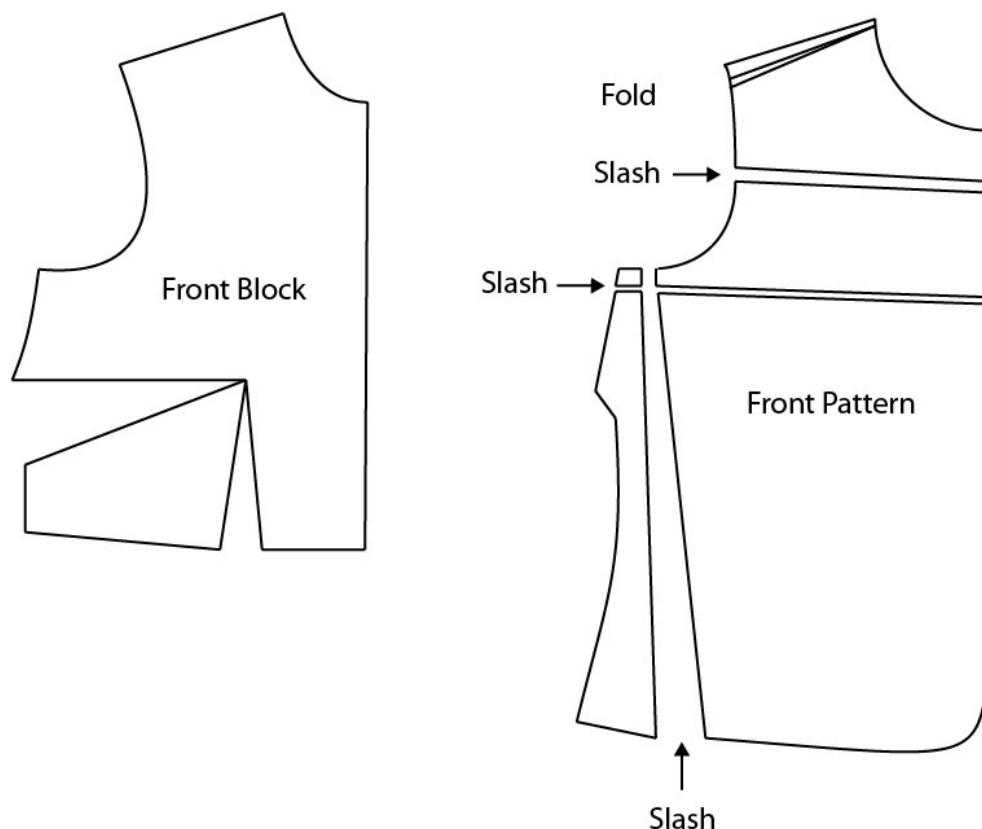
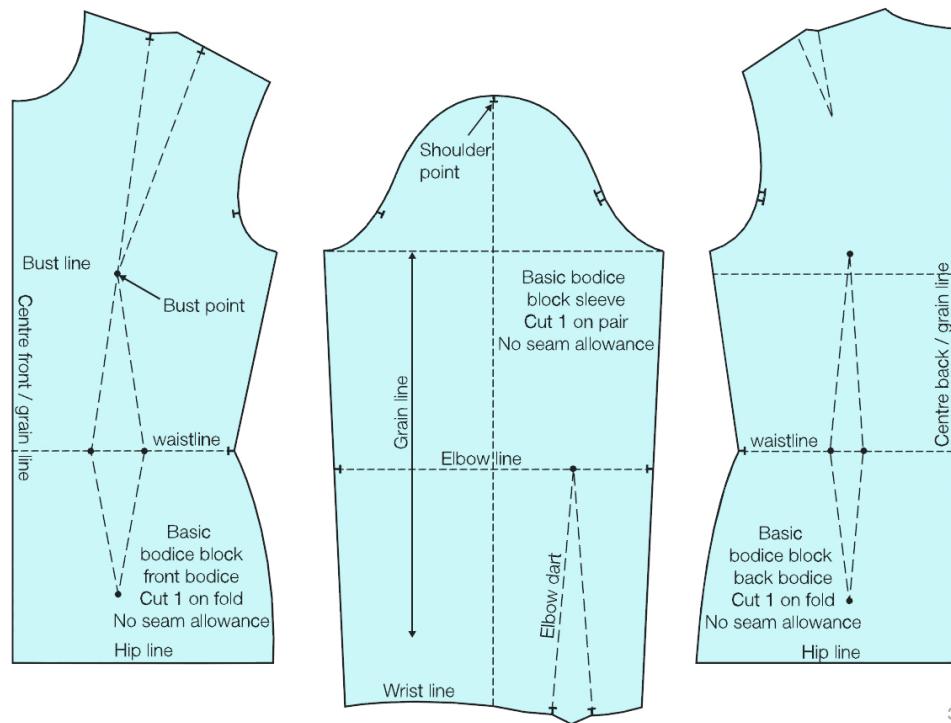


Figure 4.2: Block(left) and a pattern(right) made from it by adding details(Howland 2008).



3

Figure 4.3: Garment blocks for woman shirt(Rosen 2004).

Figure 4.3 demonstrates landmarks that are associated with datum points on a character on a set of blocks. Pattern grading is the process of systematically increasing and decreasing dimensions of patterns in to a range of sizes for production. During pattern grading, patterns are scaled proportionally into different sizes with predefined intervals introduced by a size chart. This process not only retains the original design of the cloth during distribution but is also very cost effective during the process of manufacturing due to the people who fit a predefined size are normally distributed into an interval introduced by a size chart (Schofield & LaBat 2005; Moore et al. 2001).

However, in computer animation the situation is much more complexer than in reality. In reality, for each ethic or certain group of people, the similarity of body proportion exists among individuals. In virtual world, the body proportion is only limited by the imagination of artists, therefore using size

chart to perform proportional scaling to patterns to create fit cloth no longer applies to the virtual character. In most cases, clothes for animation character are bespoke to a particular character. Reusing a cloth to another character with different body shapes and proportions requires large amount of modification.

4.1.2 Pattern Resizing Criteria

During pattern grading process, all patterns are scaled proportionally to preserve the shape of pattern respectively (Moore et al. 2001). However, as aforementioned, the body proportion of the virtual character is significantly different from humans. Therefore, performing proportional scaling to patterns might leads to an undesired result.

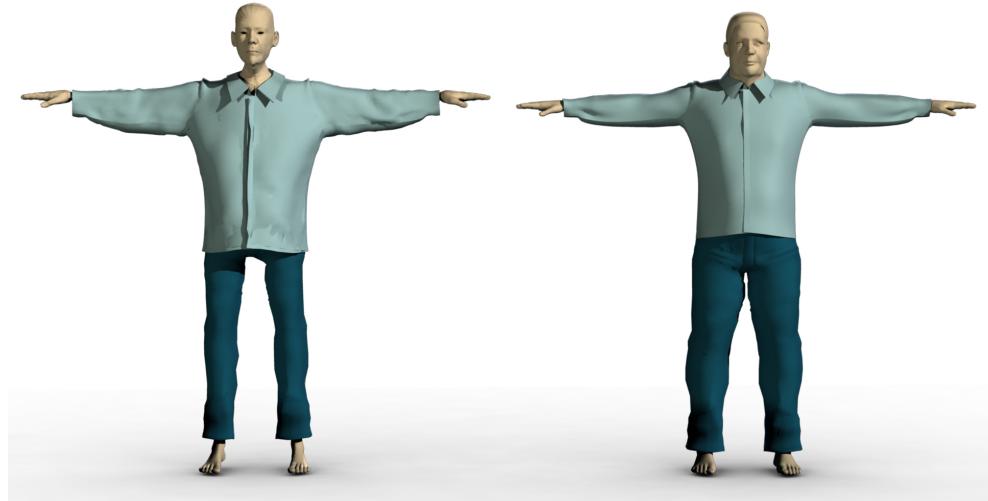


Figure 4.4: Same shirt proportional scaled to different characters with same arm span and height using method presented by Moore et al. (2001)

Figure 4.4 demonstrates the result of using same shirt to dress two characters with same height and arm span using proportional scaling method presented by Moore et al. (2001). Because these two characters share same

height with is the major measurements used to define the interval of a size chart, both of the character falls into the same grading intervals. Therefore, clothes are adjusted into same size. However, the character on the right is much stronger than the one on the left, the body of right side character supports cloth much better than the other. On the contrary, the skinny character has much thinner limbs which results a baggy shirt. In order to achieve better fit, during made-to-measure cloth tailoring, extra measurements such as “Arm Hole Length” or “Cuff Girth” are taken from customer in order to fine tune the cloth for better fit. With those extra measurements, different parts of a pattern are scaled in different manners. In order to generate patterns based on the measurements, a few geometric criteria need to be evaluated in order to ensure the fit of cloth as well as maintaining the design of cloth.

Character Measurements Cloth pattern represents different parts of a garment, which responsible for covering certain parts of body, therefore, all the patterns need to be deformed in a manner that each pattern matches to the measurements of its corresponding body part. Because all the patterns are derived from blocks, during pattern adjustment, for each block which a pattern is developed from, the distance between landmarks that are associated with the datum points on the character body needs to satisfy the requirement set by the measurements taken from the character. Especially when the character has an unusual body proportion, different parts of a block might be scaled in different fashions. For instance, in Figure 4.4, the arm of both two characters have the same length, however, the circumference of arm are mush different, Moreover, despite the difference between the circumference of arms, the wrist girth are the same for both characters. Therefore, in order to satisfy the all the measurement requirements, different parts of sleeve pattern need to be adjusted in different manners.

Pattern shape Cloth pattern is the most basic component of a complete gar-

ment and the shape of each pattern defines the shape of cloth. Pattern grading process is considered as the most difficult process during the tailoring which only the professionals is able to master. The overall shape of a cloth pattern can be altered significantly to accommodate the difference of body proportions between different customers. However, details of the pattern such as slops, darts and their relative location are kept (Moore et al. 2001). In computer animation, the goal of transferring cloth from one character to another is to fit a cloth onto a different character without altering the design of the cloth. However, each cloth pattern needs to be adjusted independently in order to match the measurements from character. Therefore a shape evaluation process is needed after adjusting cloth pattern in order to maintain the design of the cloth.

Seam Line In reality, cloth patterns need to be stitched together to form a complete garment. The adjacent edges between two patterns is the seam-line. A seam-line consists of many pairs of stitching points which are located on the boundary of two adjacent patterns. During the adjustment of pattern, each pattern is adjusted individually to meet the measurements. Therefore there is no guarantee that after the adjustment of the patterns, the seam-line remains consistent. Because seam-line determines the location of the patterns after it is assembled into a complete garment, without the preservation of the seam-line, original design of the cloth cannot be preserved.

4.2 Cloth Resizing Algorithm

In this section, an automatic cloth pattern adjusting method is presented. This method operates cloth size adjustment directly on 2D patterns and optimizes each pattern to ensure all the patterns satisfy the criteria introduced in previ-

ous section.

Two inputs are required for this method. The first input is a group of patterns representing a cloth design. Within each pattern, the landmarks that associated with the datum point on character body are defined. Pattern landmarks are key points on a pattern inherited from block. They are defined on the cloth block by patternmaker to associate body measurements with the length on a block (Rosen 2004). In real world, pattern landmarks are used as a guide line for pattern resizing where tailor adjusts distance between pattern landmarks to meet body measurements. Moreover, seam-lines are presented along with the pattern. Figure 4.5 demonstrates an example of input patterns.

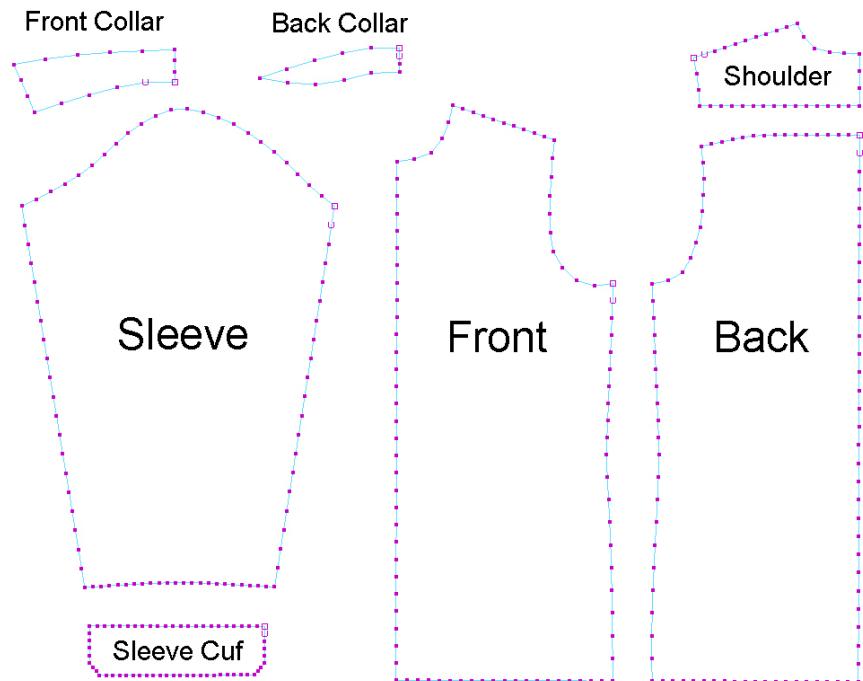


Figure 4.5: An example of a cloth pattern

The second input is the 3D character model which is modelled in a “T- pose”. Because this method involves physical simulation for pattern assembling process, the contact between limbs will leads to the penetration between cloth mesh and character skin. Moreover, the measurements of the character

are also provided by using the method from previous chapter.

In order to properly determine the size of patterns, block for each pattern is generated at first. In fashion industry, cloth pattern is created from an unique block by adding details. Therefore, in this method, a block is refers to a 2D bounding box of each pattern and all the landmarks that are associated with datum points on character body are defined on the blocks. Moreover, because a block and a pattern have a strictly one to one association which means for any pattern, it is generated from only one unique block. Also, because a block contains much less details than the pattern, performing pattern adjusting algorithm on blocks can saves lots of computational resources.

When constructing a block from a pattern, firstly, the “critical points” on a pattern are selected. The “critical points” are defined by points which are the most capable of representing the geometrical features of a pattern. For example, the sharp turning points on the contour of the pattern or extreme points are considered as “critical points”. Figure 4.6 illustrates “critical points” on the sleeve pattern. Finally, a bounding box is created for each pattern, within the bounding box, for every “critical point”, two orthogonal line are created to form a grid. This is demonstrated in Figure 4.6.

Next, based on the grid, a nurbs plane is constructed and control points of the nurbs plane are intersection points on the grid(this is demonstrated by the green cross in Figure 4.6). All points on each pattern can be represented by a parametric coordinate on the nurbs plane. For every pattern, a unique nurbs plane is created and it is considered as the block of the given pattern.

Then, the proportional scaling used in traditional cloth grading method is performed onto each pattern. In general, the measurements that are associated with the length of limbs are used as the reference for scaling such as “Back Length” for torso patterns and “Arm Length” for sleeve patterns. Figure 4.7 demonstrates the result of performing proportional scaling to sleeve block based on the “Arm Length” measurement.

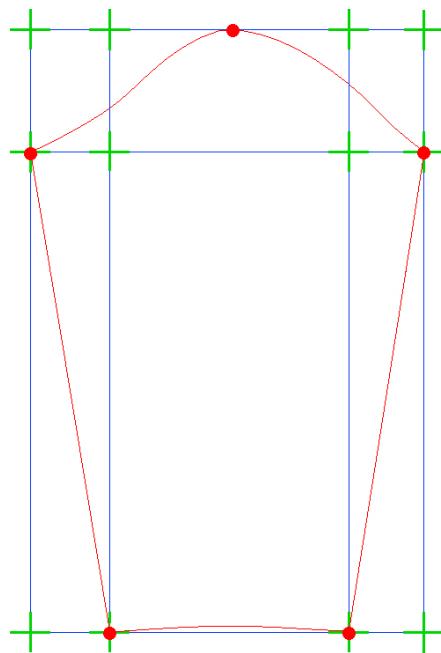


Figure 4.6: Block generated for sleeve pattern(red closed curve). The red points are “critical points” on sleeve pattern.

