

Virtual Clothing

WENXI LI

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

Doctor of Philosophy



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

Sept, 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

This thesis investigates a framework for automatically dressing virtual characters for fit. Virtual clothing is the generic terms of reproduces both visual features and physical behaviours of textile objects in computer simulated virtual reality. Creating cloth for a virtual character involves both textile engineering knowledge, artistic expertise as well as computer graphic knowledge, therefore, it is considered as a challenging and time-consuming task. Especially when dressing multiple characters with different body proportion, repetitive manual operation is requires in order to accomplish such task, therefore this is a laborious process. Contrasted with cloth modelling, in garment industry, thanks to the cloth patternmaking and pattern grading, modern cloth manufacture is able to fit the majority of person with each cloth design. Moreover, the made-to-measure cloth bespoke technique allows the tailor to produce cloth for any customer.

This thesis reviews existing work of cloth modelling method and offers a framework to tackle the problem of dressing multiple character with different body proportion automatically. Inspired by tape-measuring technique used be tailor, a framework for geodesic curvature flow based discrete geodesic computation has been proposed for the acquisition of the measurements. Three algorithms are given for the computation of accurate geodesic paths on polyhedron; approximate geodesics on polyhedron; and approximate geodesics on point cloud. In order to modelling cloth that fit the subject, a multiple objective genetic algorithm for cloth pattern resizing has been pro-

vided. This cloth pattern resizing genetic algorithm adjust the shape of each cloth pattern based on the measurements taken from the subject respectively and meanwhile maintain the original design of the cloth.

The main contribution of this thesis is to have proposed a systematic framework to crate bespoke cloth for multiple character automatically. Results of this work can potentially provide a solution for animator who work with crowd animation as well as the fashion designer. Moreover, this thesis presented a novel desecrate geodesic algorithm that able to achieve linear time complexity and easy to be extend onto point cloud data type. This provides a method for the geodesic calculation on large model with high efficiency and accuracy.

Contents

Copyright statement	i
Abstract	ii
Table of contents	iv
List of figures	vii
List of tables	x
Abbreviations	xi
List of symbols	xii
Declaration	xiv
1 Introduction	1
1.1 Virtual Human	3
1.2 Virtual clothing	4
1.3 Motivation	5
1.4 Research objective	8
1.5 Summery	9
2 Literature Review	11
2.1 Pattern-making in fashion design	12
2.2 Anthropometry	14
2.3 Virtual clothing	18

2.3.1	Geometric Virtual Clothing	19
2.3.2	Physical Virtual Clothing	24
2.3.3	Hybrid Virtual Clothing	28
2.3.4	Summary	33
3	Geodesic Algorithm for Measurements	35
3.1	Introduction	38
3.2	Previous Work	43
3.3	Geodesic Curvature Flow	46
3.4	Geodesic on Mesh	52
3.4.1	Tangent Space Constraint	52
3.4.2	Implicit Euler scheme	54
3.4.3	Least Squares scheme	55
3.4.4	Geodesic Path Projection	57
3.4.5	”Continuous Dijkstra” Propagation	61
3.5	Geodesic on Point Cloud	71
3.6	Performance Analysis	76
3.6.1	Efficiency	76
3.6.2	Accuracy	80
3.7	Geodesic in body measurement	90
3.8	Conclusions	94
4	Virtual Cloth modelling and Re-targeting	96
4.1	Introduction	96
4.1.1	Cloth Patternmaking	98
4.1.2	Block Resizing Criteria	100
4.2	Cloth Resizing Algorithm	103
4.2.1	Genetic Algorithm	108
4.2.2	Definition of Population	110
4.2.3	Crossover and Mutation	114
4.2.4	Evaluation and Selection	120

4.3	Pattern assembling	129
4.4	Conclusion	139
5	Conclusion and Future Works	141
5.1	Conclusion	141
5.2	Future work	145
	References	146
A	Appendix Title	159

List of Figures

Figure 2.1	The peplos(left) and the chiton(right) was the common cloth wear by woman in ancient Roman (McManus 2003).	13
Figure 2.2	Body proportion in ancient Egypt	15
Figure 2.3	The Vitruvian Man	17
Figure 2.4	Catenary curves presented in (Weil 1986)	20
Figure 2.5	Wrinkle patterns in (Decaudin et al. 2006)	22
Figure 2.6	Winkle generation in (Müller & Chentanez 2010) . .	23
Figure 2.7	Deformation process in (Müller & Chentanez 2010) .	24
Figure 2.8	The simulation result presented in (Terzopoulos et al. 1987)	26
Figure 2.9	2D coordinates on triangle	26
Figure 3.1	Measurements in different posture	37
Figure 3.2	Geodesic on a smooth surface.	39
Figure 3.3	Classification of vertex on polyhedral surface	40
Figure 3.4	Geodesic on a polyhedral surface	41
Figure 3.5	FrenetSerret frame on a curve	48
Figure 3.6	Geodesic curvature on piecewise curve	50
Figure 3.7	Derived point caused by updating point.	57
Figure 3.8	Geodesic on smooth surface and its projection on the its inscribed polygon mesh	58
Figure 3.9	Projection of a geodesic path.	60
Figure 3.10	Calculating geodesic path for a unvisited vertex . . .	64
Figure 3.11	Regular grid on point cloud	76

Figure 3.12	Running time comparison for geodesic algorithms . . .	79
Figure 3.13	Error estimation of Algorithm.3	81
Figure 3.14	Error estimation of Algorithm.3 with different window size	82
Figure 3.15	Error of Algorithm.3	83
Figure 3.16	Histogram of the ratio ε	85
Figure 3.17	The distribution of the estimated errors and real errors	86
Figure 3.18	Models used for tests in Table.3.3	87
Figure 3.19	Accuracy of Algorithm.5.	88
Figure 3.20	Average relative errors of Algorithm.5 on buddha . . .	89
Figure 3.21	Algorithm.3 performs on different quality meshes . . .	90
Figure 3.22	Datum point on a female body(Xiong 2008)	92
Figure 3.23	Two circumferences measurement on character	93
Figure 3.24	Two length measurement on character	94
Figure 4.1	Garment block for woman shirt(Rosen 2004).	99
Figure 4.2	Same shirt on two characters	101
Figure 4.3	An example of the cloth pattern	104
Figure 4.4	Pattern and its block	105
Figure 4.5	Block before and after the proportional scaling	106
Figure 4.6	Blocks after proportional scaling is performed.	106
Figure 4.7	single objective genetic algorithm	109
Figure 4.8	Association between body datum point and pattern landmarks	112
Figure 4.9	Crossover of parents	115
Figure 4.10	Block mutation validation	119
Figure 4.11	The seam-line on patterns	122
Figure 4.12	Shape evaluation for patterns	124
Figure 4.13	Dressing two characters	125
Figure 4.14	“Front” pattern for both character in Figure.4.13 . . .	126

Figure 4.15 Fit shirt modelled for characters	126
Figure 4.16 Domination between five solutions	128
Figure 4.17 Centre axis for the body part on the left side of the subject.	130
Figure 4.18 Patterns in a box	131
Figure 4.19 Bounding surfaces on character.	131
Figure 4.20 3D patterns	132
Figure 4.21 3D pattern sewing	133
Figure 4.22 Characters	134
Figure 4.23 Pictures of animals	134
Figure 4.24 Patterns for Characters A	135
Figure 4.25 Patterns for Characters B	135
Figure 4.26 Patterns for Characters C	136
Figure 4.27 Patterns for Characters D	136
Figure 4.28 3D Patterns on the bounding surface of characters . . .	137
Figure 4.29 Front view of dressed up characters	138
Figure 4.30 Rear view of dressed up characters	138
Figure A.1 Models used in geodesic computation experiments . . .	160
Figure A.2 Models used in geodesic computation experiments. . .	161
Figure A.3 Models used in geodesic computation experiments. . .	162

List of Tables

Table 2.1	The basic anthropometry measurements	18
Table 3.1	Resolution of the bunny models	79
Table 3.2	Error of Algorithm.3	84
Table 3.3	Performance of Algorithm.5	87
Table 3.4	The definitions of the measurements and their associated datum points (Armstrong 2000; EN:13402 2001)	91
Table 4.1	The structure of gene	111
Table 4.2	Fitness value of final solution	138
Table A.1	The comparison of running of presented algorithms . .	163

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
$T_p S$	Tangent plane of Smooth surface S at point 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
\vec{n}	Normal vector of S
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
\vec{T}	Tangent vector of a curve
\vec{N}	Unit vector of C'

\vec{B}	Unit vector of C''
$C_{s,t}$	Geodesic curvature flow
$\overrightarrow{p_i p_{i+1}}$	A vector from p_i to p_{i+1}
$\langle A, B \rangle$	Dot product operation of matrix A and B
\times	Cross product operation bwtween two vectors
$\widetilde{p_{i-1} p_i p_{i+1}}$	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 $T_p S$
$f p_i$	Vertex on polyhedron with index i

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

Recreate world surrounds us in an interactive virtual manner has been a hot research topic studied intensively throughout the history of computer science and interactive virtual world has become a powerful tool in many research fields. Equally important as human-being to the real world, virtual human is a fundamental component to the virtual environment.

Our species, as the most intelligence creature on earth, has many distinctive feature than other animals, one of the feature that differs us from others is that, human-being is the only species that able to design and produce cloth. Cloth not only provides a shield that protects us against many hazards in the environment, which vastly widens the range that human-being able to reach even in the outer space but also satisfy our ascetic needs. Nowadays, cloth has become a major medium for self-expression.

Today, following with the high speed development of computer hardware, realistic virtual character has been wildly used not only in research area, but also in film production, TV, games and so on. On virtual character, virtual clothing is a crucial factor that directly affects the fidelity of virtual character. Moreover, virtual clothing has brought a big leap in garment manufacturing. With the help of virtual clothing techniques, this traditional labour-intensive

industry has been transferred into a highly automated modern industry.

Virtual clothing consists with two major tasks, cloth modelling and cloth simulation, as the area of application differs, the focus point of virtual clothing varies. For computer aided cloth design in fashion industry and automated cloth pattern cutting in garment manufacturing, the accuracy of cloth modelling and cloth simulation are crucial. However, in computer animation and gaming, the visual fidelity is the main objective.

This thesis focuses at the modelling cloth for multiple characters. Followed with high speed development of computer hardware and cheaper storage, more and more characters are needed in a virtual world. However, creating virtual cloth for a virtual character involves both textile engineering knowledge, artistic expertise as well as computer graphic knowledge. It is being considered as a challenging and time-consuming task. Especially when dressing multiple characters, the cloth modelling process needed to be repeated for every character. This repetitive still required when dressing multiple character with different body proportion with one cloth design.

However, in the real world, one cloth design is produced in many different size to fit different people. This process is called pattern grading. This thesis adapted pattern grading technique and made-to-measure tailoring technique from the garment manufacturing industry into computer graphic area. By using this well-developed techniques from garment manufacturing industry, the method presented in this thesis able to tailor made cloth for every character automatically regardless of their body proportion. Once the cloth is created by a set of patterns, the measurements are taken from the character and the cloth is resized based on the measurements automatically to ensure the cloth fits character, therefore largely improves the efficiency of dressing multiple characters.

1.1 Virtual Human

virtual human are substitution of real human-being in computer simulated virtual world. Cavazza et al. (1998) classified virtual human character into four categories based on the method of controlling virtual human:

1. **Pure avatars or clones**

A virtual character that controlled by a virtual reality device in real time whilst applying the data acquired by the device onto the virtual character on screen. The virtual character must appear realistically so as their motion of faces and body. This method is called the real-time roto-scopy method Thalmann (1992).

2. **Guided actors**

A virtual character that controlled by users but does not correspond directly to user's input. The medium control device calculate change of virtual character incrementally based on user inputs. The appearance of the character does not need to be realistic. Usually, the character are portrayed abstractly.

3. **Autonomous actors**

An intelligent virtual character that able to interact with the environment. This kind of character must have the ability to perceive the environment such as object or other virtual character s in its environment and the action they performed must be calculated based on the perceived information.

4. **Interactive-perceptive actors**

A virtual character that not only autonomous but also has ability to communicate interactively with other virtual characters or real people.

In general, the state of virtual human modelling can be characterized into five dimensions: appearance, function, time, autonomy and individuality, in which a wide range of application that evolves virtual human was studied in Badler (1997) such as engineering, virtual conferencing, interaction, monitoring, virtual environments, games, training, education, military and Design/Maintenance. Within those field, the fidelity of virtual human various depends on the applications. Some areas such as games or design requires high appearance and time but low on autonomy and function. While the autonomy or function of virtual human are highly required by medical or training. This thesis focus at creating high visual fidelity virtual human, therefore, the fidelity of autonomy, individuality as well as time are not in the concern.

1.2 Virtual clothing

Virtual clothing has been a research topic in the field of computer graphics for a very long time. Especially given the high speed development of computer hardware and film industry, character animation has become a major part of film. Although many methods have been developed to achieve good results in this field, new needs also exist being produced by high speed development in many related area such as garment design, computer game, and animation.

Virtual clothing is the generic terms of simulating clothes and clothing in 3D computer representations, it reproduces both visual features and physical behaviours of textile objects in computer simulated virtual reality Volino & Thalmann (2000). In general, there are two phases of work involved in virtual clothing. Modelling which constructs the geometric structure of the cloth, and simulation which calculates the changing of the cloth when external forces change. Volino & Thalmann (2000) indicates in general, two method are used for modelling cloth, firstly, geometrical modelling method,

in which cloth is modelled based on the basic geometry such as vertex and face. This method is widely used for modelling 3D objects because polygon is the most popular 3D shape representation in computer animation and game industry, and polygon is consisted by faces and vertices, therefore, by manipulating the basic element of polygon, geometrical modelling method is able to model any cloth for any character. However, this method is also the most time-consuming method. Another method is pattern based cloth modelling method, this method is inspired by cloth manufacturing techniques. Cloth are consisted by patterns and cloth are modelled by using physical simulation to assemble all the pattern together. This method does not require vertices wise manual operation however, to create the correct shape and size of patterns requires expertises in cloth making.