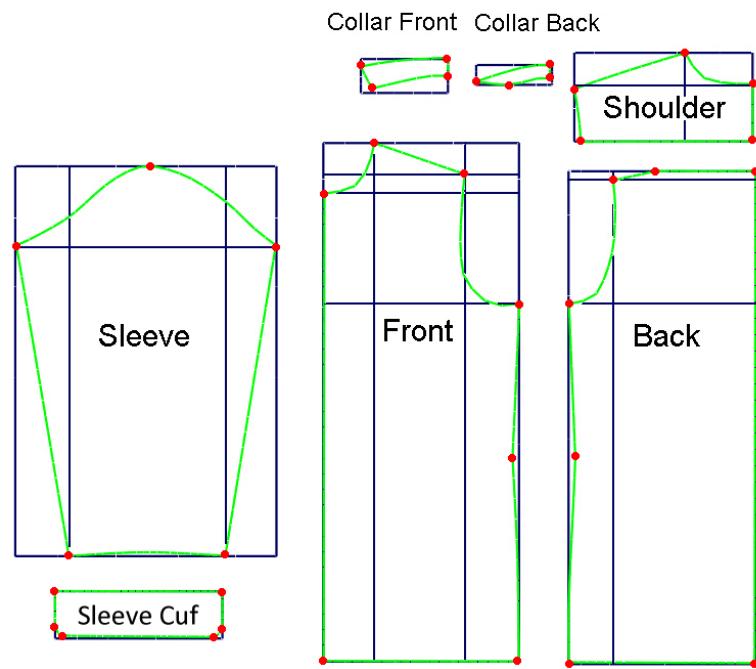


Figure 4.8: Blocks after proportional scaling is performed.

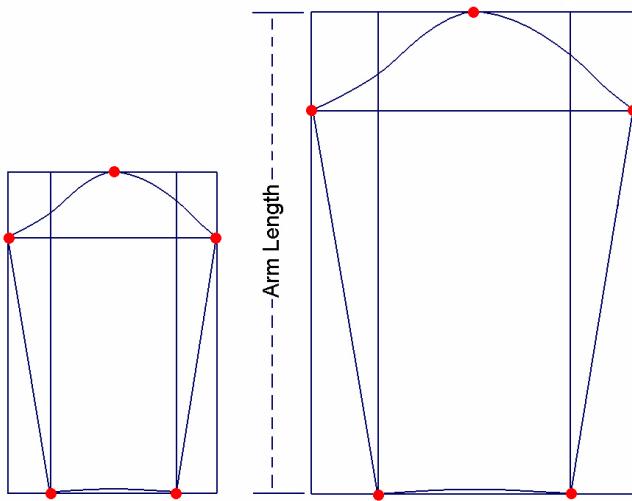


Figure 4.7: Block before(left) and after(right) the proportional scaling. The height of the sleeve pattern after scaling has reached the measurement “Arm Length”

Performing proportional scaling to cloth patterns has a significant drawback that needs to be improved. This drawback is demonstrated in Figure 4.9.

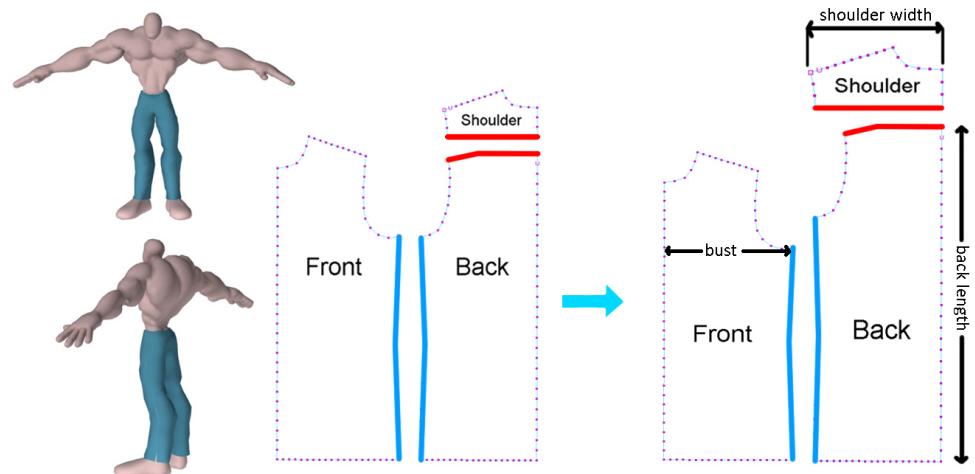


Figure 4.9: Inconsistency of seam-line after scaled patterns proportionally

Where ‘Front’, ‘Back’ and ‘Shoulder’ are three patterns of a cloth for Character A. Patterns are scaled according to their major measurement, For example, pattern “Front” is proportionally scaled based on measurement “bust”, pattern “Back” is proportionally scaled based on measurement “Back

length”. However, when pattern was initially designed, each pair of seam-lines between two adjacent patterns usually have same length and interval for each stitch, this is shown in Figure 4.9 as a pair of blue curves indicates a pair of seam-lines between pattern “Front” and “Back”, a pair of red curves indicates seam-lines between pattern “Shoulder” and “Back”. However, after proportional scaling is performed, the length of seam-line is no longer the same which further leads to undesired wrinkles around seam-line during pattern assembling process. At this stage, each pattern has good fit on measurements criteria but worsen on the seam-line criteria. If patterns are adjusted to meet the seam-line criteria, the fit on measurements will be break. Moreover, if patterns are firstly proportional resized based on measurements and then adjusted each pair of seam-line to satisfy seam-line criteria, the shape of pattern will be changed. Actually, in many cases, these conditions may not concord with each other and in some cases even conflict to each other. Therefore, it is very difficult to adjust cloth patterns to fit a character based on one criteria.

4.2.1 Genetic Algorithm

Modelling a cloth from cloth patterns for a character requires each pattern to be adjusted individually. During the pattern adjustment, three criteria need to be satisfied in order to fit the cloth to the character as well as preserve the design and integrity of cloth. The goal of cloth fitting process is to generate a set of cloth patterns that constitute a complete garment which fits character while having the least distortion compare to original design of this garment. In this thesis, this process is considered as a multiple objective optimization process.

Optimization problem is a very hot research area that has been studied intensively for decades and many techniques have been developed to tackle

such a problem. In the case of cloth pattern adjustment, since a complete cloth is comprised by a set of patterns, each pattern is formed by a set of points (sampling points on the contour of a pattern). The parameters of this optimization problem are the coordination of these sampling points.

For current gradient based optimization methods, handling large amount of parameters requires a lot of computational resources. Among many optimization methods, genetic algorithm dwarfs others by its simplicity and efficiency. Genetic algorithm works on chromosomes, which are encoded parameters of solutions. Because this encoding mechanism largely reduces the number of parameters used for optimization, much less computation is required by genetic algorithm. Moreover, unlike traditional optimization method, which searches cost surface from a single point, genetic algorithm searches cost surface in a parallel manner, it is not only able to scan a large number of potential solutions very quickly, but also able to avoid local optimal solution effectively due to its parallel searching method. In general, the advantages of the genetic algorithm over other optimization method are,

1. Genetic algorithm searches through a wide range of the cost surface simultaneously. Therefore it is able to deal with very complex cost surfaces and avoid local minimum.
2. Genetic algorithm works on the chromosome instead of real parameters, therefore it is able to handle large number of parameters.
3. The initial proposals do not effect the final solutions as bad solutions are discarded by selection at every evolution. Therefore, the genetic algorithm is not sensitive to the initial seeds.

In the thesis, three criteria are used for evaluating fit and style preservation during pattern adjustment process. However, these criteria are often conflict to each other. The optimization algorithm needs to be able to handle multiple objectives. Moreover, cloth patterns are represented by a set of

nurbs curves which are controlled by a set of “Critical Points”. Each “Critical Point” is a parameter to the optimization, large amount of parameters need to be handled at same time. Furthermore, the initial state of pattern is proportionally scaled based on each pattern’s major measurements, to some non-exaggerated human like character, it will provides a good approximation that is close to the final fit cloth. However, for some largely exaggerated character such as character A in Figure 4.9, proportional scaling will worsen other criteria. Therefore, cloth pattern optimization should not be sensitive to its initial seeds. Genetic Algorithm has advantages these three aspects, therefore, this thesis utilises genetic algorithm for cloth adjusting process. Because cloth patterns are adjusted based on the measurements of the character, adjusting cloth patterns using genetic algorithm has three extra benefits. Firstly, once the parameters are set, genetic algorithm evolve the initial seeds towards best solution automatically, therefore, given measurements of any character, this method can generates cloth patterns that fit the character automatically. By automating the pattern adjustment process using genetic algorithm, traditional tailoring knowledge is no longer needed for this process, the duplication of effort required by traditional pattern based cloth modelling method can be eliminated and the efficiency for modelling cloth for different characters with different body shapes and proportion can be largely improved. Secondly, this method empowers the creativity of animation artists and amplifies their productivity by allowing them to use a large amount of existing cloth patterns in the fashion industry to create various clothes that fit different characters. Thirdly, because the 3D cloth is generated based on the adjusted patterns, the work flow of modelling cloth for the character is one direction. There is no turning back for extracting patterns from 3D cloth which is required by current cloth modelling method. Therefore, the shape distortion that is introduced by 3D surface flattening process can be avoid.

Genetic algorithm(GA) is an optimization method which is based on the principles of natural selection (Haupt & Haupt 2004; Deb 2001). During

natural selection, effected by environment, the biological traits of organism become more or less common in a population through generations of reproduction. A genetic algorithm allows a population that consists of many individuals to evolve under certain rules and to a state that minimizes the cost function (Holland 1992).

A genetic algorithm starts from a group of randomly generated solutions called “initial population”. Within the initial population, each individual is a set of variables that represents a solution of the problem. By evaluating every individual using cost function, a ranking is assigned to the individual in terms of performance of cost function. Then, a selection method is applied to select a group of individuals to perform crossover and mutation operation. Finally, the convergence of current generation is evaluated. If current generation does not reach the minimum on the cost surface, this process loops back to step one, the algorithm is executed iteratively till the minimum cost is reached, this procedure is demonstrated in Figure 4.10

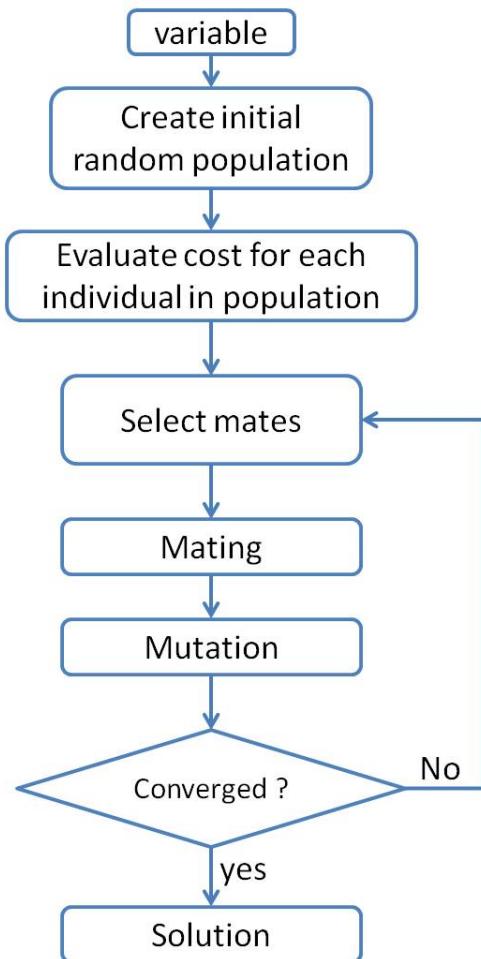


Figure 4.10: Workflow for single objective genetic algorithm

In genetic algorithm, “fitness” refers to the cost of an individual in terms of its performance to the cost function, higher the fitness, lower the cost. In single objective optimization, only one individual who has the highest fitness is selected as the final solution of the problem. However, in many applications, more than one objective are needed for describing the problem. Moreover, among these objectives, conflicts often occur. For example, adjusting a pattern for a particular measurement usually leads to worsen the consistency of seam-lines. In this case, it is impossible to locate a single best solution that is optimum with respect to all objectives. This kind of optimization is called multi-objective optimization, the purpose of multi-objective optimization is to find as many good solutions as possible. The solution set results from a

multi-objective optimization is called Pareto-optimal solutions (Pareto 1906). However in many applications, only one solution is needed, therefore, usually, after Pareto-optimal solutions have been formed, a higher-level objective is used to select one solution out of the Pareto-optimal solutions as the final solution for the problem.

In the following section, a multi-objective genetic algorithm for pattern adjustment is presented in detail. This algorithm uses multi-objective optimization to find a combination of patterns that has best fits on measurement criteria, seam-line criteria and pattern shape criteria.

4.2.2 Definition of Population

Within a population, individual is the basic element represents one solution to the problem (Haupt & Haupt 2004). In our case, each pattern is described by a set of points located on the contour of a pattern $Pattern_i = [p_0, p_1, \dots, p_n]$. A block is a nurbs plane that the pattern is inscribed to and it is defined by a set of control points, noted as $b_i = [cp_{i1}, \dots, cp_{im}]$. After a block is created based on a pattern, Points on the pattern can be interpolated by parametric coordinate on the block plane. Each pattern has its unique block, therefore, for each pattern, it can be represented uniquely by control points of the corresponding block plane. Note that $m \ll n$, which means the number of control points is much less than the actual number of points on the pattern, particularly, in experiments, m ranging from 4 to 30 and n ranging from hundreds to a thousand. Consequently, a pattern can be represented by control points of the block effectively.

$$\begin{cases} b_1 = cp_{1_1} \quad cp_{1_2} \quad \cdots \quad cp_{1_m} \\ b_i = cp_{i_1} \quad cp_{i_2} \quad \cdots \quad cp_{i_{m'}} \\ \vdots \\ b_n = cp_{n_1} \quad cp_{n_2} \quad \cdots \quad cp_{n_{m''}} \end{cases} \quad (4.1)$$

Table 4.1: *The structure of gene*

Table 4.1 illustrates the structure of chromosomes. Where cp_i denotes a control point on block b_i , an individual is comprised by a group of blocks $ind = b_1, \dots, b_n$ which representing a complete cloth. In general, for each individual, a gene represents a control point, a chromosome represents a block, and an individual is constituted by a group of chromosomes (blocks).

The initial population is the first sample over the cost surface. Because in the case of pattern adjustment, the measurements of character can be largely different form character to character, the boundary of the cost surface can be difficult to determine. Although for GA, the initial population will not affects the formation of Pareto-front, setting the initial population close to the final objective can largely reduce the number of generations required for generating the Pareto-front. Therefore, before performing genetic algorithm, traditional proportional scaling method is used to initialize the first generation of population. For every input pattern, the measurements taken from the subject that associated with this pattern is stored into an array, denoted as $M_i = [m_{i_1}, \dots, m_{i_n}]$. For each stored measurement, it contains datum points that this measurement is taken from and every datum point on the subject has an unique corresponding landmark on the pattern. This is illustrated in Figure 4.11

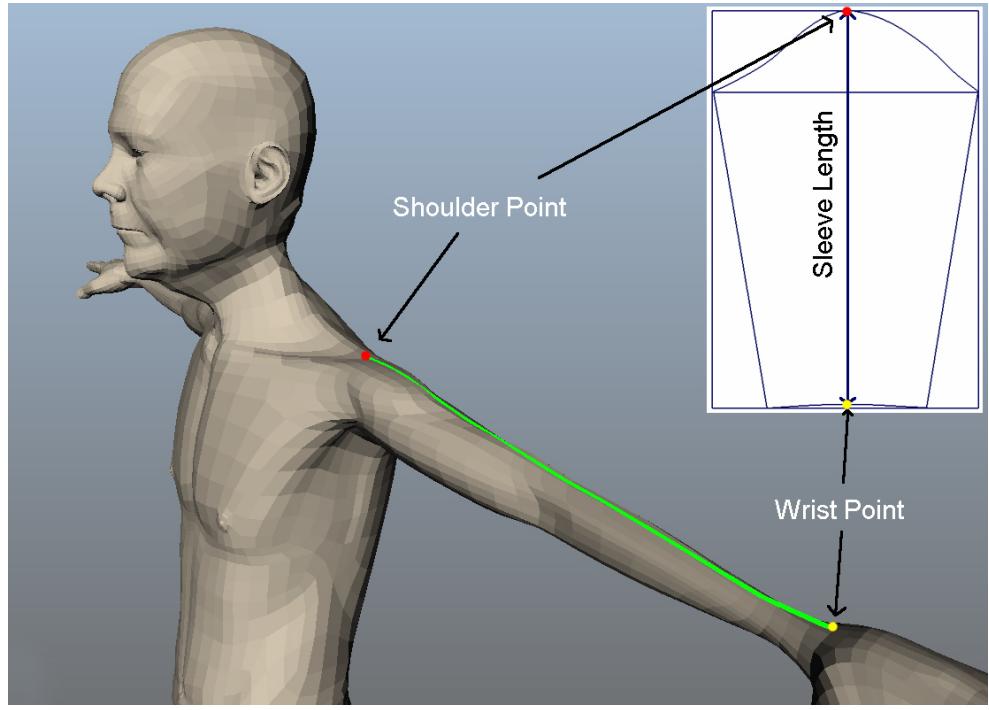


Figure 4.11: Association between body datum point and pattern landmarks. The green line is the geodesic from shoulder point(red point) to wrist point(yellow point). The red and yellow point on the sleeve pattern is the pattern landmarks that associated with the datum points on the body of character. When sleeve is well fitted, the “Sleeve length” should match the length of this geodesic.

The scaling factor can be calculated by Algorithm 4.

Algorithm 4 Initial Pattern

```

1: procedure PROPORTIONALLY SCALE PATTERN(Pattern  $P$ , Measurements  $M$ )
2:   for measurement  $m_i \in M$  do
3:      $p_s$  and  $p_e$  is two datum points of  $m_i$ 
4:      $l_s$  and  $l_e$  is two landmarks on a pattern that  $l_s \sim p_s$  and  $l_e \sim p_e$ 
5:      $d \leftarrow$  distance between  $l_s$  and  $l_e$ 
6:      $Sr \leftarrow m_i/d$ ( $Sr$  denotes scaling factor)
7:   end for
8:   Select largest  $Sc$  from  $S$  as the scaling factor for  $P$ 
9: end procedure

```

By using Sc as the scaling factor, the input patterns are proportionally scaled. Although the proportional scaling method cannot produce a well fit

cloth for the character, it is able to resize the patterns to a certain degree so that the cloth roughly fits the character. Then, the proportional scaled patterns are used as the seed for generating the initial population. In the next step, in order to create individuals that spread over the cost surface, a mutation operation are carried out on each pattern to vary its shape. This operation will be explained in detail in “Crossover and Mutation” section.

In order to improve the efficiency of the evolution, the individual can age during the evolution. In nature, the size of the population changes over generations because individuals who carries a fit gene has better crossover opportunity than those who carries less fit gene. Therefore, naturally, a fit gene is able to live longer during evolution than a less fit genes. When a gene dies, the individual who carries this gene also dies. Consequently, the number of the individuals in a generation is determined by the death of the less fit gene.

$$life_i = \begin{cases} life_{min} + \eta \frac{cost_{max} - cost_i}{cost_{max} - cost_e} & \text{if } cost_i \geq cost_e \\ \frac{1}{2}(life_{min} + life_{max}) + \eta \frac{cost_e - cost_i}{cost_e - cost_{min}} & \text{if } cost_i < cost_e \end{cases} \quad (4.2)$$

Equation 4.2 is a bilinear method that introduced by Michalewicz (1996) for calculating life span of a gene. Where, $life_{min}$ and $life_{max}$ denote the shortest and longest life span for a gene, $cost_{max}$ and $cost_{min}$ denotes the highest and lowest cost of the current individual that carries this gene. $cost_e$ denotes the expected value of an individual and $cost_i$ is the actual cost value for current individual. When a gene reached its life span, it dies before it is evaluated by the cost function. During the evaluation, if lots of individuals died in one generation, inadequate number of individuals may results the limited searching range across the cost surface which further leads to premature convergence to local minima. Therefore, dead individuals are replaced by

newly mutated individuals to maintain the coverage over the cost surface.

In the presented algorithm, the lifespan of a gene is assigned every time when evaluation is performed to the newly generated individual. For every generation, the $cost_{max}$ and $cost_{min}$ of a gene are determined by the worst and the best gene of a pattern within the previous generation in terms of three evaluation functions accordingly. The $cost_e$ is determined by the average cost of all the contour points of a pattern that are derived from the block in the previous generation. This ensures a gene in the current generation can not live longer than the previous better gene. When selection is performed, any individual who carries a gene that has reached its lifespan is discarded.

4.2.3 Crossover and Mutation

In nature, crossover is the source power for evolution. Because crossover is the only way for creating new individual that carries genes from their parents. The chromosome of offspring is the recombinations of the genes from their parents. Mutation is another method for alternating genes, it is able to introduce the new gene into the chromosomes of an individual without the need of pairing. Both method enables the genetic algorithm to explore new area on the cost surface. In general, mutation are often used to provide exploration and crossover are mostly used to lead the population to converge into current good solutions.

Crossover is an exploitation to a certain area on the cost surface. It requires two individuals to be involved to the process and normally generates two offspring. In order to perform crossover, two parents need to be selected from the population first, the difference between two parents needs be large enough to create an effective offspring, because two similar parent will result a very similar offspring and further leads to an over dense sampling in a small region near the parents, which will cause premature convergence. In

each generation, the individuals are sorted based on domination. Therefore, when selecting parents from population for crossover, an index interval is used for maintaining the difference between two selected parents. After both parents are selected, then recombination of the chromosomes takes place, this is demonstrated below.

$$Father = [cf_0, cf_1, cf_2, cf_3, \dots cf_n]$$

$$Mother = [cm_0, cm_1, cm_2, cm_3, \dots cm_n]$$

where cf_i is the chromosomes from father side and cm_i is the chromosomes from mother side. Normally, crossover generates two offspring, in which both child carries both part of chromosomes from their parents.

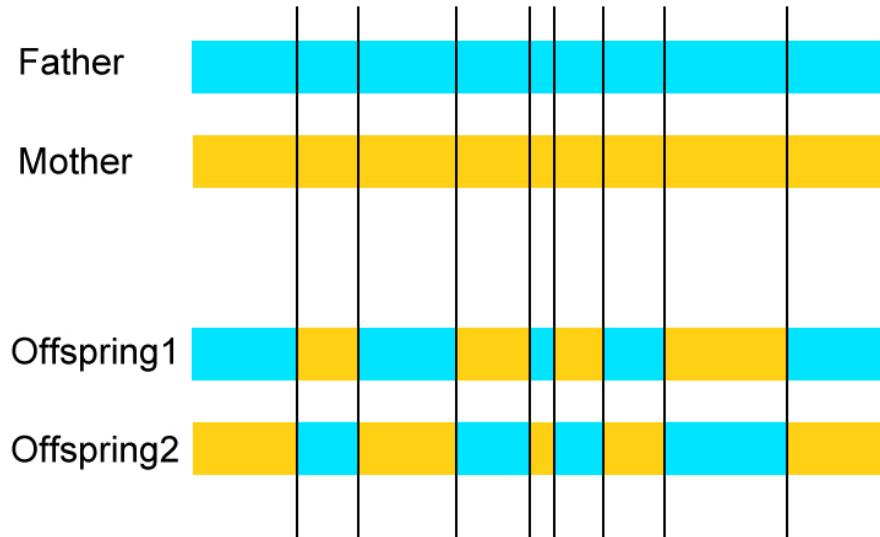


Figure 4.12: Crossover of parents

Figure 4.12 demonstrates the uniform crossover method introduced by Spears & Anand (1991); Gwiazda (2006); Ghosh & Tsutsui (2003). This method uses a fixed ratio exchange rate between two parents to generate off-

spring. The black line indicate the exchange point in chromosome that are randomly selected by a random number generator.

For pattern adjustment, each individual is constituted by a group of different patterns, when two individual mates, crossover operation only performs on same pattern in both side of parents. For example, crossover operation can only be performed on two parent “Front” pattern to generate two offspring “Front” patterns. The crossover operation is summarized in Algorithm 5.

Algorithm 5 Crossover

```

1: procedure CROSSOVER(Two individuals  $ind_1$ ,  $ind_2$ , crossover rate
    $cxbp$ )
2:   if  $cxbp > \text{random}()$  then
3:     for same block( $b_1, b_2$ ) in  $ind_1$  and  $ind_2$  do
4:        $N \leftarrow$  number of control point in current block
5:       for  $i = 0; i < N; i + +$  do
6:         if  $\text{rand}() < 0.5$  then
7:            $offspring_{1i} \leftarrow b_{1i}$ 
8:            $offspring_{2i} \leftarrow b_{2i}$ 
9:         else
10:           $offspring_{1i} \leftarrow b_{2i}$ 
11:           $offspring_{2i} \leftarrow b_{1i}$ 
12:        end if
13:      end for
14:    end for
15:  end if
16: end procedure
```

Because crossover only operates on the existing gene pool that is comprised by genes from two parents. It samples the area near their parents much more denser than mutation does. A higher crossover rate will increase the speed of convergence. In real world, a species will extinct without introducing new gene because the limited gene pool will be exhausted during the new born of the individuals. In genetic algorithm, this phenomenon appears as the over dense sampling into a local area around the initial population. This often results premature termination of the evolution.

Mutation is a process that new genes can be introduced into the gene pool so that the diversity of the gene pool is enriched and gene degradation can be avoided. In genetic algorithm, mutation can explore new area of the cost surface much more efficient than crossover. Haupt & Haupt (2004) indicates that two issues should be taken into the consideration when performing mutation operation to an individual, the type of the mutation and the rate of the mutation. Grefenstette (1986); Srinivas & Patnaik (1994a) point out that the choice of the mutation rate is heavily problem specified. For different problems, best mutation rate varies significantly, it needs to be set empirically for each particular problem.

When performing genetic algorithm, a high mutation rate enlarges the searching range on the cost surface, whilst prevents the population to converge to a specific point. In the meantime, a very low rate of mutation will leads to a premature convergence very easily. According to the work of Yaman & Yilmaz (2010), in different stages of the evolution, the genetic algorithm usually requires different exploration-exploitation ability. In this thesis, the non-uniform mutation, which presented by Michalewicz (1996), is applied, in which the possible impact of mutation to an individual decreased when generation evolves.

When initializing the initial population from seeds, a very wide distribution over the sampling cost surface is preferred, therefore, large mutation rate is set. Before evolution comes to the end, exploitations is much preferred than exploration so that the convergence can be ensured. At this stage, a high mutation rate will disturb the convergence of the algorithm. In the implementation of this algorithm, the mutation rate starts from 0.6 and decreased linearly by 0.002 for every generation that has evolved. Assume that Gen_{max} is the predefined maximum number of generations of evolution. Then, for each individual, the randomly chosen chromosome cp_i is replaced by one of

the two values demonstrated in Equation 4.3,

$$cp_i = \begin{cases} cp_i + \Delta(gen, t) & \text{if } \gamma \geq 0.5 \\ cp_i - \Delta(gen, t) & \text{if } \gamma < 0.5 \end{cases} \quad (4.3)$$

where, gen denotes the index of the current generation, γ is a normally distributed random number from 0 to 1, $\Delta(gen, t)$ is a random variable that mutates cp_i in range $[0, t]$. The value of $\Delta(gen, t)$ is determined by the index of current generation gen by Equation 4.4, introduced by Michalewicz (1996),

$$\Delta(gen, t) = t * \left(1 - \lambda^{(1 - \frac{gen}{Gen_{max}})^r}\right) \quad (4.4)$$

where, λ is an normally distributed random value from 0 to 1, Gen_{max} is the maximum number of generations of the evolution. The exponential factor r controls the influence of gen on the distribution of $\Delta(gen, t)$ within its range. In which this operation becomes an uniform mutation if $r = 0$. The mutation process is introduced in more detail by Algorithm 6

For each mutation, only one orthogonal direction is modified in either row wise or column wise. For example, this mutation operator performs on U direction only for a selected row of the control points or on V direction only for a selected column of the control points. The choice between two directions are randomly selected for each block. For each row of the block, The mutated U value of each control point is stored into an array denoted as U_{row} . The original U value of every control point is also stored into an array denoted as U'_{row} . For each pattern, its block is created as a rectangle nurbs plane that the pattern is inscribed to. Therefore the U value of each row or column of a block is in the same incremental or decremental order. this is demonstrated in Figure 4.13.

Algorithm 6 Mutation

```
1: procedure MUTATION(Individuals ind, Generation index gen, Mutation  
rate mutpb)  
2:   if mutpb > random() then  
3:     for blocki ∈ ind do  
4:       dpb ← random()  
5:       U ← the number of control point on U direction  
6:       V ← the number of control point on V direction  
7:       if dpb < 0.5 then           ▷ Variate cpi in U direction  
8:         for Each row of cpj on blocki do  
9:           Append u value of cpj into array Variables  
10:          end for  
11:          Apply Equation 4.3 to each element in Variables and  
evaluate the newly formed Variables'  
12:          if Variables' is invalid then  
13:            Reapply Equation 4.3 to each element in  
Variables and evaluate the newly formed  
Variables' till Variables' is valid  
14:          end if  
15:          for Each row of cpj on blocki do  
16:            Assign back u value of cpj from corresponding  
element in Variables  
17:          end for  
18:        else           ▷ Variate cpj in V direction  
19:          for Each column of cpj on blocki do  
20:            Append v value of cpj into array Variables  
21:          end for  
22:          Apply Equation 4.3 to each element in Variables and  
evaluate the newly formed Variables'  
23:          if Variables' is invalid then  
24:            Reapply Equation 4.3 to each element in  
Variables and evaluate the newly formed  
Variables' till Variables' is valid  
25:          end if  
26:          for Each column of cp on blocki do  
27:            Assign v value of cpj with corresponding element  
in Variables  
28:          end for  
29:        end if  
30:      end for  
31:    end if  
32: end procedure
```

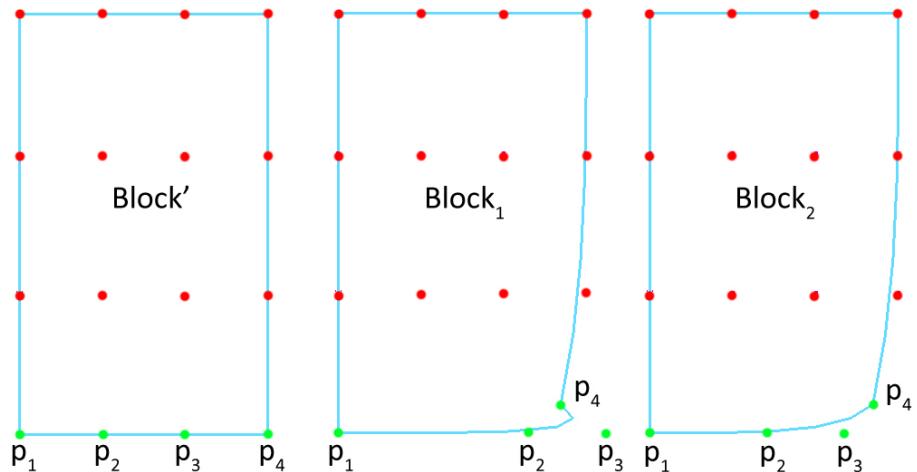


Figure 4.13: *Block'* is at its initial status, *Block*₁ and *Block*₂ are the results from mutation operation applied on the bottom row of control points(green points).

When a block is at its initial status, such as *Block'* indicated in Figure 4.13, the order of the control points in the selected row is denoted by $R' = [p_1, p_2, p_3, p_4]$. *Block*₁ and *Block*₂ are two possible results generated by mutation operation on *Block'*. Sorted by the *U* coordinate, the order of the control points in the corresponding row can be written as $R_1 = [p_1, p_2, p_4, p_3]$ and $R_2 = [p_1, p_2, p_3, p_4]$. In order to avoid face overlapping on a cloth pattern during mutation, the order of control points in any row or column need to be maintained. Since $R' \neq R_1$ and $R' = R_2$, *Block*₁ is marked as an invalid gene for current individual. In order to model cloth, each pattern need to be triangulated first to from the polygon from the outline of the pattern. In the case of *Block*₁, because the order in R_1 is changed, it will cause faces overlap to each other during the triangulation of the pattern. When a pattern is invalidated, mutation needs to be reapplied till $R' = R_1$. In this case, *Block*₂ is a valid gene of the current individual.