In real world, when wearing an item of cloth, its fit is the key that determines the degree of comfort and functionality of the cloth. In computer graphics, fit not only affects the appearance of the cloth, but also influence the physical simulation significantly because an ill-fitted cloth could cause the failure of stitching during the assembling process or penetration between the body and cloth mesh. Moreover, both methods mentioned above are not designed to model cloth for fit and to fit a cloth to a character requires significant amount of manual operation. Most importantly, duplication of effort cannot be avoided when adjusting a cloth for many different character with largely different body proportion, the modelling process need to be repeated from beginning for a new character. The duplication of effort become worse when dressing multiple character.

1.3 Motivation

Margolis (1964) indicates in general, five steps are involved in cloth making, choose the style, measure body of customer, adjust the cloth patterns,

assemble patterns and try-on. For bespoke clothing, before final assembling, patterns will be loosely stitched together and put onto customer to further trim off the excise to achieve better fitness. In 1863, Ebenezer Butterick 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 easy to access to the rapidly expanding lower middle class. This innovation also provided a standard for fashion designs to be preserved and spread with much lower cost and a design can be adapted to different human figure with ease.

In film and game industry, cloth of character is usually considered as a part the character itself. In order to complete cloth animation for a character, a substantial amount of work need to be carried out. The lack of standard for virtual cloth representation in computer graphics makes transferring a cloth from one character to another a very difficult task. In order to complete such a task, modeller usually need to start from the scratch for a new character if the body of the subject differs largely.

In the past, only few characters can be handled in a virtual scene due to the limitation of computer hardware. In order to cope with the computational limitation, the cloth of a virtual character usually considered as a second layer of skin on character. In many cases, the cloth and character cannot be separated. However, given the high speed development of computer hardware, paralleling computing technique such as CUDA and OpenMP. More and more characters are put in a scene and the number of character multiplies when crowd techniques applies. With hundreds and thousands of different characters, model cloth for every each of them individually consumes large amount of labour.

The major problem of the current cloth modelling method is that, most of the method are not designed to model from fit, a cloth is treated as a part of virtual character and can only be wear on a particular character. One cloth

cannot be putting on another character without involving large amount of manual adjustment to the cloth model. Therefore, in order to model cloth for multiple characters, repetitive manual operation can not be avoid.

However, In garment manufacturing, modern cloth manufacturing techniques has already been developed for over hundred years. Pattern based cloth making method has become a standard in this industry Moore et al. (2001). Cloth pattern is an important medium that transfers cloth design sketch into wearable object. Cloth pattern is the partitioned textile piece that based on the human body structure and cloth design. It is the fundamental parts of the cloth Moore et al. (2001); Digest (2010); Armstrong (2000). Each pattern is produced based on the measurements taken from our body. Therefore, tailor can make cloth for any customer When cloth is massive produced, based on the human body statistic data, standard body size for different ethnic are developed to standardize the cloth size, thanks for this techniques, today, we can just go into the shop and pick the cloth with the size that fits us and no matter where we go, as long as the grade of cloth correct, the cloth that we brought will be fit.

Moreover in the apparel industry there is a huge number of cloth designs that represented by patterns. However, in computer animation, an animator could not use those cloth patterns and resizing techniques directly to assist them to model the cloth for characters. In fact, as mentioned above, the major methods of cloth modelling in animation still require the animator to model every part of the cloth manually. The topology structure and modelling accuracy for cloth are seriously constrained by the workload of the animator. The research presented in this thesis is driven by such needs that is, to develop a automatic cloth modelling method that is able to takes measurements from the body of the subject and adjust all patterns respectively in order to model the cloth for an individual character. By automating the process of pattern based cloth modelling, dressing multiple character no longer requires manual

adjustment to every character. In fact, only the measurements need to be extracted from every character and the cloth modelled by the method presented in this thesis is able to generate cloth that fit to the character.

1.4 Research objective

In order to fulfil the needs of dressing multiple characters efficiently and effectively, two major problems need to be tackled.

1. Measurement acquisition

In order to model cloth for any character, fit is a fundamental requirement. A cloth can only be functional if the character who wears it is able to fit in. To create fit cloth, the common method is to manually adjust the vertices of the cloth mesh during modelling. However, this process requires large amount of user interaction. To automate this process, the principle of tailoring is adapted from cloth making technique. As aforementioned, the made-to-measure cloth making process starts from gathering length and circumference of the body parts that associated with the cloth.

Body size differs largely among people, therefore, in real world, measurements are taken by experienced tailor because the accuracy of the measurements directly determines the fit of the cloth. In computer graphics, the body shape of character varies even more because the deformation and exaggerations are the sprite of animation, a character can be designed into any body proportion. The body of a character is a complex surface and the cloth needs to follow the complex shape of the body to achieve fit.

When measuring taken place in real world, tailor applies tension on the tape-ruler in order to follow the body curvature to maximize the

measuring accuracy. The tape-ruler no longer stays straight, therefore, for the length measurements, the Euclidean distance between two end point of tape-ruler can not be considered as the length of the body part that are measured. When applying tension on tape-ruler, the tape-ruler reaches to a stable state where the elastic potential energy is minimum. Based on hulk's law, the length of the tape-rule between two end point is the shortest at this state. Therefore, the curve that formed by tape-ruler can be considered as the geodesic between two end point of the tape-ruler. In this thesis, a geodesic algorithm is developed to extract the geodesic distance between measuring points.

2. Cloth pattern grading

Cloth is consisted by a group patterns, among patterns, seam-lines are defined to indicate the assembling location of each pattern and each pattern are designed to cover a certain part of body. Because the body proportion of the character in virtual world does not like human body in which the body proportion follows certain statistical regularity Pheasant & Haslegrave (2006); ISO/TR-10652 (1991); EN:13402 (2001) therefore, patterns that adjusted based on body part can be largely different. However, for the pattern assembling, the seam-lines between each pattern need to be consistent and the original design of the cloth need to be preserved after the pattern adjustment. In order to tackled this problem, a cloth pattern resizing genetic algorithm is presented in this thesis that is able to find the best combination of adjusted patterns that fit the character.

1.5 Summery

The reminder of the thesis is organised as follow:

Chapter 2 provides an in-depth literature review of the techniques re-

lated to the research presented in this thesis. Firstly, the development of cloth pattern-making techniques is introduced. Secondly brief introduction of anthropometry is conducted which is the key problem to gain the correct human body measurement data from character model. Finally, the related work in cloth modelling and cloth simulation are reviewed.

Chapter 3 introduces a geodesic algorithm that inspired by the measuring method used in tailoring. A brief literature review of geodesic algorithm is presented. After that, three geodesic schemes that suit for different error bound and object representation are described in detail. Finally, the experiments that performed on many different model are demonstrated and comparison between our algorithm and two most popular geodesic algorithm are presented.

Chapter 4 introduces the automated cloth modelling and re-targeting scheme. To illustrate the work of the scheme, the main functionalities of the automated cloth modelling technique are outlined. This scheme mainly integrated two parts: character body measurements acquirement and global pattern resizing. The first part details the definition of datum points on character and its related measurement. Two measuring method associated with two types of measurements are introduced. The second part explains a method that evolves genetic algorithm to locate the global optimization solution for pattern resizing problem.

chapter 5 draws the conclusions of the thesis and future work are discussed.

Chapter 2

Literature Review

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

In the world of today, with its high speed development of computer hardware, realistic virtual character has been wildly used in film production, TV and games. Apart from the character's body and facial expression, same as in reality, costumes play a very important role in acting. For visual realism, techniques have been developed, such as motion capture for animation and muscle system for skin deformation, etc. However, virtual clothing, which involves both textile engineering knowledge and artistic expertise, is still being considered as a challenging and time-consuming work.

Although this topic has been studied for decades, Many techniques have been developed to create the shape of the cloth object and simulate its dynamic behaviours. Most previous work addresses these two problems in iso-

lation. In the following subsections, several reviews will be made respectively in these areas. Firstly, a general overview of pattern-making in fashion design is presented, followed by the research in anthropometry that explains human body measurements in detail. And then, the state of the art virtual clothing techniques are being introduced. Finally, according to the related research topic covered in this thesis, a brief conclusion of previous research is presented.

2.1 Pattern-making in fashion design

The creation of a garment is constituted by several interdependent yet inseparable processes. Each of the process heavily affects the appearance and fit of the garment. Within these processes, pattern-making settles among the earliest few steps in the creation of a garment. It is a highly skilled craft that has evolved over the centuries.

In the ancient Roman period, creating textile 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 structure of the cloth at that age, the cloth was mainly consisted by a set of uncut, rectangular shaped fabric pieces to minimize waste (Vout 1996).

In the fifteenth century the seminal art of patternmaking began. The fabric was carefully trimmed to fit the contour to the body (MacDonald 2009). The foundation of modern fashion design was built since then. Prior to the Industrial Revolution the art of patternmaking was highly revered. Tailors meticulously worked with their client's personal measurements to customize patterns. Clothing made by tailors was elaborate and relegated only to the very rich. With the onset of the Industrial Revolution, standardized patterns were essential to the success of ready-to-wear clothing. Initial attempts to cre-

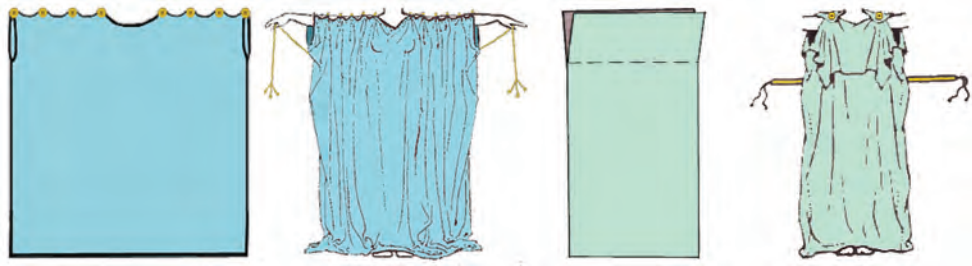


Figure 2.1: *The peplos(left) and the chiton(right) was the common cloth wear by woman in ancient Roman (McManus 2003).*

ate standardized patterns resulted in poorly fitting garments with little detail. Men's suits were boxy, plain, ill-fitting sacks. After lengthy experimentation and standardized sizing, patternmaking made a triumphant transformation from customization to standardization (MacDonald 2009). During that time patterns are made into one size, the sewer had to grade the pattern to the size that was needed. Because pattern grading requires lost of expertise and experiences, cloth pattern was not widely accepted in home sewing. The fashion was a phenomenon exclusive to ladies of high standing. Because only professional tailor acquires the skill of pattern grading and only the people from upper class could afford to pay for the latest styles from Paris, New York and other fashion centres(Gorsline 1993). In the end of 19th century, Butterick invented the graded patterns(Company 1994). This revolutionary idea opens up the market of home sewing and the effects of graded pattern were significant. With the advent of Butterick patterns, not only did dress-making become much easier, fashion became available to men, women, and children of all classes all over the world. With the help of the development of anthropometry, the graded pattern could fit the general public better than before, and the techniques of patternmaking used in today's garments manufacturing industry are developed from Butterick patterns(Noke & of Technology. Patternmaking Dept 1987; Rosen 2004).

Today's patternmaking has two general castigators based on the differ-

ent application, one is patternmaking for massive production(Nugent 2008; Shoben & Ward 1987; Stecker 1996; Staff 2007), the other is patternmaking for custom tailoring (Margolis 1964; Cabrera & Meyers 1983; Browne 2011). The patternmaking in massive production mainly focus at ease of distribution and standardization, the patterns are graded into different size in order to provide clothing for general public. On the other hand, in custom tailoring, pattern is distributed as an one size design concept, the final cloth requires experienced tailor to reproduce the pattern based on the measurement from the subject and product the fit cloth. In this thesis, in order to fit costume onto any character, the techniques from custom tailoring is adapted into computer graphic. Based on the measurements from subject, the cloth pattern are adjusted into particular size that follows the patternmaking principle of custom tailoring to ensure the fit of the cloth for the subject.

2.2 Anthropometry

Fitness is one of the essential factors that directly determines the functionality of the garment. To achieve fit, measurements of the body of the wearer need to be acquired. Anthropometry is the branch of human science that study the measurements of body size, shape, mobility, flexibility and strength. Human body dimension, as ours personality, is largely varied among the population. Many user-centred applications require the understanding of this variability. Especially for the 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 measurement. This section will debrief the research achievement in anthropometry and introduce the method that acquires the anthropometry data.

Research of Human physical stature was the first topic in the anthropometry that was studied systematically. The history of it can be traced back

to 18th century (Tanner 1981). However the recognition of human proportion is far earlier than it. In ancient Egypt, when painting human figures, a modular grid was employed to the preparation of some painters drawings (Pheasant & Haslegrave 2006). This modular grid consists of 18 units from the crown of the skull to the foot. , Figure.2.2 demonstrate three ancient Egyptian figures that have 18:11 relationship between the height of the hair-line and navel(Robins & Fowler 1994).

This separation provides a consistent point upon which a figure's proportions could be based on. (Robins & Fowler 1994) points out that by laying a hypothetical grid over figures from early dynasties it can be demonstrated that their proportions are identical to those of later dynasties.

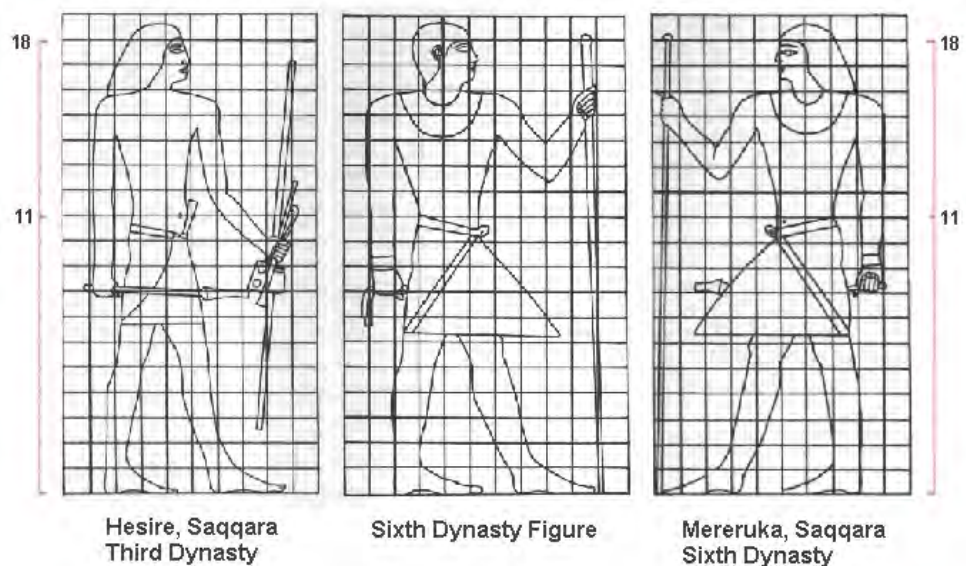


Figure 2.2: *Body proportion in ancient Egypt*

This modular system evolved initially as a drawing standard that still being used today. The most detailed system of human proportions that today's anthropometry researches built on is from the 15 B.C the Roman architectural theorist Vitruvius (Selin 2008). His theory of human proportions is well known to us as a "well-shaped man" (Arnheim 1955), in which the

height of the human stature are equal to its 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 Vitruvius's human proportion visualization is the drawing created by Leonardo da Vinci (Stemp 2006). This piece is accompanied by the notes that based on the human proportion theory developed by Vitruvius. It depicts a male figure in two superimposed postures that the arms and legs of the male are circumscribed by a circle and square respectively. When this piece was created, the theory of human proportion had become bound up with the "golden ratio". That is the umbilicus divides the stature of the 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 greater part (Stemp 2006).

However, the study carried out by Vitruvius was based on the Roman population 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). The study on anthropometry was not systematically carried out until 1870. 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. In a very long time after the foundation of this science, anthropometric was mainly used for taxonomic or physiological studies. The development of this science accelerated during WWII powered by aircraft industry due to the needs to design better aircraft cockpits. Even today, much of the anthropometry researches are initiated by military industry.

The measurements involved in modern anthropometry varies from field to field. In general, there are 7 measurements illustrated in the table below:

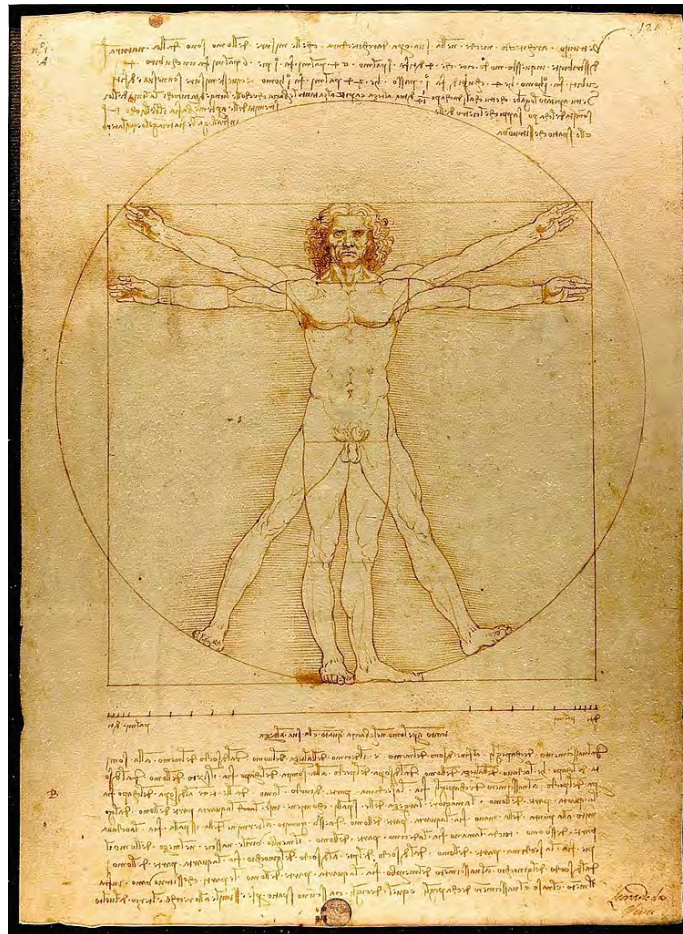


Figure 2.3: *The Vitruvian Man created by Leonardo da Vinci in 1490 (Stemp 2006)*

Driven by the industrialization of the early nineteenth century, the development of standardization reached to an unpretentiously speed. In order to largely produce cloth for the general public, the grading system was introduced to the cloth making industry. This is the process that used by clothing manufacturers to produce garments in a range of sizes. Grading is a standard method of applying increases and decreases to make the pattern larger or smaller.

Generally, there are three steps involved in defining a sizing system (Schofield & LaBat 2005): Firstly divide general population into several cat-

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

Table 2.1: *The basic anthropometry measurements*

egories of body types with similar characteristics. Then a body measurement is selected as a primary size interval for each category, finally choose the intervals for remaining body measurements for each category. Gupta & Zakaria (2014) indicates that, one of the greatest challenges for the apparel industry is to produce garments that fit customers properly, especially for the massive produced cloth, anthropometry provides crucial data for the design of sizing system (Norton et al. 1996; Samaras et al. 2007; Mehta 2009).

2.3 Virtual clothing

Followed by wide use of virtual human in many fields, virtual clothing has became a hot research topic that many researches have been done to improve the realism or efficiency of the result. Since there are many processes that involved in a complete reproduction of cloth in virtual world, the computational ability of the hardware is the biggest constrain that limits the quality of the modelling and simulation. When research on virtual human just started, to prevent naked human model, the cloth was usually mimicked by a group of texture that mapped onto the human model. Later, to improve the realism of the cloth, 3D modelled cloth mesh was introduced that cloth are modelled as a part of the character, and the skin that covered by cloth were discarded to

reduce the computational requirement. In terms of cloth animation, it is directly driven by the same joint system that drives the skin of the character.

After many years of research, today's virtual clothing methods are much different than before. In general, according to the core technique that determines the deformation method of virtual clothing, modern virtual clothing technique can be classified into three categories: geometric virtual clothing, physical virtual clothing and hybrid virtual clothing.

2.3.1 Geometric Virtual Clothing

Geometrical method can be traced back to 1986. It was widely used for cloth modelling in the early age. Weil (1986) presented a method to generate a hanging cloth using geometrical techniques, the cloth was represented by a grid of vertices and the shape of cloth was generated from catenary curves between the hanging points. Figure 2.4, it is not suitable for dynamic simulations but works very well for stationary or single-frame renders. This technique creates an underlying shape out of several hanging points. Then it passes through each set of three 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 the render. This method can only simulate certain shape of hanging object and it cannot be used to simulate the whole piece of cloth.

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

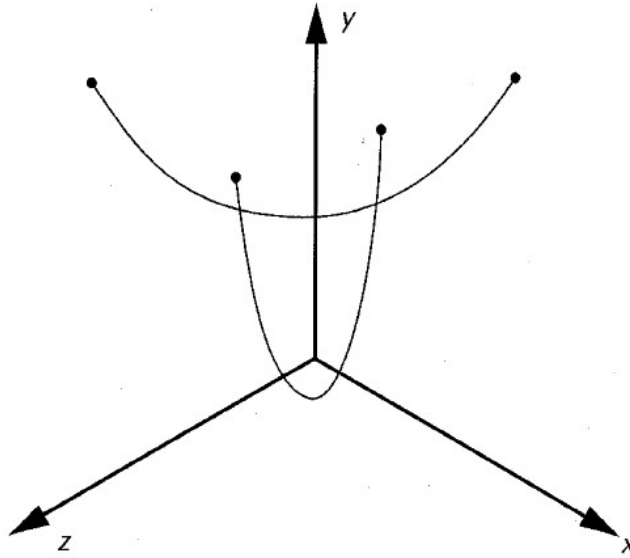


Figure 2.4: *Catenary curves presented in (Weil 1986)*

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 of translate 3D pattern into 2D patterns by using the method presented by Calladine (1986).

Miller et al. (1991) presents a method named Geometrically Deformed Model. This method is developed 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. Then based on a set of constraints, this simple object is deformed to fit the object within a volume. At the same time, the deformed object maintains its closed and non-self-intersect. The major advantage of this method is that the sampled data are aggregated by placing geometrical relationship on the model. This allows a initial convex model to be transformed into a concave object.

Since the computational cost are associate with number of the constrain that requires during the deformation, the level of detail can be easily contorted. Later, (Thomas Stumpp 2008) extend this method into cloth simulation. In their method, the cloth mesh are partitioned into triangular clusters. Using these triangular clusters, the optimal rotation can be calculated efficiently for the shape matching presented in (Miller et al. 1991) without iterative calculation for polar decomposition. This method advanced itself with high efficiency, according to the experiment section in this paper, the computation time reached linear time complexity in terms of number of the mesh vertices.

Decaudin et al. (2006) presents a geometric method for modelling the cloth by warp the developable surface around the character's body in a natural manner to create the visually realistic wrinkle. By flattening the developable surface, this method can also provide the distortion-free sewing patterns for real industry use. In this method, the first step is to use a sketch based interface to generate an initial 2D surface from the contour of the cloth drawn by the user. Then by using distance field technique, the surface is converted into a 3D surface. After that, the seam line is added directly by hand and drawn on the surface. Each surface panel is enclosed by seams while all the panels are kept assembled along the seam line. For the wrinkle handling, in this paper, the wrinkles are categorized into two types, diamond wrinkle which caused by axial compression and aligned wrinkle which caused by twisting Figure.2.5. They assumed all the wrinkles occurring on the cloth are different combination of these two types of wrinkles. A procedure buckling pattern is developed to create the natural shape for the wrinkles on the cloth. However, limited by using developable surface to recreate the wrinkles and the numbers of the wrinkle pattern, it can only produce rough shape of wrinkles, which means it cannot reproduce the wrinkle with fine details on it. Moreover, to create a huge variety of wrinkle pattern requires significant large amount of parameter, which will leads to a very heavy computation.

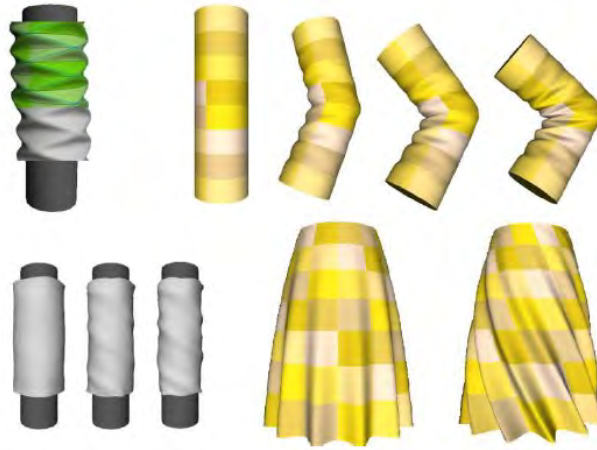


Figure 2.5: Two wrinkle pattern presented in (Decaudin et al. 2006). First row demonstrates the diamond wrinkle pattern and its derivative. The second row is the aligned wrinkle and its derivative.

(Müller & Chentanez 2010) introduced a geometrical approach to add fine detailed wrinkles onto dynamic simulated mesh such as cloth or skin. The approach presented uses a higher resolution wrinkle mesh as a reference which attaches to the lower resolution mesh Figure.2.6. The low resolution mesh is composed of the index of its triangle and a pair of barycentric coordinates that defines a point in the base triangle. This allows the wrinkle vertices on lower resolution mesh to deviate from their attachment positions within a limited range. The method presented in (Muller et al. 2004) is used for wrinkle generation. The edge splitting method is used iteratively to base coarse mesh till the longest edge falls below a given threshold.

Chen & Tang (2010) introduced a method for simulating inextensible cloth using fully geometric approach that is subjected to a conservative force (e.g., the gravity) and collision-free requirement. This paper points out that for a piece of cloth, its stretch resistance and compression resistance are many times larger than its bend resistance. Therefore, the simulated cloth can be considered as an inextensible object. However, traditional physically-based

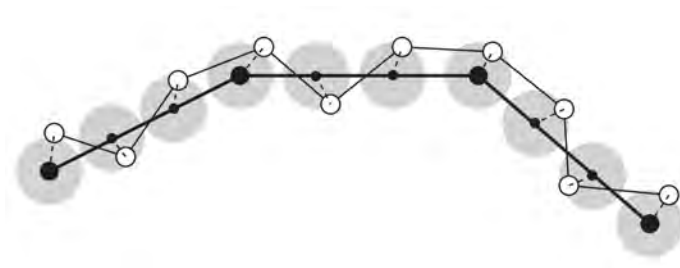


Figure 2.6: *The black dots and thick lines are the vertices and edges of the low resolution base mesh, the white dots and thin lines represents high resolution wrinkle mesh. The grey region is the constrain area for controlling the movement of white vertex (Müller & Chentanez 2010)*

cloth algorithm cannot perform correctly if some of the stiffness coefficients are set 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. The method introduced in this paper is an iterative process that based on energy minimization. Gauss-Newton iteration is used for the minimization of the energy function. However, this method can only handle the conservative force, it cannot handle non-conservative force such as friction. The other crux is that since it uses energy minimization to determine the final shape of the cloth object, the simulated cloth object can only achieve to one steady state therefore it is not suitable for dynamic use.

Geometrical cloth modelling method maybe fast on generating the static cloth shape, but cloth is a soft object whose shape is closely associated with the shape of body underneath. In addition, when it comes to the detail of the wrinkle, the number of the perimeters needed to create it, in which the computational cost will increase dramatically. However, physics can generate the detail wrinkle easily and can also reproduce the kinetics property of textile very well.

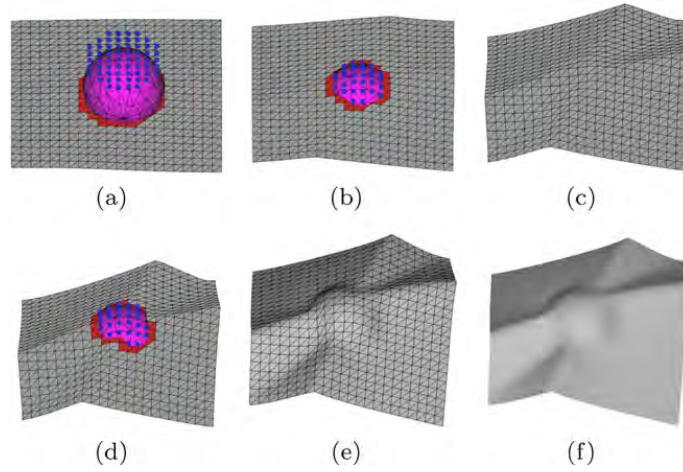


Figure 2.7: blue points are the dynamic anchor points, (a), (b), and (d) are three intermediate states in sequence. Although collision occurs the number of dynamic anchor points are reduced gradually; (c) and (e) are the two sequent intermediate states where the potential energy of the mesh is lowered gradually. (f) is the final state where the potential energy reach to the minimum (Müller & Chentanez 2010)

2.3.2 Physical Virtual Clothing

Physical method represents the cloth, using a set of particles connected to each other by springs, which can replicate the behaviour of the soft flexible object such as textile. The number of neighbourhood particles varies according to the different techniques it use. Generally, there are two kinds of model for the physical cloth modelling technique, Energy-based techniques and the Force-based techniques. Energy-based model calculate the total energy of the cloth using a set of energy equations.