4.2.4 Evaluation and Selection

In genetic algorithm, fitness function(cost function) describes the objective of the problem. Given a solution, the fitness function is used to measure how far the current solution is to the desired goal of convergence. When a population evolves into a new generation, the selection operator deletes the n worst individuals and breeds n new individuals from the best individuals. In order to perform such a process, each individual need to be awarded a rank to indicate how close it comes to the ideal solution, ranking is generated by applying fitness function to each solution. In this algorithm, objectives for evolution is to adjust size and shape of the each pattern so that each pattern fulfil the requirement of the measurement criteria . Also, among patterns, the topology of the seam-line where two patterns are joint together remains. Most importantly the consistency of cloth design must be retained after resizing. Therefore, three fitness functions are developed for evaluating the fitness for each individual respectively.

Measurements evaluation

Given a block, landmarks are associated with datum points on the body of character. To create a block which follows the correct measurements, the distance between two landmarks of the block and their associated measurements should be equal. Therefore, the fitness function for the measurement objective can be described as Equation 4.5.

$$Error_m = \frac{\sum_i \|M_n - Dist(l_{is}, l_{ie})\|}{n} \quad (4.5)$$

Where M_i denotes the i th measurements associated with current block. l_{is} and l_{ie} are two landmarks on the block that are associated with M_i . n denotes the number of the measurements that determines the size of the current

block b_m . Because cloth pattern is a 2D surface, euclidean distance between two landmarks is used to compare with the measurement resulted by measurement method presented in previous chapter.

Seam-lines evaluation

A cloth cannot be made into the right design without correct sewing. It is the most important method that joints two piece of textile together. Digest (2010) defines a stitch as a single loop of thread that on the textile and sewing is the craft of fastening objects using stitches.

In this thesis, a stitch refers to a constrain that attaches vertices on two patterns together and a seam-line is comprised by a group of stitches that from one end of the seam-line to another. Unless the design requires, normally, for a pair of seam-lines, the structure of both seam-lines are identical in order to form a flat and smooth pattern to pattern transition. Therefore, in this algorithm, two criteria is evaluated for seam-line evaluation, which are angle difference for each pair of point and length of corresponding edge on either side of a seam-line.

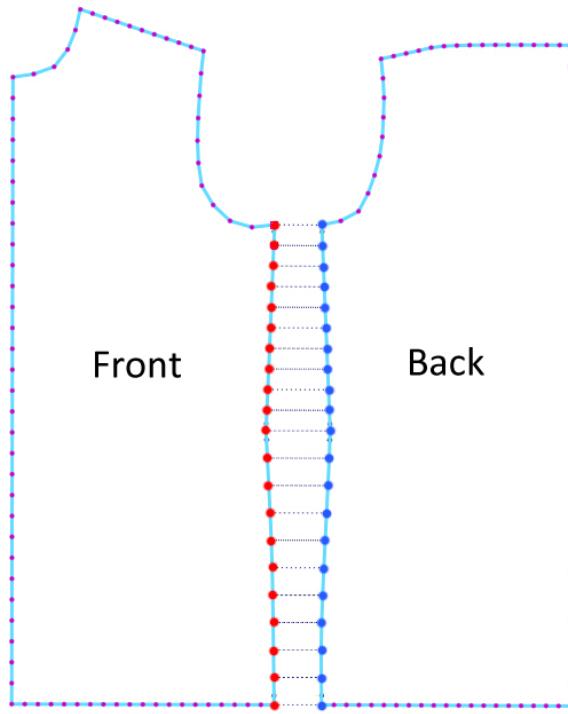


Figure 4.14: Seam-lines between “Front” pattern and “Back” pattern. The red points indicate the seam-line on the front pattern and blue points are the seam-line on the back patter.

Let $sline_f$ denotes a seam-line on “Front” pattern that depicted by the red points and $sline_b$ denotes a seam-line on “Back” pattern that depicted by the blue points in Figure 4.14. For each pair of points that connected by a constrain have the same index in its corresponding seam-line. Given a point p_{f_i} on $sline_f$, the greatest included angle θ_f between $\overrightarrow{p_{f_{i-1}}p_{f_i}}$ and $\overrightarrow{p_{f_{i+1}}p_{f_i}}$ is recorded. Next, the point at the same location on $sline_b$ is also recorded as θ_b . The total angle difference of a seam-lines on a pattern comparing to the standard input pattern is used to evaluate the angle criteria and it can be calculated by Equation 4.6.

$$Error_{angle} = \sum_{i=0}^n (\|\theta_{i_f} - \theta_{i_b}\|) \quad (4.6)$$

Where n denotes the number of point pairs in a seam-line. The other

criteria is the edge length in seam-line. For a pair of seam-lines with same arc length, different point distribution will results an undesired tension along the seam-line in which causes wrinkle to occur around the seam-line. Within every seam-line, all points are stored in an predefined order, thus in order to ensure the consistency of the distribution of points in a seam-line pairs, edge that connects two successive points in a seam-line needs to be identical to the one in the other seam-line. Therefore, the edge length criteria can be written as Equation 4.7.

$$Error_{edge} = \sum_{i=0}^{n-1} (\|l_{i_f} - l_{i_b}\|) \quad (4.7)$$

Where l_{i_f} denotes the length of edge $\overline{p_i p_{i+1}}$ in a seam-line and l_{i_b} denotes the length of edge $\overline{p'_i p'_{i+1}}$ in another seam-line. n denotes the number of points in a seam-line. Therefore, the seam-line evaluation outputs the cost as the sum of angle criteria and the edge length criteria as Equation 4.8.

$$Error_s = \frac{Error_{angle} + Error_{edge}}{n} \quad (4.8)$$

where n denote the number of point in current seam-line.

Shape evaluation

Armstrong (2000) points out a cloth design can be translated into a set of pattern with predefined shape and sewing sequence. The shape of each pattern is the critical factor that determines the shape of the cloth after patterns are assembled. The consistency of the shape for each pattern needs to be kept through out the evolution in order to ensure the cloth that fits to a character remains the same. The shape of a geometry can be described via many methods. In this thesis, the definition introduced by Kendall (1977) is used to define a pattern shape. Kendall (1977) defines shape as “all the geomet-

rical information that remains when location, scale and rotational effects are filtered out from an object”. In this algorithm, if the included angle on each corresponding crucial point is identical between two patterns, the shape of these two pattern is considered same.

In this algorithm, during the evolution, the changes of all the inner angle on every point of the contour of a block is measured as the cost of shape evaluation. Figure 4.15 demonstrates variations for each pattern, because when crossover and mutation is performed on a pattern, the location of each point on pattern contour is changed, which the shape of a pattern is changed consequently.

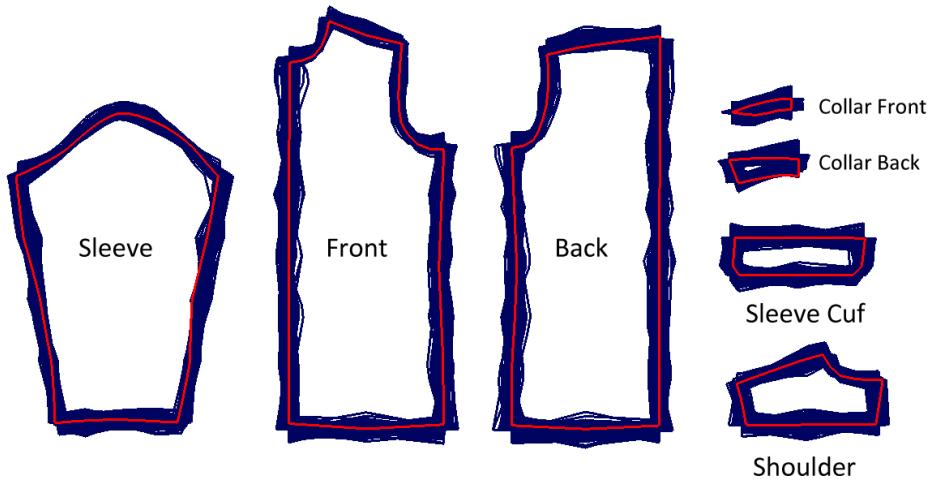


Figure 4.15: Shape evaluation for the first generation patterns, Blue lines indicate the variations of the patterns, and the patterns that are indicated by the red line has the best shape preservation.

The pattern adjustment algorithm presented in this chapter solves a multi-objective optimization problem in which objectives conflict to each other. This optimization can only results a set of solutions called “Pareto front” instead of a single best solution. Within “Pareto front”, each solution is not dominated (explained in following section) by the rest of the solutions within this set (Michalewicz 1996). However, lots of real problems in the

real world are multi-objective problems that require a single solution. The common method for finding a single solution for a multi-objective optimization problem is that, multiple fitness function are used for the construction of “Pareto front”, then a higher-level objective is used to find the “best” solution among “Pareto front”.

When adjusting a cloth, given a character, only one size of the cloth is needed for dressing the character. In order to select one solution from “Pareto front”, only one or several non-conflict criteria can be used to select the “best” solution from “Pareto front”. In this thesis, Preserving cloth style whilst fitting it to a different character is the main goal. therefore, the shape similarity between original pattern and resized pattern is used as the higher-level objective to select one solution from “Pareto front”. When evolution limit is reached, all solutions in “Pareto front” are considered as “good” solutions to the character dressing problem. In other words, all patterns in “Pareto front” has achieved good fit and all seam-lines are consistent. Therefore, at this stage, the solution which has the least shape difference comparing to the original cloth patterns is selected as the final solution of dressing character.

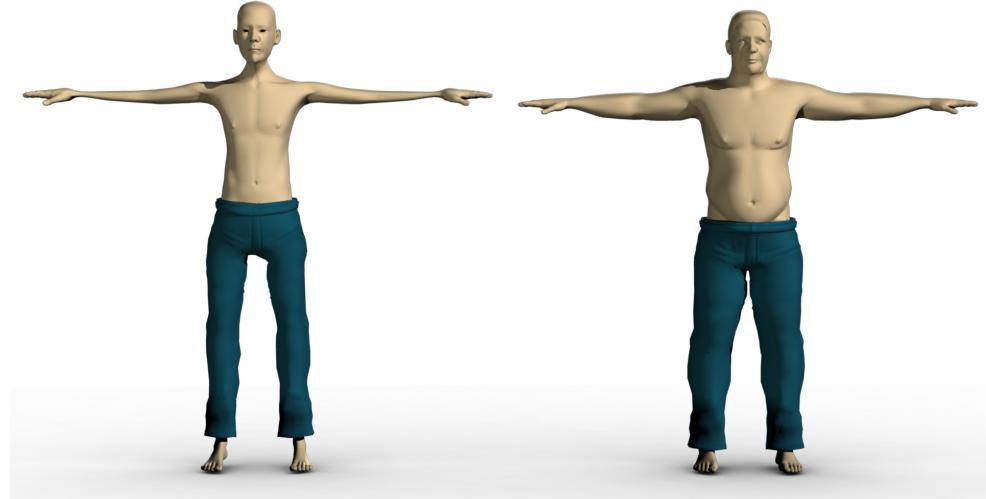


Figure 4.16: Character on the left has much thinner arms than character on the right.

For the character on the left side, because the arm is much thinner than the one on the right, the arc length of the “Arm hole” is much shorter than the one on the right. However, two character have identical height, therefore, their shirt have the same back length. Figure 4.17 demonstrates the “Front” pattern resized for the characters in Figure 4.16. The pattern is selected based on the minimum changes of inner angle of the “critical point” of the solutions. As demonstrated in Figure 4.18, the design of the shirt is preserved.

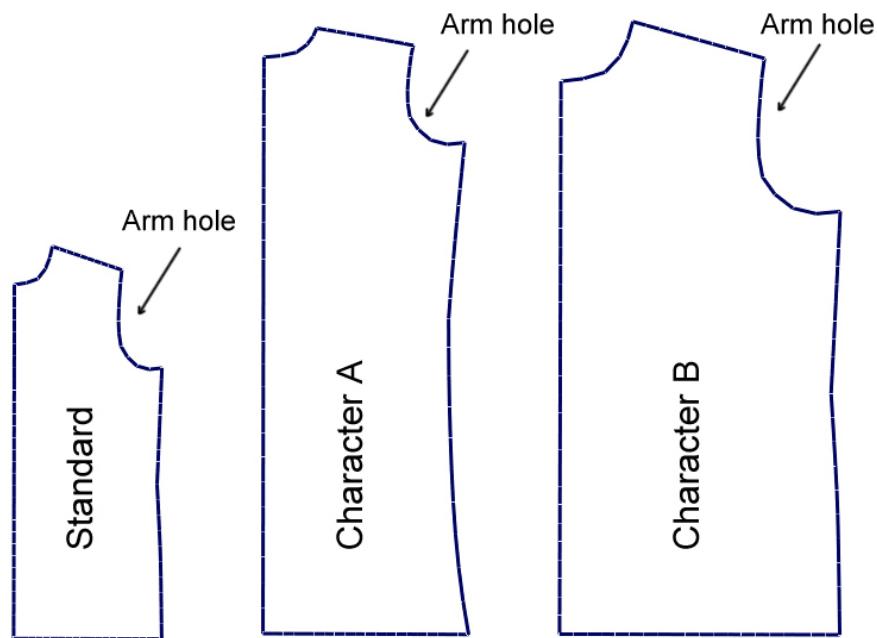


Figure 4.17: The standard pattern(left), the “Front” pattern for the character on the left in Figure 4.16(middle), the “Front” pattern for the character on the right in Figure 4.16(right)

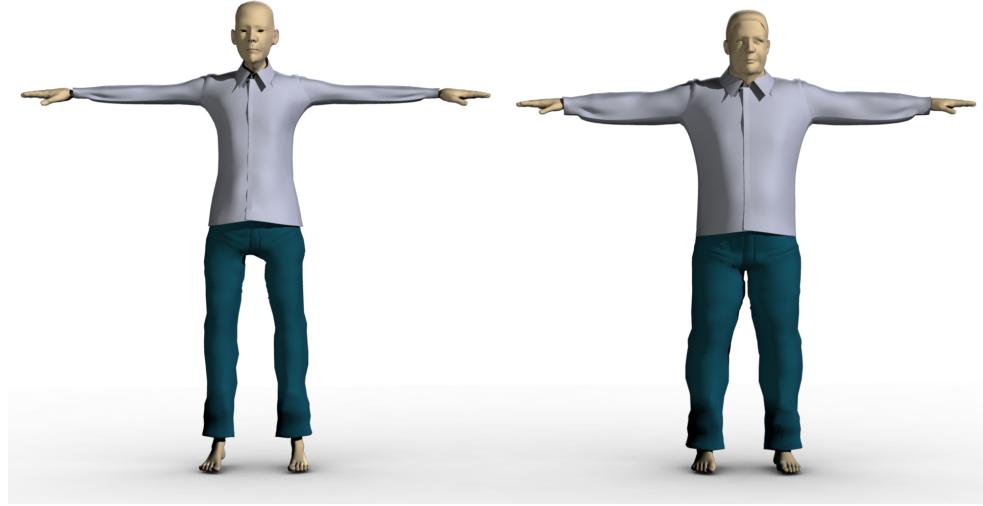


Figure 4.18: Fit shirt modelled for characters

Selection

Selection decides individuals who are fit enough to survive the natural selection and reproduce child in next generation. In single objective optimization, solutions can be sorted according to the fitness to objective. However in multi-objective optimization, because objectives are usually conflict to each other, that is, when comparing two solution A and B , A may be better than B in one objective but worse in another objectives at the same time. Therefore, the concept of domination is introduced into this area and two solution are compared by whether one is dominated by another. Deb (2001) defines the concept of domination, that is, a solution A is said to dominate solution B , if following conditions are both true:

Condition 1: The solution A is not worse than solution B in all objectives.

Condition 2: The solution A is strictly better then solution B in at least one objective.

To demonstrate the concept of domination, Figure 4.19 illustrates an optimization problem with five solutions $s_1 \dots s_5$ and two objectives f_1 and f_2 .