Those equations determine the shape of the cloth by moving the vertexes (particles) to achieve a minimum energy state. This kind of method is widely used in static simulations. Because the energy model is based on the kinematic theory in which the shapes are compositions of geometrically or algebraically defined primitives. And it does not interact with each other or with external forces. By this reason, the Force-based technique has been

brought in to this field to describe the reaction to external forces. It usually uses differential equation to represent the force among each vertex and perform a numerical integration to locate vertex position at each time step. This kind of methods is used in dynamic simulations.

Terzopoulos et al. (1987) introduced a method by using the physically-based model to construct the shape of a cloth object. Their method is based on the elasticity theory used to represent the shape and motion of deformable materials, and moreover it has the ability to be interactive 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 see Figure.2.8. Moreover, by bring the physical properties such as mass and damping in to simulation, it can simulates the dynamic of these objects. *"The simulation involves numerically solving the partial differential equation, govern the evolving shape of the deformable object and its motion through space."* (Terzopoulos et al. 1987). However, this method can only simulate interaction between simple shaped objects, it does not have the ability to simulate cloth wearing by the character.

Volino & Magnenat-Thalmann (2005) introduced a general mechanical model for cloth simulation. This model uses an accurate particle system model for dynamic simulation instead of mass-spring system because the mass-spring system cannot represents the anisotropic nonlinear mechanical behaviour of textile object. By defining three main mechanical properties of cloth: weft and warp elongation and shear, a triangle face of cloth mesh is described by three 2D coordinates correspond to these three properties. Those three coordinates describes the location of the triangle's vertices on the weft-warp coordinate system that defined by the directions U and V with an arbitrary origin Figure.2.9. This particle system model is able to simulate

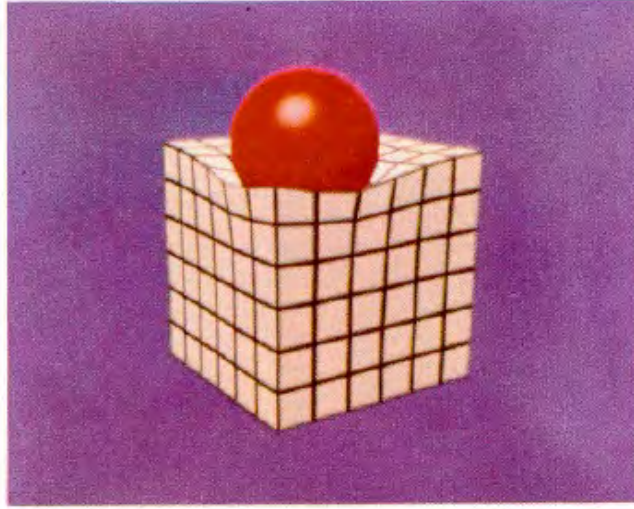


Figure 2.8: *The simulation result presented in (Terzopoulos et al. 1987)*

the anisotropic nonlinear mechanical behaviour of a cloth object using polynomial spline approximations of the strain-stress curves. Furthermore, the aforementioned three mechanical properties can be measured from real cloth piece using Kawabata Evaluation System (KES) of the SiroFAST method. Since the high accuracy their system is able to produce, it suits the requirements of garment design and textile engineering.

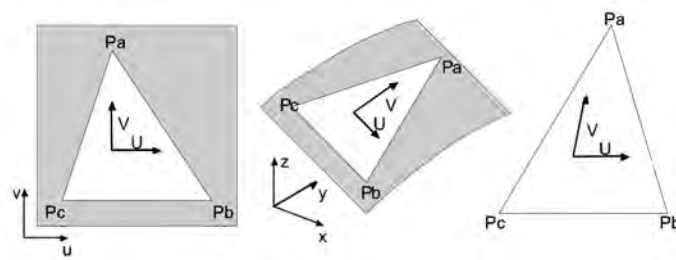


Figure 2.9: *A triangle face of cloth surface defined on the 2D cloth surface(left). The deformed triangle in 3D space(middle). The deformation of the weft-warp coordinate system of this triangle face(right).*

(Volino et al. 2009) proposed a simulation model for large deformations of textile. It can reproduce the nonlinear tensile behaviour of textile with ac-

curacy and robustness. Meanwhile the process of simulation remains simple and which is suitable for high-performance simulation system. Differing from the majority of existing cloth simulation systems, with this model, the weft, warp and shear tensile behaviour usually considered in the traditional textile engineering are represented by accurate strain-stress curves. This model also includes the elasticity and viscosity which makes it suitable for dynamic motion simulation. Most importantly, the computation involved in this method is not much more than mass-spring model but offers a significantly more accurate result.

(Kaldor et al. 2010) presented a method which particularly focused on the yarn-based cloth simulation. Yarn-based cloth simulation differs from sheet-based cloth simulation which is widely used in animation. For the Yarn-based cloth simulation, it always needs enough detail to describe the shape of yarn which leads to an expensive compute. In this paper, a method is proposed for approximating penalty-based contact forces in yarn-yarn collisions by computing the exact contact response at one time step. After that, they use a rotated linear force model to calculate forces in nearby deformation. Their method can speed up contact force by breaking it up into a group of disjoint regions and adaptively constructing local models for each region to approximate the true force response. This method only targeting on the yarn-based cloth simulation, it cannot handle other types of cloth simulation.

Physical method calculates the inter-force amongst each vertex. The number of vertex which construct the cloth directly influences the apparelling of the simulation. To produce a fine detail result the mesh resolution for cloth must stay high which leads to a very heavy calculation. Thus it has been mainly used in off-line high accurate simulation such as garment industry and off-line film production.

2.3.3 Hybrid Virtual Clothing

Hybrid method combines physical and geometrical and many other methods together to produce a fine result meanwhile minimize the calculation.

Rudomin (1990) presents a method uses geometrical approximation to reduce the computation time which required by physical techniques. In his method, firstly, the convex hull is constructed by wrapping the object in the cloth. Then the bottom of the convex hull is removed and a point is created along the line of the objects centre of mass. After that, the point is added in to the convex hull to rebuild it. In the next step, he developed an algorithm to map the upper hull onto the cloth object and removed all the points which lie outside the hull. Then the points which stay inside the hull are used to construct the mesh to give the shape of the draped cloth. After that, the deformable model presented by Terzopoulos et al. (1987) is used to refine the mesh of the cloth.

Kunii & Gotoda (1990) points out that when using the physical method to model the cloth, the textile is usually represented by many small elements, the polygons, the more detail required, the more computation time needed. Moreover, the differential equation which represents the cloth object is difficult to obtain the right parameters. This paper also indicates the limitation of the geometrical cloth modelling technique. Therefore, a hybrid method is proposed in this paper to overcome the drawback of both methods. Firstly, physical method has been used to simulate the sewing process which a piece of cloth can be encircled to the body, for example the sleeves which warped around the arms. The grid with springs connected in-between each others are used to represent the cloth object. During this process two kinds of energy are defined, metric energy and curvature energy. The shape of the cloth object is determined by the energy minima using the gradient descent method. In the next step, the wrinkles are refined by using the singularity theory. Then the

wrinkles continue to be simulated using the features which includes position of the feature points and the contours of the wrinkles captured from real cloth. Although his method simplified the physical method by integrate the geometrical modelling techniques, it still has many limitations such as the feature points which corresponding to the data captured from real cloth should remain constant. Moreover, the contours become sharper and the wrinkles become deeper after animation.

Tsopelas (1991) introduced a hybrid technique to generate folds on the cloth without collision in-between itself or between underlying bodies. The cloth is represented by the cylinder under the axial loads and the folds on the cloth are simulated by using thin-walled deformation theory. This process focuses on the area where the folds most likely occur, such as back of the knee, elbow or around the seams where the regions have large curvatures. Once the loads add onto the cylinder, diamond shape patterns occur. After the diamond shape patterns have been located, the algorithm traces several principle curves in each pattern and fit them with elastica. Then the deformation of each elastica is calculated. The non-uniform B-splines is used to interpolate between the points on the elastica to construct the deformed surface. When apply to animation, new principle curves are traced and elasticas are fitted for each step.

Hadap et al. (1999) introduced a geometric and texture based method to cloth wrinkle generation. Wrinkles are generated from the bump map on a coarse physical simulation of the cloth object and the bump map is created by user. Then the wrinkles are animated by modulating as per cloth deformation. Based on the theory of cloth having very little in-plane deformation and most of the deformation comes from buckling, using the length changing of the triangle edge to determine the depth of the bump map can simulate wrinkle efficiently. However, the direction of the wrinkle on the bump map has to significantly differ such as the wrinkle of the same location on the cloth in

two different bump is perpendicular to each other. Thus there can be a limited number of bump maps created and the variety of the wrinkle is consequently limited.

Cutler et al. (2005) introduced a kinematic method for generating wrinkles on cloth for CG characters by artist. In this method, the wrinkles are created by artists using the curve-based method on different poses. The wrinkle can be created by using any geometric sculpture software then stored into database. After that, based on the stretch distribution of the cloth on different poses, the wrinkles are sorted in to different category corresponds to the poses. During the animation, the cloth is simulated by coarse physical simulation to handle its global deformation. Then by matching the similar stretch distribution of current cloth to the reference pose, the corresponding wrinkle is mapped onto the animated cloth. Because the wrinkle generated by pure physical simulation is not what the artist wants. By using this method, all the wrinkles are sculpted by the artists, giving them more control ability. Moreover, by setting up the wrinkle database, wrinkles can be easily transferred in to a different character, which improve the production efficiency further. However, this method was mainly focused on how to create wrinkle on one layer tighter garment. The looser garment still need to be simulated by physical method and the interaction between the tighter cloth and the looser garment is very difficult to handle.

Popa et al. (2009) introduced a method for bringing the fine folds into captured cloth using data-driven dynamic wrinkling. In this method, the shape and the position of the wrinkle are captured from video footage based on the wrinkles distinguishing shape characteristics, and then the wrinkles are created by using stretch-minimizing deformation, which produces believable wrinkle shape. However in this method, the detail of the wrinkles captured is bounded from below by mesh resolution. The experiment presented in this paper has already used about 100K mesh for a single cloth during the wrin-

kle capture. But the results shown in this paper indicate that there is still a significant amount of detail lost during the capture.

Wang et al. (2010) proposed a technique which combines the synthesized fine wrinkle details with coarse physical simulation to produce high quality cloth wrinkles. In this paper, they point out in some scenarios that mostly focused on the appealing, the highly realistic wrinkle can be mimicked by matching a database of pre-extracted wrinkle configurations. The wrinkle database is constructed by pre-simulate its dynamic behavior using high resolution physical simulation. After that, the global dynamic behavior of the cloth can be generated by coarse physical simulation on low resolution mesh meanwhile the motion of the joint responsible for the certain area of skin deformation is also recorded. Although the pre-simulation is highly time-consuming, once it has been done, it can be used for a wide range of motions. In the next step, for each joint certain mesh template is used to create the individual wrinkle for certain motion. Then all the joint wrinkles are merged together into a fine mesh for the cloth simulation. Finally, the low-resolution mesh is simulated using physical method to calculate the collision, gravity, and many other interactions between the characters body and the environment.

Feng et al. (2010) introduced a hybrid method which can provide high-quality dynamic folds and wrinkle for cloth simulation while still maintaining real-time ability. This method captures the relationship between the two different resolutions of mesh by using data-driven model. Then this relationship is used to transforming the lower quality deformation of the lower resolution mesh through a mid-scale deformation transformation onto the higher resolution mesh with the fine details added on. In the animation pipeline which they developed, during each time step, it starts with the physical simulation for the 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 the mesh with the lowest resolution is mapped onto a mid-resolution object which using proxy bone to represents a local region on the high-resolution cloth object, and then the deformation of the proxy bones are mapped onto the high resolution to create the final fine-scale deformation. Collision detection is also a critical and time-consuming process during cloth simulation. In this paper, they proposed a method of using the proxy bones originally for the mid-scale cloth deformation to represents the high-resolution cloth object to interact with the model of character. The bounding volume is attached to every proxy bone which represents a small area on the cloth object. When the character moves, instead of calculating the collision between each mesh triangles of the cloth and body model, they can calculate the collision between each proxy bone. Although the accuracy of collision handing is slightly reduced, the computation time for the collision test are reduced significantly. Although the efficiency has been improved significantly, like other data-driven method, it requires a time consuming pre-simulation to gain the training data. Moreover this method cannot handle the self-collision of cloth.

Nowadays, virtual clothing has had to rely heavily on physically-simulated cloth in the production environment. The physical cloth simulation system can produce visually realistic results and have become widespread used in the production. But the computation cost of the physical method is still very high, especially with hundreds of characters in the same scene. The hybrid method combines with different kind of technique into one package to solve the problems in cloth simulation. Those techniques which has been combined with can overcome the drawbacks of pure physical simulation to provide a more effective and efficient solution for the virtual clothing. Especially in those areas which focus on the visual result more than the physical accuracy.

2.3.4 Summary

In this chapter, a brief history of cloth simulation in computer graphic has been reviewed. In general, there are three kinds of techniques to modelling and simulating the cloth.

The geometrical method is the fastest way to generate the cloth but it does not consider the physical properties of cloth. It represents the cloth by geometrical equations. For that reason, normally it is used as a modelling tool for cloth design in static situations. And it needs a considerable amount of direct user operation to create a cloth.

The physical method describes the cloth as a dynamic model which changes its shape by time or forces. The shape of cloth is determined by the forces or energies of vertexes. This method not only be able to model the cloth for static simulation but also has the ability to be applied in dynamic simulation. It has already being widely used in garment industry and animation. As the number of the elements which represents the cloth directly determines the detail of the simulation, therefore it is a heavy computation method to produce fine detail result.