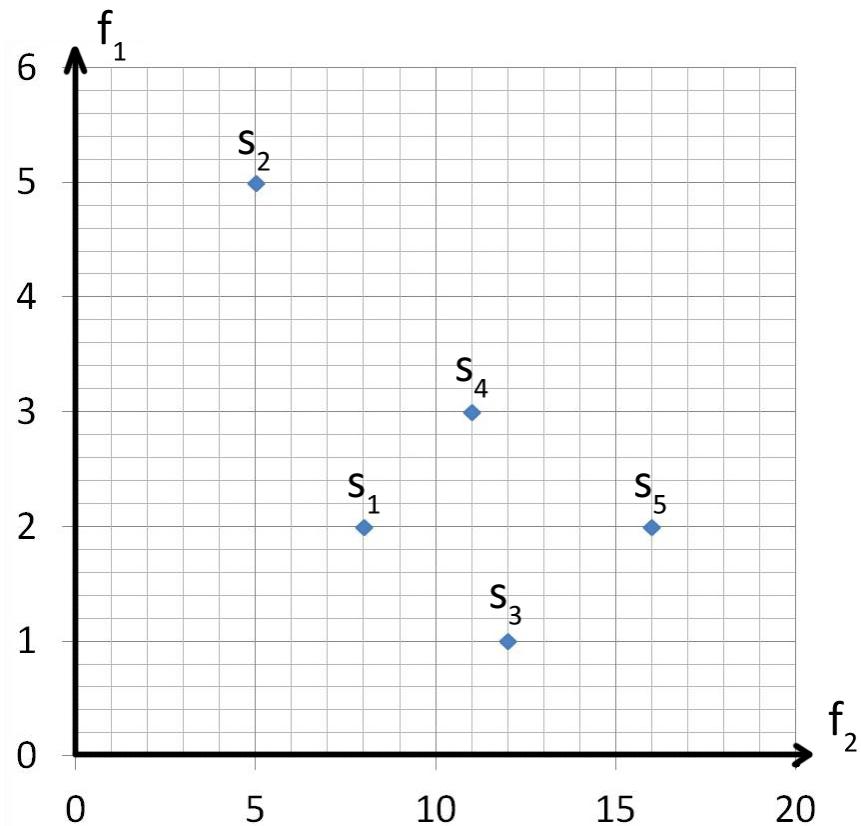


Figure 4.19: Domination between five solutions

f_1 and f_2 are two fitness functions for two objectives respectively where the f_1 needs to be minimized while f_2 needs to be maximized. In this case, two objectives f_1 and f_2 are equally important at all time, therefore it is impossible to select one best solution from this population according to both objectives. However, the definition of domination is able to decide the better solution among two solutions. As illustrated in Figure 4.19, when comparing s_1 and s_2 , s_1 is better than s_2 in both f_1 and f_2 . Thus, s_1 satisfies both conditions of the definition of domination and it can be said that s_1 dominates s_2 . When comparing s_1 and s_5 , s_5 is better than s_1 in f_2 but equals in f_1 , this situation also satisfies both two conditions of the definition of domination, therefore, s_1 is dominated by s_5 . When comparing s_4 to s_1 , s_4 performs better on f_2 than s_1 but worse on f_1 , therefore, s_1 and s_4 does not dominate each other.

Deb et al. (2002) introduced non-dominating sorting algorithm that utilises this method to select all solutions in a population that does not dominated by the rest of the population to construct “Pareto front”. To be more specific, if there are two solutions that does not dominates each other, but each of them dominates the rest of the population, it can be said that, these two solutions are “equally the best” and the “Pareto front” is constituted by these two solutions.

In this case, after comparing all pairs of the solutions, it can be concluded that s_5 dominates s_1, s_2, s_4 as well as s_3 dominates s_1, s_2, s_4 . Among s_3 and s_5 , s_3 exceed s_5 in f_1 but dwarfed by s_5 in f_2 . Therefore, s_3 and s_5 are non-dominated by each other but dominate the rest of the population. Consequently, “Pareto front” is constituted by s_3 and s_5 .

In this chapter, all solutions are compared according to the definition of domination in terms of three fitness functions, that is, measurements, seam-line and pattern shape to from “Pareto front”. In the final step, the shape objective is used for selecting one solution with less shape distortion as the final solution for constructing cloth for characters. The general process for pattern resize genetic algorithm is summarized in Algorithm 7.

Algorithm 7 Pattern Resizing

```
1: procedure EVOLVE(pop denotes a population contains multiple pattern  
variations, Npop denotes the number of individuals in pop , NGen de-  
notes the max number of evolve generation)  
2:   for gen in NGen do  
3:     parent = Selection( pop, len(pop) )  
4:     offspring = Crossover( parent )  
5:     mutated = Mutation( pop, MutPb )( MutPb denotes mutation  
probability)  
6:     for individual in offspring do  
7:       individual.fitness = Evaluation(individual)  
8:     end for  
9:     for individual in mutated do  
10:      individual.fitness = Evaluation(individual)  
11:    end for  
12:    pop = parent + offspring + mutated  
13:  end for  
14: end procedure
```

4.3 Pattern Assembling

After every pattern is adjusted into the correct shape and size based on the measurements from the character, they are ready to be placed on character and stitched together to form a complete cloth. In order to achieve such a task, all patterns are positioned onto 3D character simultaneously before sewing takes place. Three steps are involved for the construction of 3D cloth. Firstly, the bounding surface of each body part of character are created, then, patterns are triangulated and then based on the parametric coordinate of the bounding surface, all vertices on a 2D pattern are transferred onto bounding

surface to form a 3D mesh. Finally, according to the seam-line information, constraints are created to every point pair in seam-lines and physical simulation is performed to stitch patterns together to give the final shape of cloth.

Bounding surfaces are used as a guideline for transferring 2D cloth patterns into 3D mesh. Each pattern is responsible for covering certain part of character body, the bounding surfaces are used to position patterns on their corresponding body location. For human body, the general shape of each body section such as torso, legs or arms, is a cylindrical shape. For this reason, the cylindrical surface is chosen as the bounding surface. Moreover, cloth patterns are developable surface as they are cut out from a flat piece of textile materials. Therefore, construct a developable surface around the body of the character first and then transfer patterns that are also developable onto it minimize the distortion during the pattern positioning process.

In order to create bounding surface, character model is divided into six parts such as head, torso and four limbs. In this thesis, if the input character has skeleton attached to it, the body parts can be determined by the skin weight corresponding to each bone of the skeleton. If the character does not have skeleton, the body can be segmented interactively by user.

After body segmentation is completed, PCA is performed on each part of the body to calculate the principle vector of each body part. Then, by using the principle vector as the direction, central line of the each body is created as the central axis of the cylindrical bounding surface. The radius of cylindrical bounding surface is then determined by the number of pattern associated with this body part. At this step, pattern are arranged on the flattened bounding surface side by side according to position of the sewing relationship as showed in Figure 4.20. The width on the *U* direction of all the patterns is the circumference of the bounding surface. Figure 4.21 demonstrates all the bounding surfaces for a character.

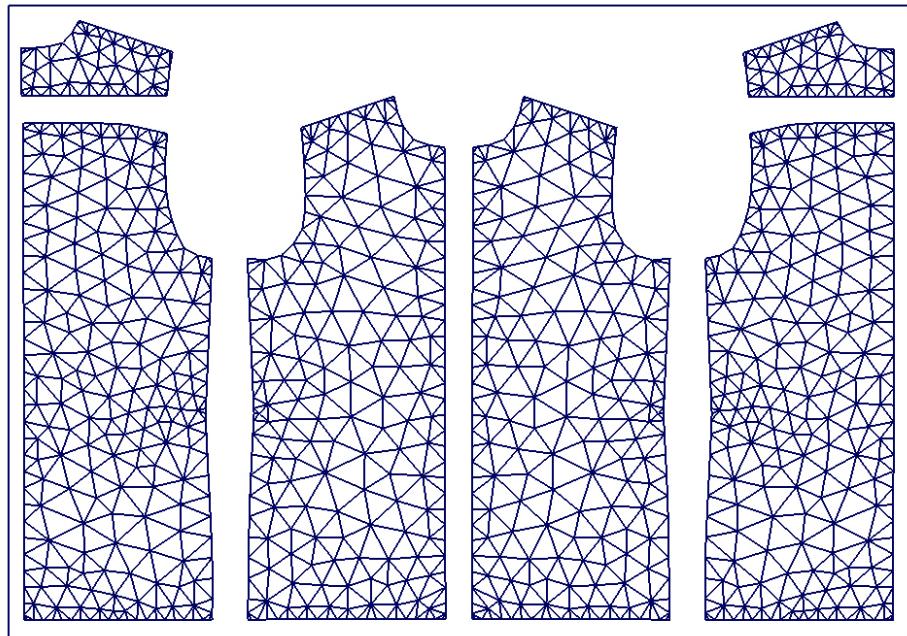


Figure 4.20: Patterns arranged side by side within the range of a box, this box is the flattened bounding surface.

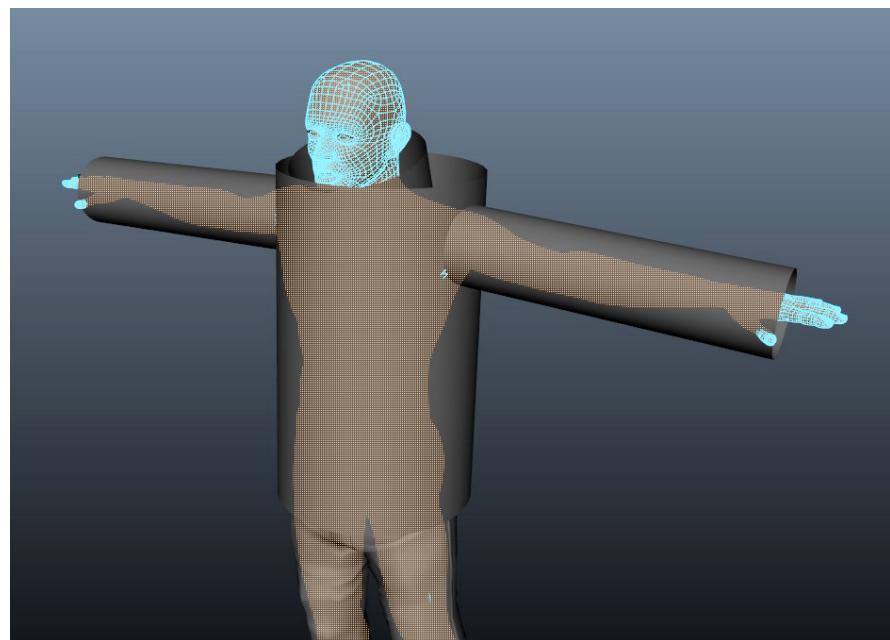


Figure 4.21: Bounding surfaces on character.

In the next step, the vertices of the 2D pattern are transferred onto bounding surface. Based on the parametric coordinate of each point of 2D

pattern on the flattened bounding surface, each point can be located by the same parametric coordinate on the cylindrical bounding surface. Finally, by applying the same topology structure to the points on bounding surface, 3D mesh be constructed from 2D cloth pattern. Figure 4.22 demonstrate the fully positioned shirt patterns of a character.



Figure 4.22: 3D patterns

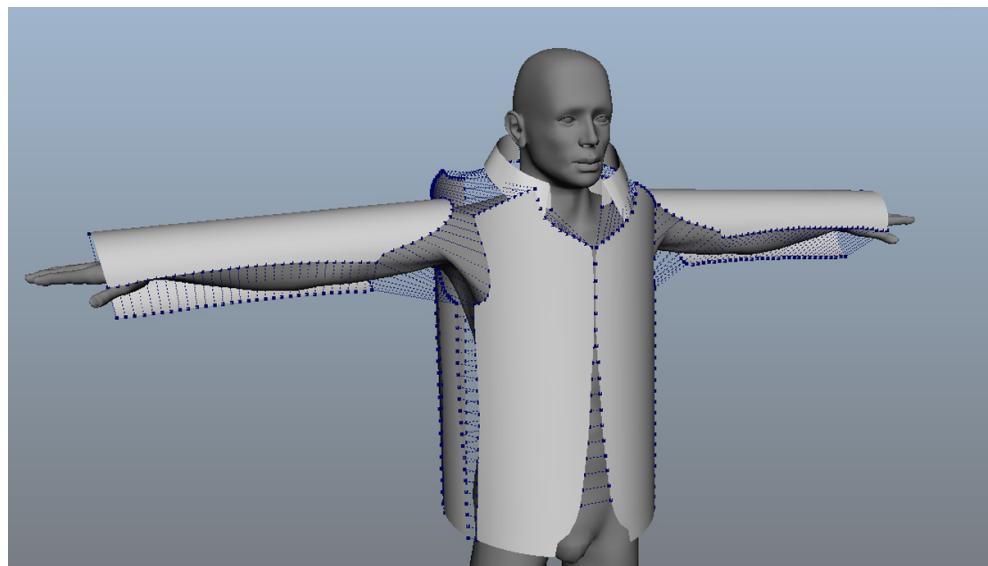


Figure 4.23: 3D pattern sewing

In the final step, based on the seam-lines between each pattern, constraints are created between two points in a point pair on seam-line. Then physical simulation is performed to pull all the patterns together. In this thesis, the nCloth module in Maya2012 (Autodesk 2012) is used to perform the physical simulation. Figure 4.23 illustrates all the sewing constrains.

In order to validate the algorithm presented in this chapter, a shirt design and a trousers design presented in Xiong (2008) are used for dressing four different characters with largely different body shapes and proportions. This experiment is performed on an Intel Xeon 3.33GHz PC with 8GB RAM running Windows 7 (64-bit) operating system and Maya2012. For cloth pattern adjusting genetic algorithm, 400 initial solutions are generated at the beginning of the algorithm. For each character, 200 evolutions are performed. Figure 4.26, 4.27, 4.28 and 4.29 demonstrate the final results for each character respectively. Figure 4.30 and 4.31 demonstrate the final cloth fit onto each character. Table 4.2 lists out the fitness value of the final solution of all characters. Notices that for character A and D, because the body proportion is largely different from the standard human body proportion that the shirt pattern and trousers pattern are based on, cloth pattern requires larger deformation to fit the character A and D than character B and C. Therefore, the result of the shape evaluation for the patterns for character A and D is worse than the patterns for character B and C.

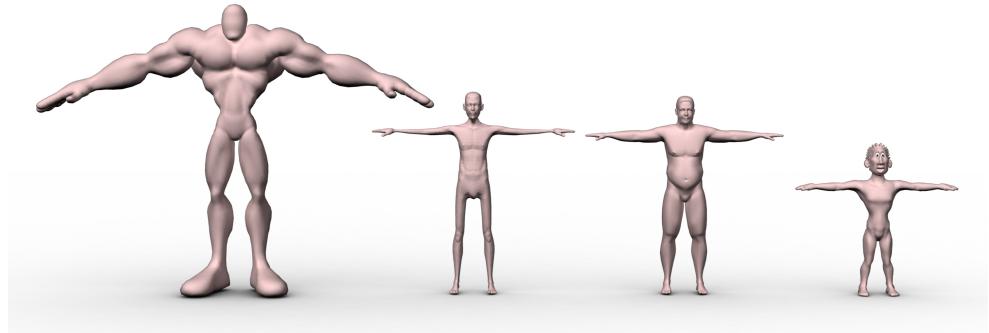


Figure 4.24: Four different characters used for the cloth modelling experiments, from left, Character A, B, C and D.