The detail of the apparelling for the simulation done by physical method is directly determined by the quantity of basic element it uses such as polygon or particle. For this reason, the only way to increase the detail level of the simulation is to increase the resolution of the mesh or particles. This directly leads to the incensement of the computation. Especially with the high speed of development of film production, TV, games and online trading, more and more applications require real-time ability of cloth simulation or seeking for a new method that in order to speed up the current simulation procedure. Thus, the hybrid method is bringing into the cloth simulation area. Based on the geometrical method or physical method, combine with other computer techniques such as, data driven method, image based modeling, etc. the most time

consuming task which is fine detail wrinkle generation is usually handled to other high efficiency methods rather than physical calculation. Thus, the hybrid method has the ability to overcome many drawback of each method on its own and provides a high efficiency approach to the application. However, the hybrid method also has its own drawbacks, because it using other techniques such as data-driven to simulate the wrinkle instead of producing them by solving the physical equation. In order to cope with the various of wrinkles, large amount of pre-calculation of wrinkle is often required to build the training data for wrinkle generation.

Although many methods has been developed and several important research achievement have been reviewed in this chapter, dressing multiple character is still considered as a challenging task. Among the aforementioned methods, few are focus at modelling cloth from fit. This is caused by the limitation of computer hardware. Virtual clothing is a very time-consuming process, few years ago, computer can only handles few character together, therefore, the research on virtual cloth is mainly focus at reproduce high visual fidelity cloth for single character. Using the methods presented in this chapter to dressing several virtual character can be complete in an accepted time. However, followed by the high speed development of computer hardware, hundreds of character can be handled at the same time, using current virtual clothing method involves to much repetitive process. This thesis focus at geometrical cloth modelling for multiple characters. In order to cope with the increasing needs for dressing multiple character efficiently, the pattern based cloth making techniques is borrowed from tailoring to “tailor made” cloth for every character individually and automatically.

Chapter 3

Geodesic Algorithm for Measurements

In anthropometric data acquisition two type of measurements are associated with cloth making, length and circumference. The length measurements are usually obtained by calculating Euclidean Distance between two datum points which is the landmarks on the skin that defines the end points of a measurement. This procedure requires the subject remains standing or sitting posture, the correctness of the posture significantly affects the accuracy of the measurements. In computer animation, the “T-Pose” is the standard posture for modelling character because the body and limbs are stretched straight so the space between different body part are maximized. This provides many convenience for rigging and texturing. Unfortunately, the concept of “T-pose” are ambiguous because different studios have different definition of the “T-Pose”.

When measuring, the changes of the posture might need different measuring method to cope with, For example, in order to measure the length of the arm, the subject require extend their arm straight and the length is the distance from shoulder joint to the wrist joint. However, when measuring a subject in a bended arm posture, the length of arm is acquired by adding the

length of upper arm and lower arm. Another example is the acquisition of the height, for a character in a standing straight posture, the height can be obtained by calculating the Cartesian Distance from top of the head to the bottom of the heel. However, when character bends 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 of measurements are also differs.

In real world, when measuring the human body for cloth making, one end of a measuring tape is held to one of two datum points and tension is applied to the tape ruler so that it follows the profile of our body as closely as possible when it reach the second of the datum point. The path followed by the tape ruler can be considered as a geodesic between two datum points on the surface of the body that is being measured.

In computer graphic, if a character is rigged, that is, the character has skeleton to associated with, the length of the body part, can be measured by acquiring the length of the skeleton associated with that part of body. However, the skeleton system only applies to the character that is used for dynamic purpose such as animation. For static purposed character such as the virtual character for still image. The measurements can only be acquired based on the datum points on the skin. In order to develop a measuring method for both static and dynamic purposed character, only the datum on the skin is considered for measuring because polyhedral surface is the most common geometrical representation used in computer graphic, and datum points are a group of vertices that located on the polyhedral surface of character, therefore, extract measurement based on datum points can be used by any virtual character represented by polyhedral surface.

When measuring character according to the datum points on polyhedral surface, the Euclidean distance will change when surface is in different status. Figure.3.1 demonstrate different measurement caused by different pos-

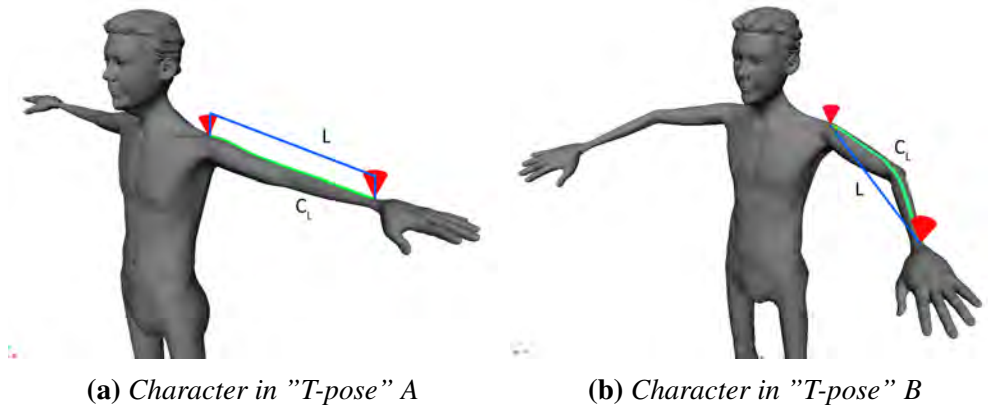


Figure 3.1: *Measurements in different posture*

ture. The red cone indicates the two datum points on the shoulder and wrist of the character. The green curve indicates the geodesic generated by the 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 and it is the Euclidean distance between two datum points, C_L denote the arc length of geodesic. The character is in two most common "T-pose".

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 subject. Therefore in thesis, the length measurement are acquired by measuring geodesic between two datum points on the body of subject in order to cope with different posture of the subject.

In order to calculate geodesic, a graph with positive edge weight is required because one of the fundamental requirement for geodesic computation is the connectivity between nodes in the graph. The discrete polyhedral surface is used as the most common shape representation in computer for a very long time and the topology of the discrete polyhedral surface suits this requirement very well. Therefore, in the past few decades, many algorithms has been developed for compute the geodesic on discrete polyhedral surface.

Nowadays, with the fast development of computer hardware, easy access of 3D shape acquisition device, cheap computer storage as well as increasingly fast networks, high-fidelity 3D scanned human body model has been widely accepted as simulation base model for cloth. However, The point cloud data that are usually outputs by the 3D shape acquirement device does not have the connectivity between sampling points. This feature makes geodesic computation much more difficult.

This chapter presents a novel discrete geodesic computation scheme. The approximate algorithm presented in this chapter is not only able to calculate geodesic with a bounded error, but also able to achieve linear time complexity. Because normally, the computation of geodesic is a time consuming process, by using algorithms presented in this chapter, not only it is able to compute accurate geodesic but also it can to extend to adapt to point cloth shape representation.

3.1 Introduction

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

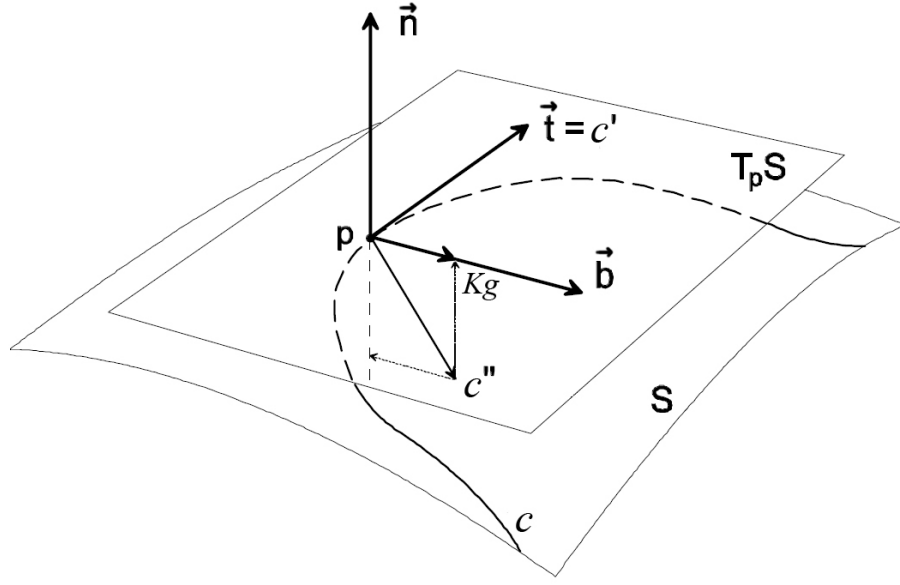


Figure 3.2: *Geodesic on a smooth surface.*

Figure.3.2 demonstrate the geodesic curvature on a point on a curve, where S is a two-dimensional smooth surface. C is a curve on S , T_pS is the tangent plane of the point $p \in S$, \vec{n} is the normal at p , \vec{t} is the tangent vector at p and C' denote first order derivative of C at p . \vec{b} is a vector perpendicular to \vec{n} and \vec{t} . C'' denotes the second order derivative of C at p in which is the geodesic curvature of C at p . κg is the projection of C'' on \vec{b} .

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

1. C is locally shortest curve.
2. C'' is parallel to the \vec{n} .
3. C has vanishing geodesic curvature $\kappa g = 0$.

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. Polthier & Schmies (2006) defines total vertex angle as,

Definition 2 Let S be a polyhedral surface and vertex $v \in S$. S is consisted by a set of faces denoted as $F = f_1, \dots, f_m$. Let p be a vertex contained by face f_i and θ_i is the interior angle of f_i at p . Then the total vertex angle $\theta(p)$ is given by,

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

All vetex of a polyhedral surface can be categorized based on the sign of the *vertexangleexcess* $2\pi - \theta(p)$. Figure.3.3 demonstrate three types of vertex on polyhedral surface according to Definition.2.

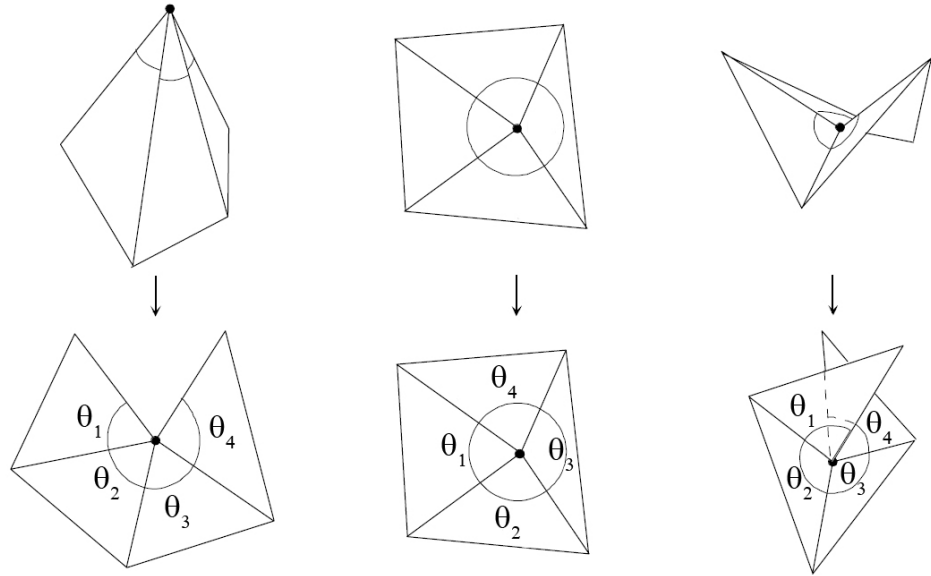


Figure 3.3: Spherical Vertex, where $2\pi - \sum\theta > 0$ (left). Euclidean Vertex, where $2\pi - \sum\theta = 0$ (middle). Hyperbolic Vertex, where $2\pi - \sum\theta < 0$ (right).

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

Definition 3 Let S be a smooth two-dimensional surface, A smooth 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 see Figure.3.4 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 in more than one time in a edge sequence(Mitchell et al. 1987a).”

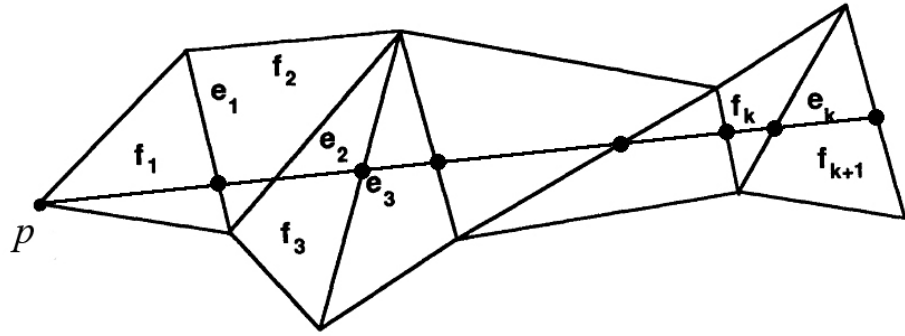


Figure 3.4: Geodesic on a polyhedral surface. Where $f_1 \dots f_k$ is the planar unfolding of p . $e_1 \dots e_k$ is the edge sequence E .

Figure.3.4 illustrate a geodesic starts at point P and goes through a sequence edge $e_1 \dots e_k$.

Geodesic computation is a common operation in computer aided design, machine learning, medical image analysis and computer animation. Throughout years of research, many geodesic algorithms have been developed for various applications, such as surface-based brain flattening (Bartschgi & Sapiro 2001; Wandell et al. 2000), mesh refining (Peyré & Cohen 2006), mesh segmentation (Katz & Tal 2003) and terrain navigation (Alexandrov et al. 2005). However, limited by the computational cost, the existing algorithms may not be able to cope with much larger models that have well over a million vertices, this is further proofed in the experiments presented in this chapter. In addition, most of the algorithms focus on the computation

of geodesic distances rather than geodesic paths which is done by performing “backtracing” process. The performance achieve was measured by not perform the “backtracing” process. If not so, the total running time would be dramatically increased.

Followed by the fast development of shape acquisition devices, such as MRI, laser scanner. The data provides by shape acquisition device comes in form of unorganized point range data (point clouds). Geodesics computation on mash-based data has been well studied for decades, many efficient algorithm has been developed (Dijkstra 1959; Hershberger & Suri 1999; Kanai & Suzuki 2000; Mitchell et al. 1987b; Peyré & Cohen 2006). However, geodesics computation on point clouds is less studied (Daniel Boissonnat & Cazals 2001; Thielhelm et al. 2012; Ruggeri et al. 2006). The current methods of computing geodesics on point clouds involves an intermediate step in which a surface is fit to a subset of points. The accuracy of the geodesic is determined by the quality of the fitted surface and the time cost for the surface generation affects the performance.