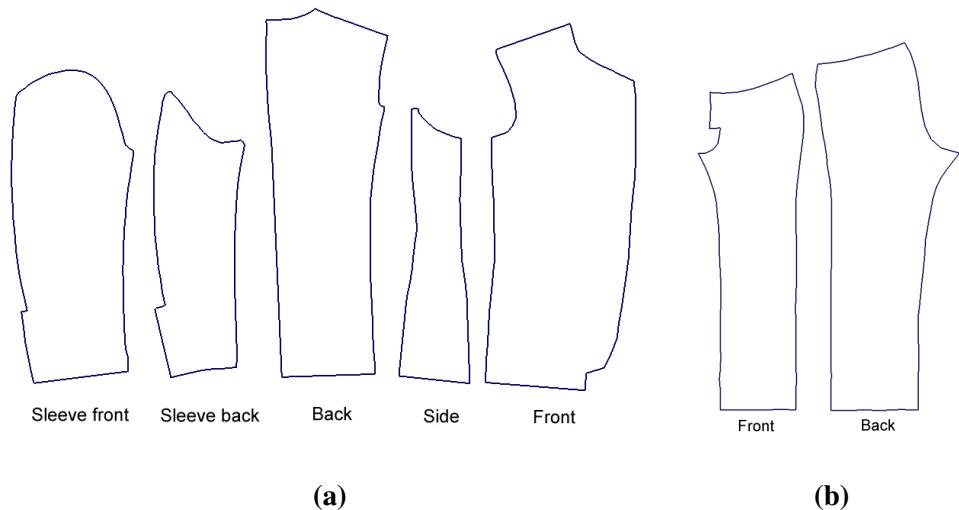


Figure 4.25: Cloth patterns(Xiong 2008), standard shirt patterns(left); standard trousers patterns(right)

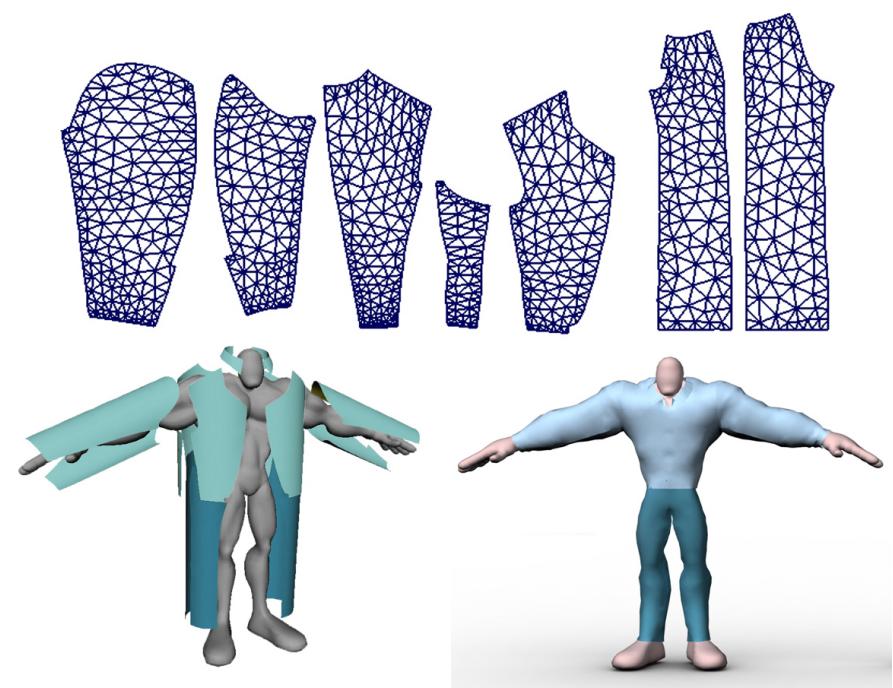


Figure 4.26: Dressing result for character A

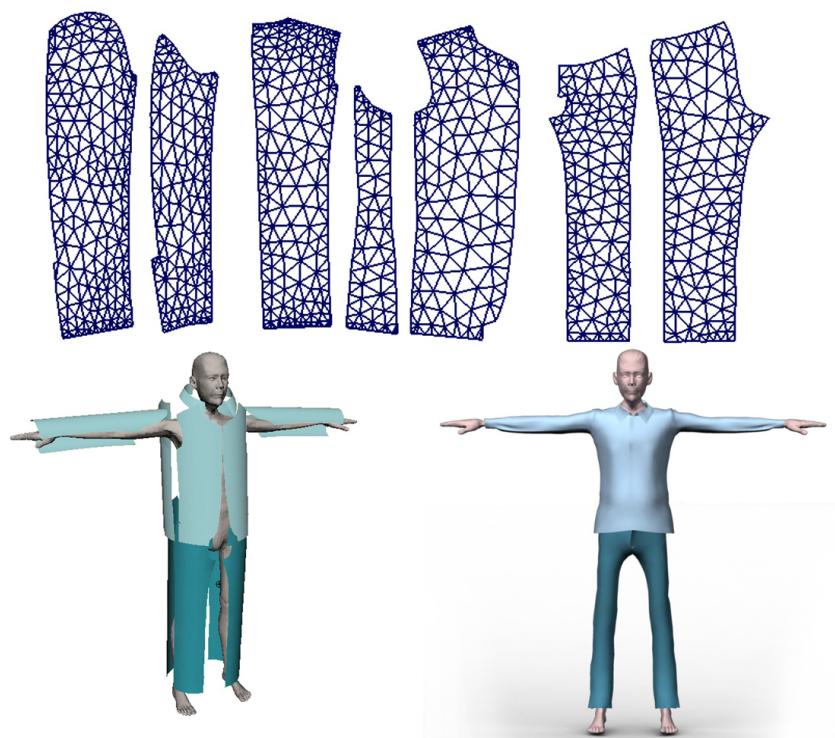


Figure 4.27: Dressing result for character B

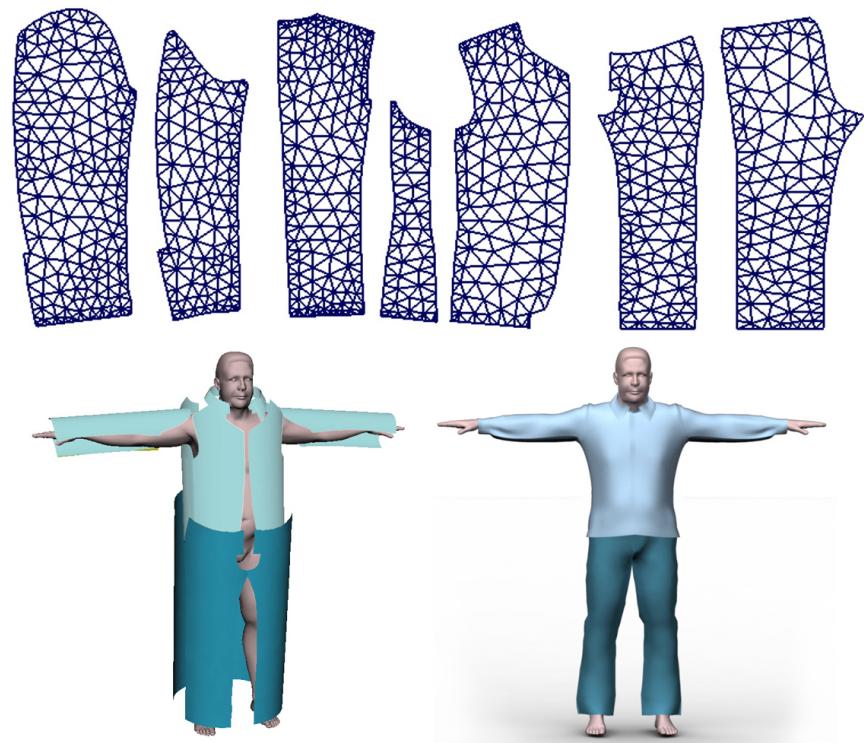


Figure 4.28: Dressing result for character C

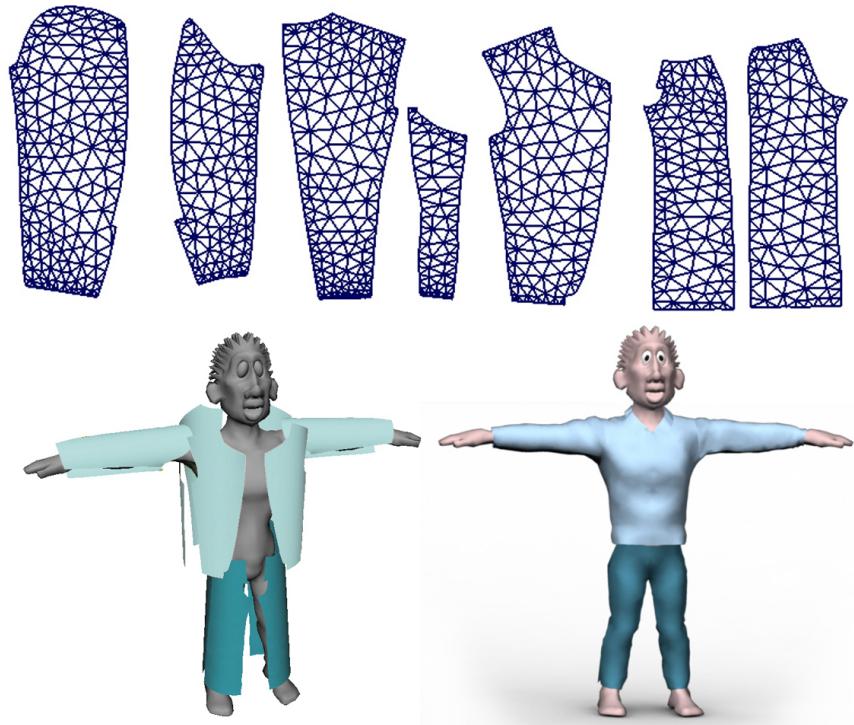


Figure 4.29: Dressing result for character D



Figure 4.30: Front view of four different characters in the same cloth design.

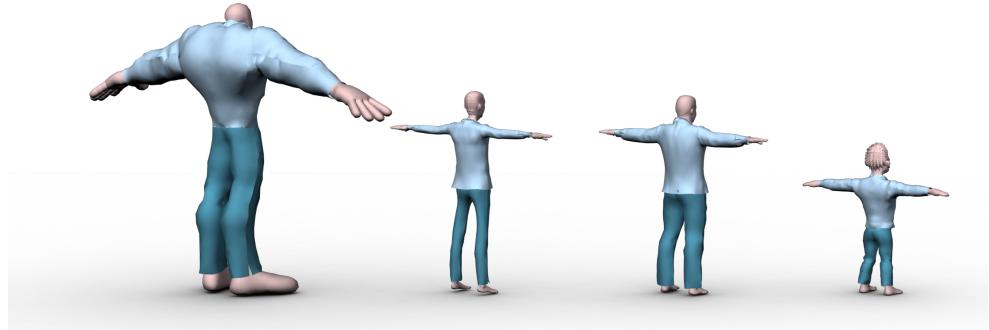


Figure 4.31: Rear view of four different characters in the same cloth design.

Evaluation function	Character A	Character B	Character C	Character D
Measurements	2.927	1.803	2.125	2.842
Seamline	1.142	1.248	1.440	1.384
Shape	0.446	0.112	0.092	0.389
Running time	784 sec	791 sec	752 sec	743 sec

Table 4.2: The fitness values of three evaluation function of the final solution for all characters and the running time for each character, unit of time is second, note that smaller value indicates better fit to the criteria.

In order to further demonstrate the modelling ability of proposed cloth modelling method, three new character is introduced and several types of

cloth ranging from female dress to t-shirt are modelled for these characters, the results are demonstrated in Figure 4.32 to Figure 4.35.

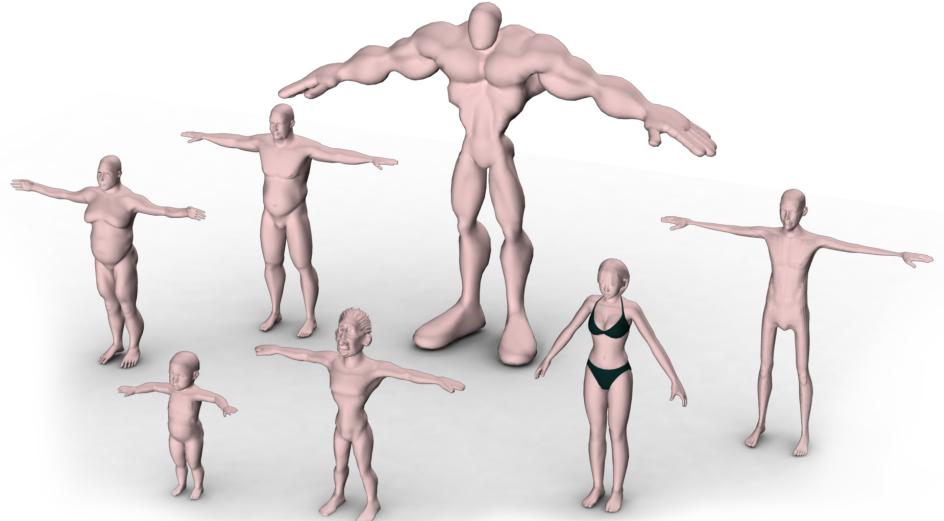


Figure 4.32: Characters used for dressing various clothes

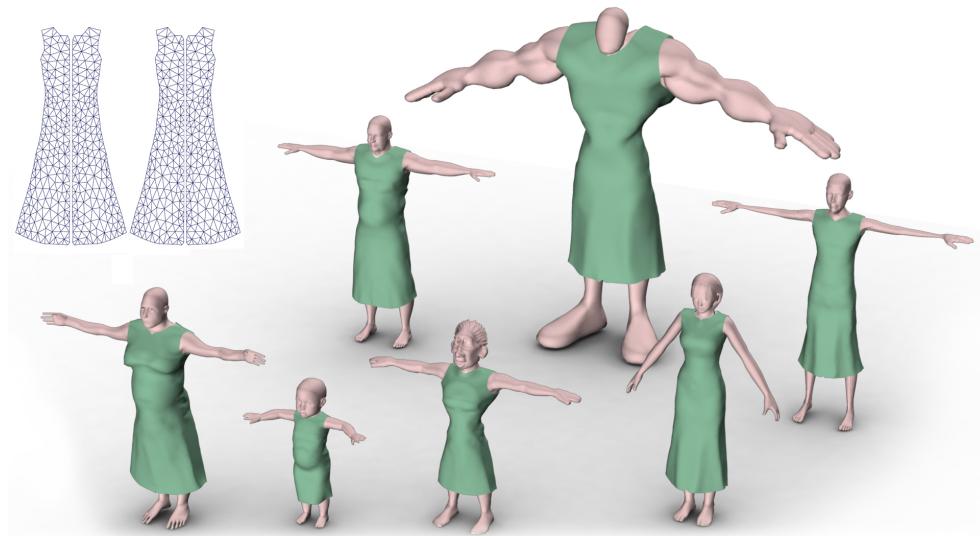


Figure 4.33: Tight dress on different characters

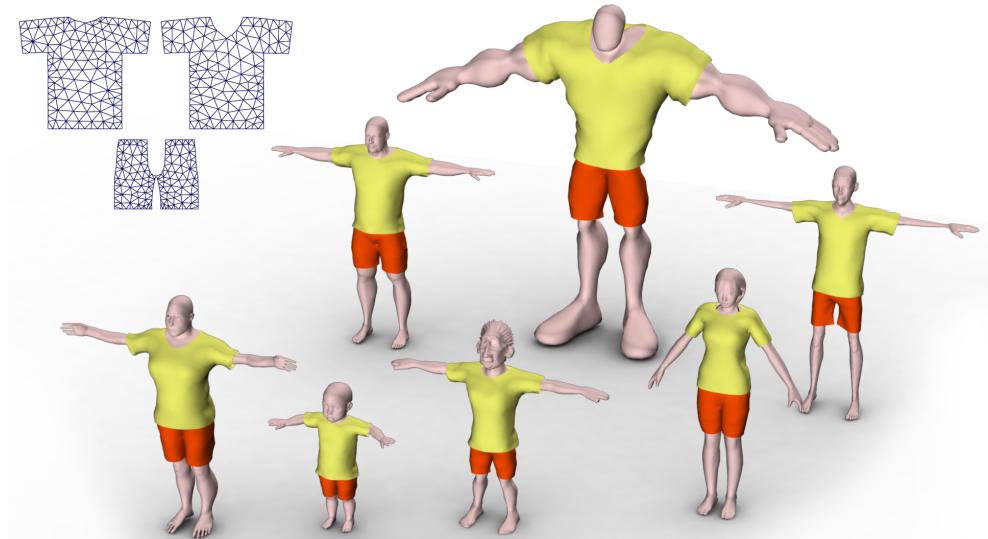


Figure 4.34: *T-Shirt and short on different characters*

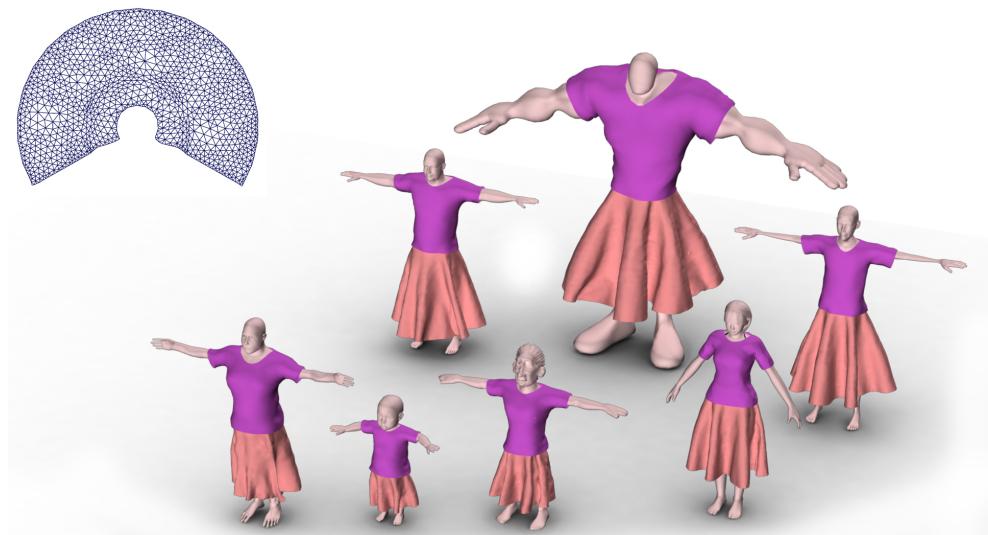


Figure 4.35: *Skirt on different characters*

4.4 Conclusions

This chapter presented an automatic pattern based cloth modelling method for dressing cloth onto characters with different body shapes and proportions. This method takes 2D cloth patterns as inputs. By extracting the measure-

ments from a character model, the shape and size of the cloth patterns are adjusted automatically to fit the cloth onto a character. During pattern adjustment, by considering the measurements, seam-line among the patterns as well as the shape of each pattern, the original design of the cloth is preserved throughout the fitting process. Finally, all patterns are transformed into mesh and positioned around the body, physical simulation is used to stitch all the patterns together based on seam-lines on each pattern to construct the final 3D cloth.