In the following sections, a novel geodesic computation scheme that is based on the geodesic curvature flow is introduced in detail. Three algorithms has been developed to solve the different problem,

1. Accurate geodesic paths on polyhedral surface
2. Linear time complexity approximate geodesics on polyhedral surface
3. Approximate geodesics on point clouds

Of these three algorithms, the approximate geodesics algorithm on manifolds has a linear-time complexity while maintaining an error bound. The main advantages of the scheme are list below:

1. High efficient: the approximate geodesics algorithm on manifolds has an $O(n)$ time complexity, therefore, it is suitable for large models and complete the computation within an acceptable time. Furthermore, the

topology quality of the mesh does not effect accuracy of the algorithm.

2. High Accuracy : the accuracy of the approximation algorithm for our geodesic scheme has a relative error threshold. The error is only related to the "window" size determined by the user. The topology of the mesh has no affect on the accuracy of the algorithm.
3. Adaptability: well-known algorithm such as Dijkstra (1959), MMP (Mitchell et al. 1987b; Surazhsky et al. 2005) Chan and Han (Chen & Han 1990; Xin & Wang 2009) and Fast Marching Method (Kimmel & Sethian 1998) requires manifold meshes. Furthermore, these algorithm cannot handle point cloud data produced by 3D scanner or sampled implicit-surface. the geodesic scheme introduced in this chapter does not strictly rely on the connectivity of the vertices of the mesh. Therefore it can be easily adapted to handle the point cloud data.

3.2 Previous Work

Dijkstra (1959) was the first to address the problem of "single source all destination shortest path" problem on a directed non-negative weighted graph. Early work mainly focused on convex polyhedrons. Sharir (& Schorr(1984) were the first to extend the Dijkstra's algorithm into three dimensions, with a time complexity $O(n^3 \log n)$. 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 the hyperbolic vertices of a polyhedron, see Fig.3.3. Based on the geodesic definition of Mitchell et al. (1987a), a geodesic path on the planner forms a straight line. However, when flatten the face surround to a hyperbolic vertex, one edge will appear twice in two different edge se-

quence. In order to deal with this problem, Mitchell et al. (1987b) (MMP algorithm) extends Dijkstras algorithm and introduce the technique of "continuous Dijkstra" that propagates the distance information from the source outward in a Dijkstra-like manner as follows. Each edge is partitioned into several sections called "windows", 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, they extend the original MMP algorithm (Mitchell et al. 1987b) to a bounded error approximation algorithm note as MMP approximate algorithm. The time complexity of the MMP approximate algorithm has been improved to $O(n(\log n))$ (Surazhsky et al. 2005).

The other 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 $O(n \log n)$. 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) point out that the approximation error of FMM is unbounded.

Chen & Han (1990) (CH algorithm) provide an exact solution to the geodesic problem based on a key observation "one angle one split" principle with time complexity of $O(n^2)$. This method consists of a two part calculation. In the first part, the shortest path from a given source to each vertex on the mesh is computed and a set of windows that contain the information about the shortest path from the given source point to points on each edges. In the second step, the windows are used to compute the decomposition of the surface and the shortest path to any destination point on the polyhedral surface can be calculated. However special case need to be dealt with in the circumstance of geodesic pass through a hyperbolic vertex. Since the planer unfolding can be self-overlapped Fig.3.3, pseudo-sources are used for win-

dow propagation.

Xin & Wang (2009) (Improved CH algorithm) discovers that in CH algorithm, many windows are unnecessary for the propagation. Therefore, their method improved the efficiency of the algorithm by introducing less windows during the propagation. Although the asymptotic time complexity is $O(n^2(\log)n)$, according to the numerical experiments, the Improved CH algorithm show great advantage over CH algorithm and MMP algorithm in terms of computational time costs.

The aforementioned algorithm were developed for relatively low resolution models, where delays in running time are not always apparent. Therefore these algorithm are unable to cope with much larger models with well over a million vertices.

For point cloud data, since there is no connectivity among the sampling point and the connectivity is the fundamental requirement for the propagation based method such as MMP (Mitchell et al. 1987b; Surazhsky et al. 2005), CH/ICH (Chen & Han 1990; Xin & Wang 2009) and FMM (Kimmel & Sethian 1998), The approximate graph need to be construct based on the points to form the connectivity for point cloud data. For instance, Mémoli & Sapiro (2001) extends the FMM to point cloud data, where the resulting geodesic is an approximation over a coarse grid on an offset from the point cloud surface. In order to constrain the discrete curves on the point cloud surface, Moving Least Squares was employed by Hofer & Pottmann (2004).

Pottmann et al. (2010) introduced a method of compute geodesics by energy-minimizing piecewise curve constrained on MLS surface. Although their method is efficient on high dimensional data, the accuracy of the geodesic path is depends on the noise of the constructed MLS surface.

Ruggeri et al. (2006) computes an approximate geodesic on point set surface, using an energy-minimization function in order to define piecewise

linear approximations of the geodesics. The initial curve, created by Dijkstra's algorithm (Dijkstra 1959) is refined by minimizing an energy function. However, the computational cost is unacceptably high for large numbers of geodesic computation requirement.

3.3 Geodesic Curvature Flow

In the following section, a geodesic curvature flow based scheme for computing geodesics over mesh and point clouds is discussed in detail.

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 $C(t)$ be a curve in Euclidean space that represents the position vector of the particle as a function of time, The FrenetSerret formulas apply to non-degenerate curves which have non-zero curvature.

Let $s(t)$ represent the arc length which the particle has moved along the $C(t)$ in time t . The curve $C(t)$ then can be parametrized by it arc length s and a point on curve $C(t)$ that has arc length s can be denotes by $C(s)$. Therefore at point $C(s)$, the FrenetSerret frame is defined by three vectors,

The tangent unit vector \vec{T} is defined as:

$$\vec{T} = \frac{dr}{ds} \quad (3.1)$$

The normal unit vector \vec{N} is defined as:

$$\vec{N} = \frac{\frac{d\vec{T}}{ds}}{\left\| \frac{d\vec{T}}{ds} \right\|} \quad (3.2)$$

The binormal unit vector \vec{B} is defined as the cross product of \vec{T} and \vec{N} :

$$\vec{B} = \vec{T} \times \vec{N} \quad (3.3)$$

Figure.3.5 demonstrate Frenet-Serret frame on a curve, note that \vec{T} , \vec{N} and \vec{B} are perpendicular to each other, the plane defined by \vec{T} and \vec{N} is the osculating plane at point $C(s)$, therefore, the matrix form of FrenetSerret formulas is(Kreyszig 1991),

$$\begin{bmatrix} T' \\ N' \\ B' \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix} \begin{bmatrix} \vec{T} \\ \vec{N} \\ \vec{B} \end{bmatrix} \quad (3.4)$$

where \vec{T} is the unit tangent vector, \vec{N} is the unit normal vector, \vec{B} is the unit binormal vector, τ is the torsion, κ is the curvature, T' denotes $\frac{d\vec{T}}{ds}$, N' denotes $\frac{d\vec{N}}{ds}$ and B' denotes $\frac{d\vec{B}}{ds}$. s is the arc length of the curve. Equation.3.4 can be written into following form,

$$\begin{cases} \frac{d\vec{T}}{ds} = \kappa \vec{N} \\ \frac{d\vec{N}}{ds} = -\kappa \vec{T} + \tau \vec{B} \\ \frac{d\vec{B}}{ds} = -\tau \vec{N} \end{cases} \quad (3.5)$$

Based on Definition.1, \vec{n} is the standard unit normal to surface S , $\vec{N} = \vec{n} \times \vec{t}$, see Figure.3.2. Therefore, \vec{N} , \vec{n} and \vec{t} are mutually perpendicular unit vectors. Because geodesic is a curve on smooth surface S , the Frenet-Serret

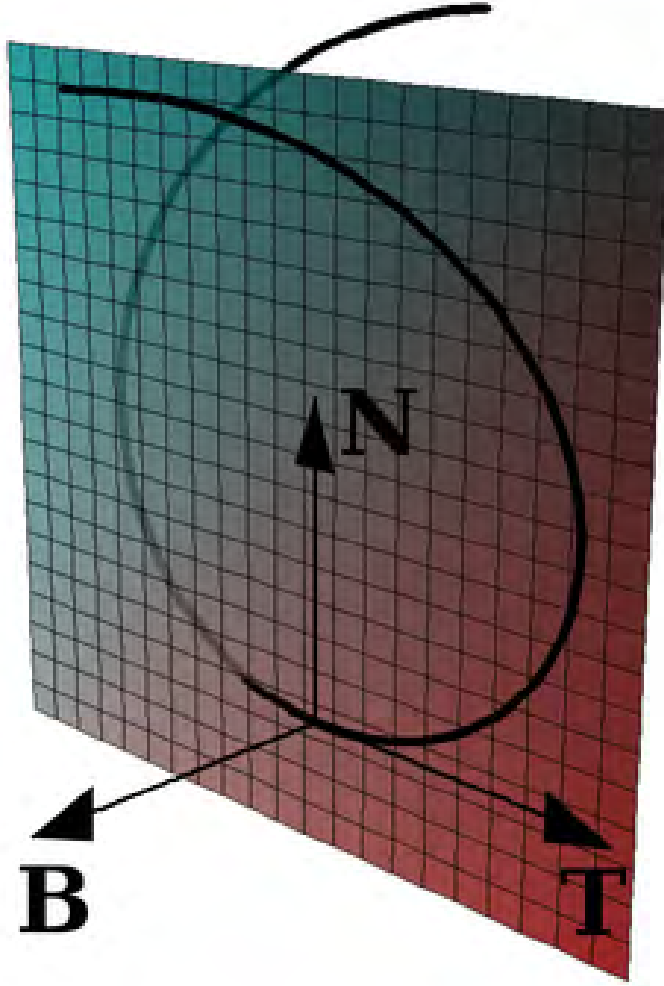


Figure 3.5: *FrenetSerret frame on a curve*

formula can be written as,

$$\begin{cases} \frac{d\vec{T}}{ds} = \kappa\vec{n} + \kappa_g\vec{N} \\ \frac{d\vec{N}}{ds} = -\kappa_g\vec{T} + \tau_g\vec{n} \\ \frac{d\vec{n}}{ds} = -\tau\vec{N} - \kappa_g\vec{T} \end{cases} \quad (3.6)$$

where $\kappa = \left\langle \frac{\delta^2 C}{\delta s^2}, n \right\rangle$ is the normal curvature at $C(s)$, $\kappa_g = \left\langle \frac{\delta^2 C}{\delta s^2}, \vec{N} \right\rangle$ 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 $\frac{\delta^2 C}{\delta s^2}$. Let C''_s denote the second order derivative to arc length s . therefore, $\frac{\delta^2 C}{\delta s^2} = C_{ss}$.

Now, the 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 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 Cs 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 C'_s , $\vec{N} \times C'_s$. C'_s is the first order derivative of $C(s)$, Thus, for any point $x(s)$ on $C(s)$, the velocity under this flow is given by

$$F(\kappa_g) = \left\langle x', (\vec{N} \times C'_s) \right\rangle \quad (3.7)$$

the expression of geodesic curvature is,

$$\kappa_g = \left\langle (\vec{N} \times C'_s), C''_s \right\rangle \quad (3.8)$$

therefore, the geodesic curvature flow is given by,

$$C(s, t) = k_g \vec{N} = C''_s - \left\langle C''_s, \vec{N} \right\rangle \vec{N} \quad (3.9)$$

where $\left\langle C''_s, \vec{N} \right\rangle \vec{N}$ is the projection of C''_s on \vec{N} , note that t is a time variable of the flow in which when $t = 0$ states the initial statue of the flow Cs, 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.3.9 can eliminate the geodesic curvature pointwise on $C(s, t)$ by moving $C(s)$ in the gradient direction.

Polthier & Schmies (2006) proved that, on polyhedron 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 to the discrete surface geodesic. In computer graphic, a polyhedral surface can be considered as the inscribed polygon approximation of a smooth surface. Therefore the a curve on the smooth surface can also be approximated by a piecewise curve consisting 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 the large model, that has millions of vertices, to generate parametric approximation surface for the polyhedron and the piecewise curve require large amount of computation resources. To avoid this excess calculation, the second order derivative of $C(s)$ is directly defined on the piecewise curve by a triangle $\triangle p_{i-1}, p_i, p_{i+1}$ as shown in Figure.3.6.

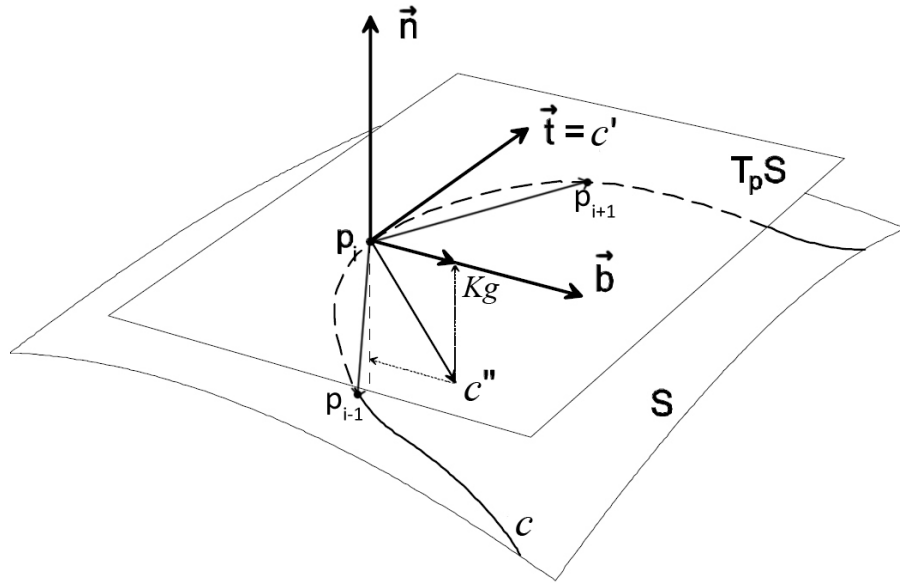


Figure 3.6: *Geodesic curvature on piecewise curve*

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 point on C and p_i is its arc length s to curve C . Thus $\widetilde{p_{i-1}p_i p_{i+1}}$ denotes the a section of the piecewise approximation of smooth curve C . In this case, at

p_i , the second order derivative of Cs at p_i can be estimated by Equation.3.10,

$$C_s'' \approx \overrightarrow{p_i p_{i+1}} + \overrightarrow{p_i p_{i-1}} \quad (3.10)$$

In order to define natural coordinate system on a vertex of a polyhedral surface, the tangent plane of S at p_i need to be confirmed. Assume the polyhedral surface is a desecrate approximation of smooth surface S , given a point s_i on $C(s)$, within a fixed radius to p_i , there exist vertices v_m, \dots, v_n that on the polyhedron. Therefore, the fitting plane of v_m, \dots, v_n defines the tangent plane of the polyhedron at vertex p_i . According to Definition.1, it can be said that, if the projection of C_s'' onto this tangent plane at p_i vanishes, the geodesic curvature will also vanish at p_i accordingly. Furthermore, if the geodesic curvature at all the point on the piecewise approximation curve of $C(s)$ vanishes, the $C(s, t)$ is converged, 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 $C_s'' = KP$.

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

and

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

where K is a $3m \times 3m$ coefficient matrix and m denotes the number of

the sample points on piecewise curve. Now, geodesic curvature flow $C(s, t)$ can be rewritten as,

$$C(s, t) = k_g \vec{N} = KP - \vec{n} \vec{n}^T KP \quad (3.11)$$

where \vec{n} is a block diagonal matrix consisting of the normal vectors of the tangent plane at all the point in P ,

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

Let P_t denotes all the sample point on geodesic curvature flow $C(s, t)$ 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 the geodesic curvature vector. Therefore, the updated curve denoted as P_{t+1} can be expressed as,

$$P_{t+1} = P_t + \mu(KP_t - \vec{n} \vec{n}^T KP_t) \quad (3.12)$$

3.4 Geodesic on Mesh

3.4.1 Tangent Space Constraint

In order to calculate a geodesic on polyhedral surface using Equation.3.12, the normal vector need to be estimated at every iteration for all the points on polyhedral surface corresponding to every sample point till Equation.3.12 converged.

According to the experiment presented in this chapter, during the iteration of Equation.3.12, the updated sample points tends to deviate from the

surfaces. 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 follows.

Firstly, P denotes all the sample point on curve, all the normal vector of the face that within one-ring neighbour of $p_i \in P$ is stored in to a matrix N ,

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

Applying PCA(Jolliffe 2002) to N , the tangent space are defined by $vecn_i$ which denotes the eigenvector associated with the smallest eigenvalue. After that, for each $p_i \in P$, let $p_{i,t}$ denotes the current statue 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,

$$p_{i,t+1} = p_{i,t} + \mu \vec{n}_i \vec{n}_i^T (c - p_{i,t}) \quad (3.13)$$

where, μ is an iterative step size and c denotes the centre point of the one-ring neighbour points of p_i . In essential, the tangent space constraint of Equation.3.13 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.3.13 with Equation.3.12,

$$P_{t+1} = P_t + \mu(\vec{n}_i \vec{n}_i^T c - \vec{n}_i \vec{n}_i^T P_t + KP_t - \vec{n}_i \vec{n}_i^T KP_t) \quad (3.14)$$

where, c is a column vector consisting of the coordinates of all means of the one-ring neighbour of the closest vertex to the corresponding sample point on curve.

3.4.2 Implicit Euler scheme

In order to solve Equation.3.14, 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 a very time-consuming process.

The implicit Euler method avoid this cost(Butcher 2008), and although its error is the same as explicit Euler method, its distinct advantage is to remove the iterative step size limitation. That is the implicit Euler method will be stable with any step size(Hairer 2010).

The basic idea of the implicit Euler method is to use implicit differencing, i.e, the right-hand side of Equation.3.14 is evaluated at the new location of $t+1$. This is called as the backward Euler scheme(Enns & McGuire 2000). Applying the backward scheme to Equation.3.14 yields,

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

where I is an identity matrix, μ denotes the step size. The resulting geodesic curvature flow of Equation.3.15 is remain stable even at $\mu \rightarrow \infty$ (Hundsdorfer & Verwer 2003).

Equation.3.15, is linear equation of form of $AX = B$ where $A = I + \mu(\vec{n}^* - K + \vec{n}^* K)$ and $B = p_t + \mu \vec{n}^* c$ and can be solved efficiently by LU factorization with full pivoting presented by Trefethen & Bau (1997).

3.4.3 Least Squares scheme

Although the implicit Euler scheme provides an efficient method for solving Equation.3.15. The experiment in this thesis shows that the geodesics result from Equation.3.15 still requires one or two additional iterations in order to achieve convergence.

Experiments show that during the convergence of Equation.3.12 , all the sample points on the geodesic curve are updated with in the tangent space except two endpoints p_1 and p_m of curve $C(t)$. Therefore, Equation.3.12 can be written in a least squares form in which two endpoints p_1 and p_m are the constrains(Jiang 1998),

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

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

where \vec{n} denotes the normal of the tangent space. Essentially, the sys-

tem Equation.3.16 is the 1st order approximation to the geodesic curvature flow. A linear system can be constructed from Equation.3.16,

$$\begin{bmatrix}
 A_{0,0} & A_{0,1} & \cdots & & A_{0,j} \\
 A_{1,0} & A_{1,1} & & & \\
 \vdots & & \ddots & & \vdots \\
 A_{i,0} & & \cdots & A_{0,0} & A_{i,j} \\
 1 & 0 & 0 & 0 & \cdots & 0 \\
 0 & 1 & 0 & 0 & \cdots & 0 \\
 0 & 0 & 1 & 0 & \cdots & 0 \\
 0 & \cdots & & 1 & 0 & 0 \\
 0 & \cdots & & 0 & 1 & 0 \\
 0 & \cdots & & 0 & 0 & 1 \\
 N_{0,0}^* & N_{0,1}^* & \cdots & & N_{0,j}^* \\
 N_{1,0}^* & N_{1,1}^* & & & \\
 \vdots & & \ddots & & \vdots \\
 N_{i,0}^* & & \cdots & & N_{i,j}^*
 \end{bmatrix} p = \begin{bmatrix}
 0 \\
 \vdots \\
 p_{1_x} \\
 p_{1_y} \\
 p_{1_z} \\
 p_{m_x} \\
 p_{m_y} \\
 p_{m_z} \\
 B_{0,0}^* \\
 \vdots \\
 B_{i,0}^*
 \end{bmatrix} \quad (3.17)$$

where, $A = K - \vec{n}\vec{n}^T K$, $N^* = \vec{n}\vec{n}^T$, $B^* = \vec{n}\vec{n}^T \bar{P}$, m denotes the number of sample point on curve. \bar{P} is a column vector consisting of all the projections \bar{p}_i from sample point p_i onto the tangent plane defined by \vec{n}_i . The coefficient matrix on the left hand side of the system has size of $(2 * 3m + 6) \times 3m$ and the one on the right hand side of the system has the size of $(2 * 3m + 6) \times 1$.

Note that Equation.3.15 cannot compute 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.3.15, which is then solved to obtain the final sample locations. Ac-

cording to this scheme, the final location of a sample point is still constrained within the tangent space defined for its initial location. Note that the solution of Equation.3.15 cannot guarantee 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 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 on smooth surface to its approximate mesh.

3.4.4 Geodesic Path Projection

Although the sample point updated by Equation.3.15 only moves within the tangent space, the actual mesh which the geodesics are calculated on is not flat. Therefore, the updated points tend to derive from mesh. This is showed in Figure.3.7,

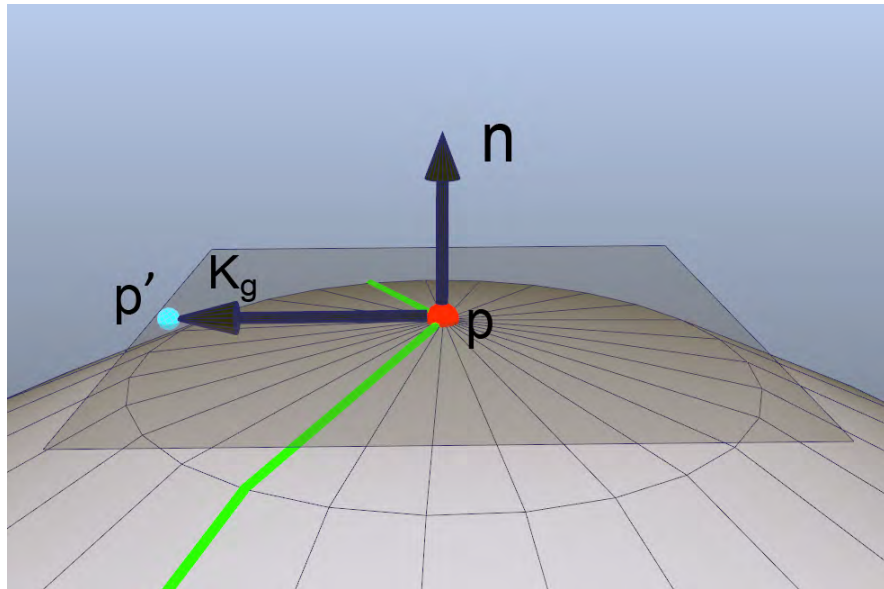


Figure 3.7: Derived point caused by updating point. green piecewise curve is the curve before updated by Eq.3.12. gray plane is the tangent plane at point p (red point). the point p' (light blue point) is the updated point.)

In order to calculate geodesics on the polyhedral surface, the offset geodesic curve need to be projected on to the polyhedral surface. Polthier & Schmies (2006) defines geodesic that a curve is a geodesic when at any point, geodesic curvature vanishes. On polyhedral surface, if a curve fits this definition and does not pass any spherical vertex on the polyhedral surface, it is both straightest and locally shortest geodesic. This is shown in Figure.3.8.

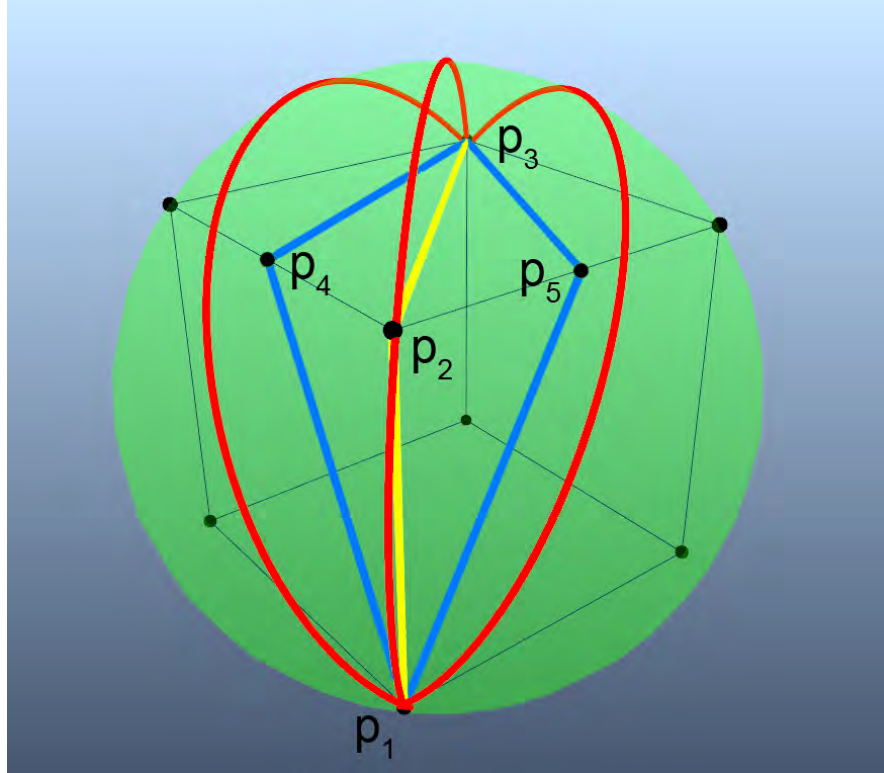


Figure 3.8: *Geodesic on smooth surface and its projection on the its inscribed polygon mesh. The green sphere contains its inscribed cubic. The red curves is the geodesic on sphere (great circle arc). p_1, p_2 and p_3 are points on sphere. p_4 and p_5 are the centre point on its edge.*

In Figure.3.8 both the yellow line $\widetilde{p_1 p_2 p_3}$ and blue lines $\widetilde{p_1 p_4 p_3}$, $\widetilde{p_1 p_5 p_3}$ are the projection of their corresponding red geodesic path on sphere. The path $\widetilde{p_1 p_2 p_3}$ that goes through the spherical vertex p_2 is the straightest but not shortest geodesic path on the cubic. However, the blue lines that pass p_4 or p_5 is both the straightest and shortest path on cubic by the definition presented in (Polthier & Schmies 2006). Note that $\widetilde{p_1 p_4 p_3}$ and $\widetilde{p_1 p_5 p_3}$ are the

projections of the great circle of the sphere on the direction of the curvature of their corresponding great circle.

In order to project the floating geodesic onto mesh, firstly, the projection direction need to be defined. According to the definition of geodesic curvature, geodesic curvature k_g of a curve c at point p is the vector projection of the curvature vector c'' of c at p onto the tangent plane $T_p S$ at point P on surface S as showed in Figure.3.2. Therefore, the curvature k at point p can be written as:

$$k = k_g + k_n$$

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

$$k_g = k \cos \theta$$

where θ is the angle between the osculating plane of c at point p and the tangent plane $T_p S$. Therefore, is we move p along the direction of \vec{c}'' , the osculating plane of c at point p will remain same so as the θ . Hence the geodesic curvature k_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 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 the one-ring face neighbour 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.

The source point p_0 is firstly selected as the starting point that always lies on a vertex of the mesh. The successive 2 sample points p_1, p_2 of a floating

geodesic form a plane cutting the mesh. This plane, $\Delta p_0 p_1 p_2$, is viewed as an approximation of the osculating plane Figure.3.9.