This method utilises genetic algorithm to adjust cloth patterns for cloth fitting. It has four major advantages.

Firstly, given the measurements of any character, this method can fit the cloth onto the character automatically. By automating the pattern adjustment process using genetic algorithm, the manual pattern adjustments that required by the current pattern based cloth modelling methods can be avoided. The duplication of effort required by traditional cloth modelling method can be eliminated and the efficiency for modelling cloth for different characters with different body shapes and proportion can be largely improved. This advantage shines itself even more when multiple different characters are required in the virtual environment.

Secondly, this method models cloth based on cloth design patterns without requiring tailoring expertise. By automating pattern adjustment process, deep tailoring knowledge involved in patternmaking process can be avoided. This method provides an efficient and easy-to-use solution for the animation artists. It empowers their creativity and improves their productivity by allowing them to use the large amount of existing cloth patterns in the fashion industry to create various clothes that fit different characters.

Thirdly, this method generates 3D cloth based on the adjusted 2D cloth patterns. The cloth modelling workflow is proceed in one direction. There is no turning back for extracting patterns from 3D cloth which is required

by current cloth modelling method. Therefore, the shape distortion that is introduced by 3D surface flattening process can be avoid and the extra computation for 2D pattern extracting can also be eliminated.

Most importantly, cloth are represented by 2D pattern initially, which is the unified form for all clothes. A modelled cloth can be stored in asset warehouse in the form of cloth pattern. When dressing a new character, only measurements of that character need to be recalculated and existing cloth pattern can be reused to fit a cloth onto the new character automatically. The reusability of modelled cloth is improved significantly by modelling cloth from patterns.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Virtual character has been widely used for the construction of virtual environment. It has become increasingly important in the production of films, TV series and computer games. The outfits of a virtual character is one of the most important elements constitutes the personality of characters. Today's virtual clothing techniques have been widely used in fashion industry and computer animation.

In fashion industry, cloth is constituted by a number of cloth patterns, it has provided a convenient approach for storing and distributing a cloth design. The main goal of virtual clothing application in fashion industry is to provide an efficient approach to create, manipulate and visualize cloth pattern. In film and game industry, cloth of a character is normally modelled by using general-purpose 3D modelling method, modelling cloth using this type of techniques require a large amount of manual operations. Because cloth is modelled manually, the appearance of the cloth is largely determined by the skill of modeller. With the increasing computer power, more and more characters can be simulated simultaneously in a virtual environment. Mod-

elling cloth for multiple characters with different body shapes and proportions efficiently has become an imperative task for the production of the films and games. Moreover, cloth is a soft object which follows the profile of its wearer. The design of animation character varies largely from film to film. The large amount of manual labour and repetitive works involved in modelling and modifying a 3D cloth to fit to a different character with different body shapes and proportions makes reuse a existing 3D cloth difficult. In fact, usually, clothes for a different are modelled from scratch which make the reusability of 3D cloth very low.

This thesis proposed an automatic pattern based cloth modelling method for computer animation. Current pattern based cloth modelling technique requires deep knowledge in textile engineering and tailoring expertise, few animation artists possess such a skill. Therefore, it is rarely used in the production of films and games. However, by automating the process of character measuring, pattern adjustment and pattern assembling, the presented modelling method no longer requires tailoring knowledge to adjust pattern size based on measurements. It eliminates the tediousness of traditional cloth modelling method and opens a door to animation artists to directly use a large amount of existing cloth pattern designs in fashion industry to create cloth for their characters. Moreover, each cloth design is represented by a set of 2D patterns, for a new character with different body shapes and proportions, only the body measurements need to be acquired from new characters and 3D cloth can be modelled automatically.

The cloth modelling method presented in this thesis consists of two major parts, character measuring and pattern adjustment. In order to use cloth patterns to generate 3D cloth for a character, measurements of the character need to be extracted first. Because the standard posture used for modelling character differs from person to person. The traditional anthropomorphic data acquisition method no longer suits the requirement of modelling cloth for

animation character. In order to solve this problem, a geodesic path based character measuring method is presented to simulate tape measuring in real world.

For geodesic computation, this thesis presented a novel scheme that utilises geodesic curvature flow to calculate geodesics on character model. Calculating geodesic on a high resolution model is a very time consuming task, in order to improve the computational efficiency, a linear time complexity geodesic algorithm is presented. With this algorithm, the time used for solving geodesic problem on a high resolution character model for extracting length measurements has been largely reduced.

For pattern adjustment, an automatic cloth pattern resizing method is presented. This method combines traditional tailoring techniques with modern evolutionary algorithm to generate fit cloth for characters. This method takes 2D cloth patterns and measurements of a character as inputs. Considering the measurement, seam-line among the patterns as well as the shape of each pattern, the original design of the cloth is preserved through out the fitting process. Then, all patterns are transformed into polygon mesh and positioned around the character body for assembling. Last but not least, physical simulation is performed to the positioned cloth pattern meshes for assembling. One of the most significant advantages of this method is the ability of dressing character with any body shapes and proportions automatically. By automating the process of cloth pattern adjustment, a pattern can be reused on different characters with different body shapes and proportions. Moreover, not only the tailoring expertise required by current pattern based cloth modelling method can be avoided, but also the reusability of character cloth in a animation film has been largely improved by this method.

Main contributions of the research presented in this thesis are,

1. An automatic virtual clothing method is presented to bridge the gap between traditional tailoring techniques and cloth modelling method in

computer animation. This method enables animation artists to use existing cloth design patterns in fashion industry to create 3D clothes that fits to their character automatically. Also, by extracting measurements from new characters, a cloth pattern can be reused onto different characters with different body shapes and proportions. Based on those new measurements, 3D cloth can be fitted automatically to a new character without changing the original cloth style.

2. This thesis described a character measuring method based on geodesic path and convex-hull computation. For geodesic computation, a novel geodesic curvature flow based geodesic computation scheme is introduced. The most important contribution is that, one of the proposed geodesic algorithms (Algorithm.3) has a linear time complexity with a small bounded error on triangulated manifolds. Numerical comparisons with existing algorithms (i.e. MMP, ICH_1 and ICH_2) have further demonstrated the advantages of this algorithm in terms of both speed and accuracy. By integrating Algorithm.3 into the measuring system, the time used for extracting multiple measurements from a high resolution character model has been largely reduced while the accuracy of the solution is still maintained.
3. An automatic cloth pattern adjustment method is proposed. This method utilizes evolutionary algorithm to generate fit cloth for animation character. It preserves the original cloth design by evaluating body measurements, seam-lines as well as the cloth pattern shape through out the fitting process. By automating the process of pattern adjustment, using pattern based cloth modelling technique no longer requires expertises in tailoring. Animation artists can directly use existing cloth design patterns in fashion industry to create cloth for their characters. The unintuitiveness and tediousness of the current cloth modelling method can be largely reduced. Moreover, the reusability of character cloth in

a animation film has been significantly improved.

5.2 Future work

There are several directions where the work of this thesis could be extended to and improved in the future.

1. Cloth animation

The method presented in this thesis only applies to the task of cloth modelling, it cannot generates motion of the cloth. The full cycle of virtual clothing involves simulating dynamic behaviour of the cloth, therefore, the proposed method could be improved by adding the simulation module. Data driven wrinkle generation is suggested. By creating wrinkle database for each type of fabrics, cloth patterns can be associated with a wrinkle database for a particular type of fabric. Therefore, same wrinkle database can be applied to multiple character to generate fine deformation details. This allows the efficiency of creating virtual cloth for multiple character to be further improved.

2. Automatic datum point detection

The presented measuring method requires user to manually define the datum points on a character. This method can be further improved by applying character body segmentation method (Liu et al. 2011) to identify each body part and select corresponding datum points automatically. This will also improves the efficiency of cloth modelling process.

3. Interactive cloth design

Given a set of accurate measurements, the cloth modelling method presented in this thesis can generates fit cloth for any character. However, sometimes, during fashion design, for a particular kind of body proportion, the design of cloth need to be amended for the best appearance.

Therefore, the research in this thesis can be improved by adding a pattern editing interface to allow cloth pattern to be edited by users.

Bibliography

- A. Aleksandrov & V. Zalgaller (1967). *Intrinsic geometry of surfaces*. Translations of mathematical monographs. American Mathematical Society.
- L. Aleksandrov, et al. (2005). ‘Determining approximate shortest paths on weighted polyhedral surfaces’. *J. ACM* **52**(1):25–53.
- H. J. Armstrong (2000). *Patternmaking: for fashion design*. Pearson Prentice Hall.
- R. Arnheim (1955). ‘A Review of Proportion’. *The Journal of Aesthetics and Art Criticism* **14**(1):pp. 44–57.
- S. Ashdown (2007). *Sizing in Clothing*. Woodhead Publishing Series in Textiles. Elsevier Science.
- M. A. Toups, et al. (2011). ‘Origin of Clothing Lice Indicates Early Clothing Use by Anatomically Modern Humans in Africa’. *Oxford Journals* **28**(1):29 – 32.
- Autodesk (2012). ‘Maya2012’. Software. Autodesk Inc.
- L. Backwell, et al. (2008). ‘Middle Stone Age bone tools from the Howiesons Poort layers, Sibudu Cave, South Africa’. *Journal of Archaeological Science* **35**(6):1566 – 1580.
- M. Barnard (2002). *Fashion as Communication*. Routledge.

- A. Bartesaghi & G. Sapiro (2001). ‘A system for the generation of curves on 3D brain images’. *Human Brain Mapping* **14**(1):1–15.
- P. Bose, et al. (2011). ‘A survey of geodesic paths on 3D surfaces’. *Computational Geometry* **44**(9):486 – 498.
- Brad Bird (2007). ‘Ratatouille’. Movie. Pixar Animation Studios.
- R. Brouet, et al. (2012). ‘Design Preserving Garment Transfer’. *ACM Trans. Graph.* **31**(4):36:1–36:11.
- M. P. Browne (2011). *The Practical Work of Dressmaking and Tailoring: With Illustrations*. BiblioBazaar.
- J. R. Bunch & J. E. Hopcroft (1974). ‘Triangular Factorization and Inversion by Fast Matrix Multiplication’. *Mathematics of Computation* **28**:231–231.
- J. Butcher (2008). *Numerical Methods for Ordinary Differential Equations*. Wiley.
- J. C. Butcher (1987). *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*. Wiley-Interscience, New York, NY, USA.
- Butterick (1994). *Butterick’s 1892 metropolitan fashions*. Dover Publications.
- R. Cabrera & F. Meyers (1983). *Classic Tailoring Techniques: A Construction Guide for Men’s Wear*. Fairchild Books.
- C. R. Calladine (1986). ‘Gaussian Curvature and Shell Structures’. In *The Mathematics of Surfaces*, pp. 179–196, Oxford, England. Clarendon Press.
- A. Chang (2014). ‘What Goes On Behind Those Closed Doors’. Website. <http://http://www.media-freaks.com/articles/3d-animation-studios-what-goes-on-behind-those-closed-doors>.
- J. Chen & Y. Han (1990). ‘Shortest paths on a polyhedron’. In *Proceedings*

of the sixth annual symposium on Computational geometry, SCG '90, pp. 360–369, New York, NY, USA. ACM.

M. Chen & K. Tang (2010). ‘A fully geometric approach for developable cloth deformation simulation’. *Vis. Comput.* **26**(6-8):853–863.

S. J. A. Chopp, David L. (1993). ‘Flow under curvature: Singularity formation, minimal surfaces, and geodesics.’. *Experimental Mathematics* **2**(4):235–255.

Clairaut (1731). *Recherches sur les courbes a double courbure [microform]*. Nyon, Didot, Quillau Paris.

D. Coppersmith & S. Winograd (1987). ‘Matrix multiplication via arithmetic progressions’. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, STOC '87, pp. 1–6, New York, NY, USA. ACM.

T. Cormen, et al. (2001). *Introduction To Algorithms*. MIT Press.

D. Crane (2012). *Fashion and Its Social Agendas: Class, Gender, and Identity in Clothing*. University of Chicago Press.

L. D. Cutler, et al. (2005). ‘An art-directed wrinkle system for CG character clothing’. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '05, pp. 117–125, New York, NY, USA. ACM.

M. de Berg, et al. (2008). *Computational Geometry: Algorithms and Applications*. Springer.

A. G. De Boos, et al. (1997). *SiroFast, Fabric Assurance by Simple Testing : a system for fabric objective measurement and its application in fabric and garment manufacture / Allan G. De Boos and David H. Tester*. CSIRO Division of Wool Technology Geelong.

K. Deb (1999). ‘Multi-objective genetic algorithms: Problem difficulties and construction of test problems’. *Evol. Comput.* **7**(3):205–230.

- K. Deb (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley Interscience Series in Systems and Optimization. Wiley.
- K. Deb, et al. (2002). ‘A fast and elitist multiobjective genetic algorithm: NSGA-II’. *Evolutionary Computation, IEEE Transactions on* **6**(2):182–197.
- P. Decaudin, et al. (2006). ‘Virtual Garments: A Fully Geometric Approach for Clothing Design’. *Computer Graphics Forum (Eurographics’06 proc.)* **25**(3).
- R. Digest (2010). *New Complete Guide to Sewing: Step-By-Step Techniques for Making Clothes and Home Accessories, Simplicity Patterns*. Reader’s Digest. Reader’s Digest Association, Incorporated.
- E. W. Dijkstra (1959). ‘A note on two problems in connexion with graphs’. *Numerische Mathematik* **1**:269–271.
- R. Dunn (2012). ‘Of lice and men: a very intimate history’. *New Scientist* **216**(2889):36 – 39.
- A. Ebert, et al. (2006). ‘Rule-based Morphing Techniques for Interactive Clothing Catalogs’. In G.-P. Bonneau, T. Ertl, & G. Nielson (eds.), *Scientific Visualization: The Visual Extraction of Knowledge from Data*, Mathematics and Visualization, pp. 329–343. Springer Berlin Heidelberg.
- EN:13402 (2001). *Size designation of clothes*.
- R. Enns & G. McGuire (2000). *Nonlinear Physics With Maple for Scientists and Engineers*. Springer.
- W.-W. Feng, et al. (2010). ‘A deformation transformer for real-time cloth animation’. *ACM Trans. Graph.* **29**(4):108:1–108:9.
- M. Fontana, et al. (2005). ‘3D virtual apparel design for industrial applications’. *Computer-Aided Design* **37**(6):609 – 622. {CAD} Methods in Garment Design.

- M. Frings (2002). ‘The Golden Section in Architectural Theory’. *Nexus Network Journal* **4**(1):9–32.
- A. Ghosh & S. Tsutsui (2003). *Advances in Evolutionary Computing: Theory and Applications*. Natural Computing Series. Springer.
- D. E. Goldberg & K. Deb (1991). ‘A comparative analysis of selection schemes used in genetic algorithms’. In *Foundations of Genetic Algorithms*, pp. 69–93. Morgan Kaufmann.
- R. L. Graham (1972). ‘An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set’. *Inf. Process. Lett.* **1**(4):132–133.
- J. Grefenstette (1986). ‘Optimization of Control Parameters for Genetic Algorithms’. *Systems, Man and Cybernetics, IEEE Transactions on* **16**(1):122–128.
- D. Gupta & N. Zakaria (2014). *Anthropometry, Apparel Sizing and Design*. Woodhead Publishing Series in Textiles. Woodhead Publishing Limited.
- T. Gwiazda (2006). *Crossover for single-objective numerical optimization problems*. Genetic algorithms reference. TOMASZGWIAZDA E-BOOKS.
- S. Hadap, et al. (1999). ‘Animating wrinkles on clothes’. In *Proceedings of the conference on Visualization ’99: celebrating ten years*, VIS ’99, pp. 175–182, Los Alamitos, CA, USA. IEEE Computer Society Press.
- G. Hairer (2010). *Solving Ordinary Differential Equations II*. Springer Berlin Heidelberg.
- G. M. Hannah (1919). ‘Dressmaker’s Pattern Outfit’. US Patent No.1313496.
- E. Harms (1938). ‘The Psychology of Clothes’. *American Journal of Sociology* **44**(2):pp. 239–250.
- R. Haupt & S. Haupt (2004). *Practical Genetic Algorithms*. Wiley.

- B. Hinds & J. McCartney (1990). ‘Interactive garment design’. *The Visual Computer* **6**(2):53–61.
- B. Hinds, et al. (1991). ‘Pattern development for 3D surfaces’. *Computer-Aided Design* **23**(8):583 – 592.
- J. H. Holland (1992). *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA.
- J. Horn, et al. (1993). ‘Multiobjective Optimization Using the Niched Pareto Genetic Algorithm’. Tech. rep.
- J. Horn, et al. (1994). ‘A niched Pareto genetic algorithm for multiobjective optimization’. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pp. 82–87 vol.1.
- K. Howland (2008). ‘The Merits of a Basic Fitting Pattern’. *Threads* **79**:48 – 52.
- W. Hundsdorfer & J. Verwer (2003). *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer Series in Computational Mathematics. Springer.
- T. Igarashi & J. F. Hughes (2002). ‘Clothing Manipulation’. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*, UIST ’02, pp. 91–100, New York, NY, USA. ACM.
- H. Ishibuchi & T. Murata (1996). ‘Multi-objective genetic local search algorithm’. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pp. 119–124.
- H. Ishibuchi, et al. (2008). ‘Evolutionary many-objective optimization: A short review’. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pp. 2419–2426.

- ISO/TR-10652 (1991). *Standard sizing systems for clothes*.
- B. Jiang (1998). *The Least-Squares Finite Element Method: Theory and Applications in Computational Fluid Dynamics and Electromagnetics*. Lecture Notes in Mathematics. Springer.
- I. Jolliffe (2002). *Principal Component Analysis*. Springer Series in Statistics. Springer.
- S. Katz & A. Tal (2003). ‘Hierarchical mesh decomposition using fuzzy clustering and cuts’. *ACM Trans. Graph.* **22**(3):954–961.
- S. Kawabata, et al. (1999). ‘A guide line for manufacturing “ideal fabrics”’. *International Journal of Clothing Science and Technology* .
- D. G. Kendall (1977). ‘The Diffusion of Shape’ .
- R. Kimmel & J. A. Sethian (1998). ‘Computing geodesic paths on manifolds’. *Proceedings of the National Academy of Sciences* **95**(15):8431–8435.
- R. Kittler, et al. (2003). ‘Molecular Evolution of Pediculus humanus and the Origin of Clothing’. *Current Biology* **13**(16):1414 – 1417.
- J. Knowles & D. Corne (1999). ‘The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation’. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 1, pp. –105 Vol. 1.
- J. R. Koza (1995). ‘Survey of genetic algorithms and genetic programming’. In *WESCON’95. Conference record. ’Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies’*, pp. 589–.
- E. Kreyszig (1991). *Differential Geometry*. Differential Geometry. Dover Publications.

- T. L. Kunii & H. Gotoda (1990). ‘Singularity theoretical modeling and animation of garment wrinkle formation processes’. *Vis. Comput.* **6**(6):326–336.
- Lectra (2014). ‘Kaledo’. Software. Lectra Inc.
- Q. Liu, et al. (2011). ‘Automatic body segmentation with graph cut and self-adaptive initialization level set (SAILS)’. *Journal of Visual Communication and Image Representation* **22**(5):367 – 377.
- T. F. Lynch, et al. (1985). ‘Chronology of Guitarrero Cave, Peru’. *Science* **229**(4716):864–867.
- N. M. MacDonald (2009). *Principles of Flat Pattern Design 4th Edition*. Bloomsbury Academic.
- A. Margolis (1964). *The complete book of tailoring*. Doubleday.
- Marvelous (2014). ‘Marvelous Designer’. Software. Marvelous Inc.
- B. F. McManus (2003). ‘ROMAN CLOTHING: WOMEN’.
- S. Mehta (2009). *Human Body Measurements: Concepts And Applications*. Prentice-Hall Of India Pvt. Limited.
- Y. Meng, et al. (2012). ‘Flexible shape control for automatic resizing of apparel products’. *Computer-Aided Design* **44**(1):68 – 76. Digital Human Modeling in Product Design.
- Z. Michalewicz (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Artificial intelligence. Springer.
- J. V. Miller, et al. (1991). ‘Geometrically deformed models: a method for extracting closed geometric models from volume data’. *SIGGRAPH Comput. Graph.* **25**(4):217–226.
- J. S. B. Mitchell, et al. (1987a). ‘The discrete geodesic problem’. *SIAM J. Comput.* **16**(4):647–668.

- J. S. B. Mitchell, et al. (1987b). ‘The discrete geodesic problem’. *SIAM J. Comput.* **16**(4):647–668.
- C. Moore, et al. (2001). *Concepts of pattern grading: techniques for manual and computer grading*. Fairchild Pub.
- M. Müller & N. Chentanez (2010). ‘Wrinkle meshes’. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’10, pp. 85–92, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- J. Noke (1987). *History of the Patternmaking Department, Royal Melbourne Institute of Technology 1909-1985*. R.M.I.T.
- K. Norton, et al. (1996). *Anthropometrika: A Textbook of Body Measurement for Sports and Health Courses*. UNSW Press.
- L. Nugent (2008). *Computerized Patternmaking for Apparel Production*. Bloomsbury Academic.
- R. O’Loughlin (1899). ‘Pattern for Garment’. US Patent No.632361.
- optitex (2014). ‘Optitex PDS 11’. Software. Optitex Inc.
- V. Pareto (1906). *Manual of political economy (manuale di economia politica)*. Kelley, New York. Translated by Ann S. Schwier and Alfred N. Page.
- G. Peyré & L. D. Cohen (2006a). ‘Geodesic Remeshing Using Front Propagation’. *Int. J. Comput. Vision* **69**(1):145–156.
- G. Peyré & L. D. Cohen (2006b). ‘Geodesic Remeshing Using Front Propagation’. *Int. J. Comput. Vision* **69**(1):145–156.
- S. Pheasant & C. Haslegrave (2006). *Bodyspace: Anthropometry, Ergonomics, And The Design Of Work*. Taylor & Francis/CRC Press.

- V. Pollio, et al. (1914). *The Ten Books on Architecture*. Harvard University Press.
- K. Polthier & M. Schmies (2006). ‘Straightest geodesics on polyhedral surfaces’. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH ’06, pp. 30–38, New York, NY, USA. ACM.
- T. Popa, et al. (2009). ‘Wrinkling Captured Garments Using Space-Time Data-Driven Deformation.’. *Comput. Graph. Forum* **28**(2):427–435.
- A. Quetelet (2011). *Anthropométrie Ou Mesure Des Différentes Facultés de L’Homme*. BiblioBazaar.
- G. Robins & A. Fowler (1994). *Proportion and Style in Ancient Egyptian Art*. University of Texas Press.
- S. Rosen (2004). *Patternmaking: a comprehensive reference for fashion design*. Pearson Custom Library: Fashion Series. Pearson Prentice Hall.
- I. J. Rudomin (1990). *Simulating cloth using a mixed geometric-physical method*. Ph.D. thesis, Philadelphia, PA, USA. AAI9112615.
- A. Salden, et al. (1999). ‘Linearised Euclidean Shortening Flow of Curve Geometry’. *International Journal of Computer Vision* **34**(1):29–67.
- T. Samaras, et al. (2007). *Human Body Size and the Laws of Scaling: Physiological, Performance, Growth, Longevity and Ecological Ramifications*. Nova biomedical. Nova Science Publishers.
- J. D. Schaffer (1985). ‘Multiple Objective Optimization with Vector Evaluated Genetic Algorithms’. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 93–100, Hillsdale, NJ, USA. L. Erlbaum Associates Inc.
- N. A. Schofield & K. L. LaBat (2005). ‘Exploring the Relationships of Grading, Sizing, and Anthropometric Data’. *Clothing and Textiles Research Journal* **23**(1):13–27.

- G. Schott (1992). ‘The extent of man from Vitruvius to Marfan’. *The Lancet* **340**:1518 – 1520.
- Y. Schreiber & M. Sharir (2006). ‘An optimal-time algorithm for shortest paths on a convex polytope in three dimensions’. In *Proceedings of the twenty-second annual symposium on Computational geometry*, SCG ’06, pp. 30–39, New York, NY, USA. ACM.
- H. Selin (2008). *Encyclopaedia of the History of Science, Technology, and Medicine in Non-Western Cultures*. Springer Reference Series. Springer.
- J.-A. Serret (1851). ‘Sur quelques formules relatives la theorie des courbes double courbure.’. *Journal de Mathmatiques Pures et Appliques* pp. 193–207.
- M. Sharir & A. Schorr (1984). ‘On shortest paths in polyhedral spaces’. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, STOC ’84, pp. 144–153, New York, NY, USA. ACM.
- A. Sheffer, et al. (2005). ‘ABF++: fast and robust angle based flattening’. *ACM TRANSACTIONS ON GRAPHICS* **24**:311–330.
- O. M. Shir & T. Bäck (2006). ‘Niche radius adaptation in the CMA-ES niching algorithm’. In *Proceedings of the 9th international conference on Parallel Problem Solving from Nature*, PPSN’06, pp. 142–151, Berlin, Heidelberg. Springer-Verlag.
- M. Shoben & J. Ward (1987). *Pattern Cutting and Making Up*. Butterworth.
- SmithMicro (2012). ‘Poser2012’. Software. SmithMicro Inc.
- W. Spears & V. Anand (1991). ‘A study of crossover operators in genetic programming’. In Z. Ras & M. Zemankova (eds.), *Methodologies for Intelligent Systems*, vol. 542 of *Lecture Notes in Computer Science*, pp. 409–418. Springer Berlin Heidelberg.
- A. Spira & R. Kimmel (2002). ‘Geodesic Curvature Flow on Parametric

Surfaces'. In *In Curve and Surface Design: Saint-Malo 2002*, pp. 365–373.

M. Srinivas & L. Patnaik (1994a). 'Adaptive probabilities of crossover and mutation in genetic algorithms'. *Systems, Man and Cybernetics, IEEE Transactions on* **24**(4):656–667.

M. Srinivas & L. Patnaik (1994b). 'Genetic algorithms: a survey'. *Computer* **27**(6):17–26.

N. Srinivas & K. Deb (1994). 'Muultiobjective optimization using nondominated sorting in genetic algorithms'. *Evol. Comput.* **2**(3):221–248.

E. Staff (2007). *Hand Book of Garments Manufacturing Technology*. Engineers India Research Institute.

P. Stecker (1996). *The Fashion Design Manual*. Macmillan Education Australia.

V. Steele (1998). *Paris Fashion: A Cultural History*. Bloomsbury Academic.

V. Steele (2005). *Encyclopedia of clothing and fashion*. Scribner library of daily life. Charles Scribner's Sons.

R. Stemp (2006). *The Secret Language of the Renaissance: Decoding the Hidden Symbolism of Italian Art*. Duncan Baird Publishers.

V. Surazhsky, et al. (2005). 'Fast exact and approximate geodesics on meshes'. *ACM Trans. Graph.* **24**(3):553–560.

Y. N. T. Agui & M. Nakajima (1990). 'An expression method of cylindrical cloth objects-an expression of folds of a sleeve using computer graphics'.

J. Tanner (1981). *A History of the Study of Human Growth*. Cambridge University Press.

D. Terzopoulos, et al. (1987). 'Elastically deformable models'. *SIGGRAPH Comput. Graph.* **21**(4):205–214.

- H. Thielhelm, et al. (2012). ‘Connecting geodesics on smooth surfaces.’. *The Visual Computer* **28**(6-8):529–539.
- M. B. M. T. Thomas Stumpp, Jonas Spillmann (2008). ‘A Geometric Deformation Model for Stable Cloth Simulation’. In *Workshop on Virtual Reality Interaction and Physical Simulation, 2008*.
- L. Trefethen & D. Bau (1997). *Numerical Linear Algebra*. Miscellaneous Bks. Society for Industrial and Applied Mathematics.
- N. Tsopelas (1991). ‘Animating the Crumpling Behavior of Garments’. In *Proc. 2nd Eurographics Workshop on Animation and Simulation*, pp. 11–24.
- E. Turquin, et al. (2007a). ‘A Sketch-Based Interface for Clothing Virtual Characters’. *Computer Graphics and Applications, IEEE* **27**(1):72–81.
- E. Turquin, et al. (2007b). ‘A Sketch-Based Interface for Clothing Virtual Characters’. *IEEE Computer Graphics and Applications* **27**(1):72–81.
- N. Umetani, et al. (2011). ‘Sensitive Couture for Interactive Garment Modeling and Editing’. *ACM Trans. Graph.* **30**(4):90:1–90:12.
- P. Volino & N. Magnenat-Thalmann (2005). ‘Accurate Garment Prototyping and Simulation’. *Computer-Aided Design I& Applications* **2**(5):645–654.
- P. Volino, et al. (2009). ‘A simple approach to nonlinear tensile stiffness for accurate cloth simulation’. *ACM Trans. Graph.* **28**(4):105:1–105:16.
- P. Volino & N. Thalmann (2000). *Virtual Clothing: Theory and Practice*. Springer-Verlag GmbH.
- C. Vout (1996). ‘The Myth of the Toga: Understanding the History of Roman Dress’. *Greece & Rome* **43**(2):pp. 204–220.
- B. A. Wandell, et al. (2000). ‘Visualization and Measurement of the Cortical Surface’. *J. Cognitive Neuroscience* **12**(5):739–752.

- C. C. Wang, et al. (2003). ‘Feature based 3D garment design through 2D sketches’. *Computer-Aided Design* **35**(7):659 – 672.
- J. Wang, et al. (2009). ‘Interactive 3D garment design with constrained contour curves and style curves’. *Computer-Aided Design* **41**(9):614 – 625.
- J. Weil (1986). ‘The synthesis of cloth objects’. *SIGGRAPH Comput. Graph.* **20**(4):49–54.
- Wid Ginger Software (2013). ‘Cameo v5’. Software. Wid Ginger Software Inc.
- C. Wu & X. Tai (2010). ‘A Level Set Formulation of Geodesic Curvature Flow on Simplicial Surfaces’. *Visualization and Computer Graphics, IEEE Transactions on* **16**(4):647–662.
- S.-Q. Xin & G.-J. Wang (2009). ‘Improving Chen and Han’s algorithm on the discrete geodesic problem’. *ACM Trans. Graph.* **28**(4):104:1–104:8.
- N. Xiong (2008). *World Classical fashion Design and Pattern*. Jiangxi Art Press.
- F. Yaman & A. Yilmaz (2010). ‘Investigation of fixed and variable mutation rate performances in real coded Genetic Algorithm for uniform circular antenna array pattern synthesis problem’. In *Signal Processing and Communications Applications Conference (SIU), 2010 IEEE 18th*, pp. 594–597.
- H. Yu, et al. (2012). ‘On Generating Realistic Avatars: Dress in Your Own Style’. *Multimedia Tools Appl.* **59**(3):973–990.
- G. Zigelman, et al. (2002). ‘Texture mapping using surface flattening via multidimensional scaling’. *Visualization and Computer Graphics, IEEE Transactions on* **8**(2):198–207.
- E. Zitzler (1999). *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Ph.D. thesis, ETH Zurich, Switzerland.

- E. Zitzler & L. Thiele (1999). ‘Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach’. *Trans. Evol. Comp* 3(4):257–271.