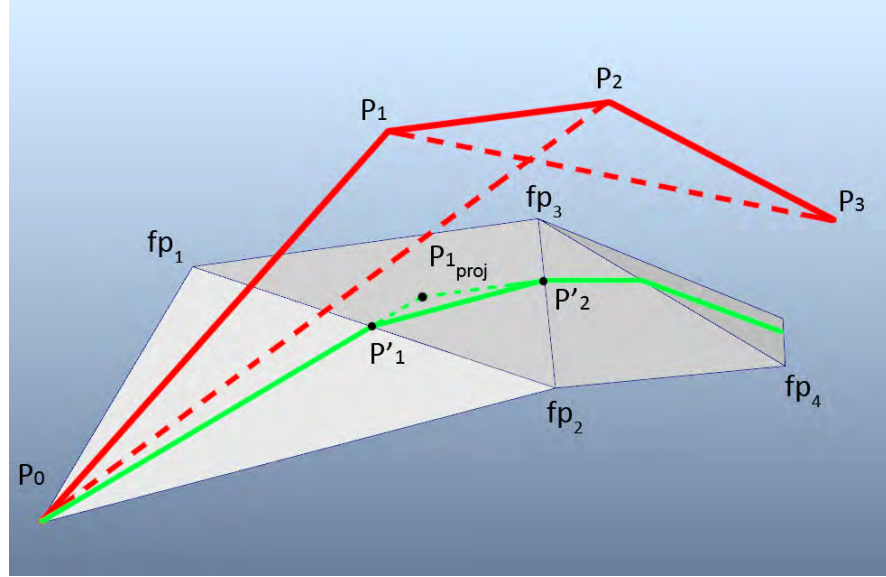


Figure 3.9: Projection of a geodesic path.

In Figure.3.9, the red line indicate the piecewise approximation of geodesic. p_0, p_1, p_2, p_3 is the sample point on the geodesic path $\widetilde{p_0 p_3}$. $\Delta p_0 p_1 p_2$ 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 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 $\Delta p_0 p_1 p_2$ and \vec{N}_2 denotes the normal vector of face $\Delta p_0 fp_1 fp_2$. Firstly, starts from the source point p_0 , the rectifying plane of projected path is approximated by $\Delta p_0 fp_1 fp_2$. As a result, the tangent vector of the projected geodesic at p_0 can be calculated by Equation.3.18.

$$\vec{v} = \vec{N}_1 \times \vec{N}_2 \quad (3.18)$$

where \vec{v} is equivalent to $\vec{p_0 p'_1}$. Follow this direction, the \vec{v} intersects

with an edge within the one-ring face neighbour of p_0 . p'_1 denotes this intersection on the edge $\overline{fp_1fp_2}$. Then, p'_1 is selected as the next starting point. A important property of manifold mesh is that, only one or two faces is incident to a edge. Therefore, the next projected face can be easily determined as the adjacent face $\Delta fp_1fp_2fp_3$ to the 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 intersected point on face $\Delta fp_1fp_2fp_3$ can be determined by performing Equation.3.18. If the projection of p_1 falls into the face $\Delta fp_1fp_2fp_3$ as shown in the Figure.3.9, \vec{N}_1 is updated by the normal vector of approximated osculating plane $\Delta p_1p_2p_3$ at point p_2 . Starting from the projection p_{1proj} of p_1 , perform Equation.3.18, the next intersection of \vec{v} and $\overline{fp_2fp_3}$ can be determined as p'_2 shown in the Figure.3.9. The projection method is summarized in Algorithm.1

3.4.5 "Continuous Dijkstra" Propagation

As described in Equation.3.12, this algorithm needs an initial path to calculate the updated geodesic path. "continuous Dijkstra" strategy presented by Dijkstra (1959) is used to generate this initial path. When performing "continuous Dijkstra" strategy, all the points in propagation boundary are kept sorted by the distance back to the source point in incremental order in a priority queue. At each step, starts from the first element in the priority queue, the boundary is propagated outward. New point are inserted into the queue at the location where the order of the queue remains.

However, keeping the priority queue in order involves "comparison sorting algorithm" that involves a searching operation cost $\log(n)$ 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

Algorithm 1 Geodesic Projections on Mesh

```
1: procedure PROJECT GEODESIC PATH ON MESH(A mesh  $S$ , and a float-  
   ing geodesic path  $\widetilde{p_0 p_n}$ )  
2:   for  $i = 1, i < n, i++$  do  
3:     Select the successive vertices of  $p_i$  and build osculating plane  
      $\triangle p_{i-1} p_i p_{i+1}$   
4:      $\vec{N}_1 \leftarrow$  normal vector of  $\triangle p_{i-1} p_i p_{i+1}$   
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:         $\vec{v}_{tan} \leftarrow p_s + \vec{N}_1 \times \vec{N}_2$   
11:        Compute the intersection point  $p'$  that an edge of current  
        face  $F'$  intersects with  $\vec{v}_{tan}$   
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
```

on large model. Because Equation.3.12 only requires an initial path that approximates to the shortest path only goes through vertices and edges. The vertices in the propagation boundary does not need to be kept in order. This algorithm is presented in more 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 connect p_s and the points in W is the geodesic path between them. This is due to with in a triangle, the shortest between two point is the edge that connects these two points. After that, the length of every edge is stored along with the path information to the vertices in W . Since this algorithm does not employ priority queue. To ensure the isometric propagation. A propagation radius limit r_{max} is introduced to constrain the propagation at every propagation step. At this moment, r_{max} is updated as the length of the longest edge that connected from p_s to W .

After the first W is formed by the one-ring neighbour of p_s , the first vertex $p_i \in W$ is selected and its one-ring neighbour excludes the vertices that has 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 to form its initial path at pn_i . This process is illustrate in Figure.3.10.

Where, pn_1 and pn_2 is 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 1-ring neighbour of pn_i is selected. In Figure.3.10, 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{v}_i and the edge that connects pn_1 and current selected visited vertex is calculated. \vec{v}_i denotes the tangent vector at current visited vertex on its geodesic path. The parent node of p_i is select with the visited vertex with the largest θ .

In Figure.3.10, the included angle θ_i between $p_i\vec{pn}_1$ and \vec{v}_1 is greater than the included angle θ_j between $p_j\vec{pn}_1$ and \vec{v}_2 . Therefore, it can be observed that, compared to the two curves of edge $\overline{p_i pn_1}$ plus geodesic path

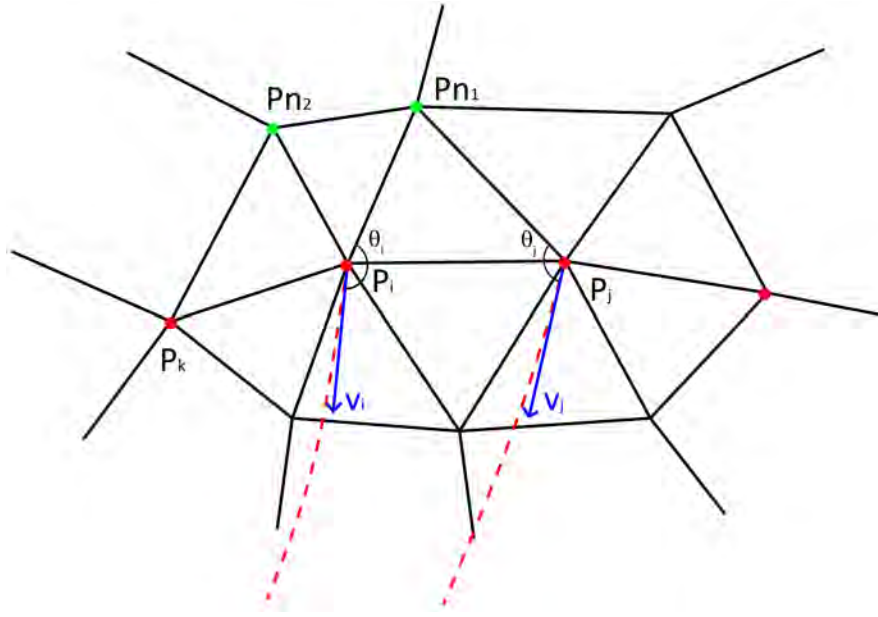


Figure 3.10: Calculating geodesic path for a unvisited vertex. red point p_i, p_j and p_k are visited vertex in wavefront W . red dash line indicates geodesic path from source to p_i and p_j . vector \vec{v}_i and \vec{v}_j is the tangent vector at p_i and p_j respectively. p_{n1} and p_{n2} are one-ring neighbour of p_i that has not been visited

at p_i and edge $\overline{p_j p_{n1}}$ plus geodesic path at p_j , the curve of edge $\overline{p_i p_{n1}}$ plus geodesic path at p_i are straighter than the curve of edge $\overline{p_i p_{n1}}$ plus geodesic path at p_i . Thus, vertex p_{n1} is pushed into the end of the geodesic path at vertex p_i as the initial path for p_{n1} . For every new point that pushed to the end of its parent point's geodesic path, the geodesic path information inherits from the parent vertex to its child. As a result, the algorithm presented here does not require “backtracing” that are necessary for MMP, CH and FMM algorithm to retrieve the geodesic path.

The aforementioned method propagates geodesic information from the interior rings to external ones. However, without the help of priority queue, there is no guarantee to propagate geodesic information from the vertex with smaller geodesic distances to the source to vertex with larger distances. Therefore, isotropy need to be maintain during the propagation in order to ensure the correctness of the causality of the algorithm that presented in this chapter.

Hence, a propagation radius limit r_{max} is employed here. 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 . The Equation.3.12 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 put into an array W_{next} as the “next wavefront” for the next propagation when both W and W' are empty. For every unvisited vertex that connected 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} , our 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. 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 geodesic paths that are calculated 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.3.12, the updated points on geodesic path always move away from the vertices on initial path. Therefore, in the process of Algorithm.2, in order to calculate the projections of sample point p on the floating geodesic path to mesh, exam every single faces on mesh for the projection is a very time consuming process. In order to maintain the efficiency of our algorithm, it is important to keep track on the corresponding vertices on the mesh where p are updated from. For every unvisited vertex, always 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.

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:        $\vec{v}_2 \leftarrow getTangent(v)$  ▷ Return the tangent vector of
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  $pc_j \in oneRingNeighbourOf(p_i)$  do
35:         if  $pc_j \in W$  OR  $pc_j.geoDist \neq \infty$  then
36:           continue
37:         else
38:            $parentNode \leftarrow GETPARENTNODE(W, pc_j)$ 
39:            $d_{est} \leftarrow distanceBetween(parentNode, pc_j) +$ 
               $parentNode.geoDist$ 
40:           if  $d_{est} < r_{max}$  then
41:             Perform Equation.3.12 on initialPath re-
              sults geodesic path from source to  $pc_j$ 
42:             Perform Algorithm.1 to project floating path
               $pc_j.geoPath$  onto  $S$ 
43:             Update the neighbourhoods for the samples
              point on  $pc_j.geoPath$  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
```

However, with the “wavefront” propagates, this assumption does not stands. Thus one-ring neighbourhood of every sample point need to be redetermined when a new point is added into the current geodesic. This is achieved by following procedure. Firstly, start from the 1-ring neighbours 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. This updating process is able to ensure the projection of updated sample point p on floating geodesic path always falls into the face that adjacent to the point that corresponds to p in its initial path.

Moreover, 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 consisting of the one-ring neighbours of these 3 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 can be easily determined. As a result, to calculate the intersection points of the cutting plane with all the related edges costs a constant time.

A challenging issue of using Algorithm.2 to calculate geodesic is to solve the large sparse linear system of Equation.3.17. Moreover, followed by the propagation of “wavefront” the number of the point in the initial path that need to be updated by Equation.3.12 is increasing. Therefore, the size of the matrix in Equation.3.17 is also increasing. This step is the most time consuming procedure in the algorithm presented here.

In order to improve the efficiency of this algorithm, an approximate algorithm is derived from Algorithm.2. In this algorithm, a small-size window is introduced to Equation.3.12 to avoid solving large sparse matrices. When an initial estimation of a geodesic path is obtained, as shown in Figure.3.10, a fixed-size window that covers the last w sample points is employed. Then Equation.3.12 is applied to w to calculate the local geodesic patch. Let $\tilde{p}_0 p'_{n-1}$

denotes the geodesic path from source to the parent of p' parent. Thus, only the section $\tilde{p}'_{n-w}p'_n$ is updated to form the approximated geodesic path. This algorithm is summarized in Algorithm.3.

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( $W, v_{unvisited}$ )
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

```

In Algorithm.3, because the window size w is a constant number, therefore, the matrix size in the linear system of Equation.3.12 does not change throughout the propagation of “wavefront”, thus the time that solving the

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  $pc_i \in oneRingNeighbourOf(p_i)$  do
35:         if  $pc_j \in W$  OR  $pc_j.geoDist \neq \infty$  then
36:           continue
37:         else
38:            $parentNode \leftarrow GETPARENTNODE(W, pc_j)$ 
39:            $d_{est} \leftarrow distanceBetween(parentNode, pc_j) +$ 
               $parentNode.geoDist$ 
40:           if  $d_{est} < r_{max}$  then
41:             Perform Equation.3.12 on the  $\widetilde{p_{n-w}p_n}$  part of
               $initialPath$  results geodesic path from  $p_{n-w}$ 
              to  $pc_j$ 
42:             Combine  $\tilde{p}_0p_{n-w-1}$  with  $\tilde{p}_{n-w}pc_j$  to form the
              approximate geodesic from source to  $pc_j$ 
43:             Perform Algorithm.1 to project floating path
               $pc_j.geoPath$  onto  $S$ 
44:             Update the neighbourhoods for the samples
              point on  $pc_j.geoPath$  from  $n - w^{th}$  to  $n^{th}$ 
              point separately
45:              $W' \leftarrow pc_j$ 
46:           else
47:             Update  $d_{maxEst}$  if  $d_{est} > d_{maxEst}$ 
48:              $W_{next} \leftarrow parentNode$ 
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:        $W \leftarrow W_{next}$ 
57:        $W_{next} \leftarrow \emptyset$ 
58:        $r_{max} \leftarrow d_{maxEst}$ 
59:     end if
60:   end while
61: end procedure
```

Equation.3.12 is also a constant. Therefore, by introducing the concept of window to each geodesic update, Algorithm.3 is able to achieve linear time complexity.

3.5 Geodesic on Point Cloud

Both Algorithm.2 and Algorithm.3 can be further extended for calculating the geodesics on point clouds easily.

Followed by the fast development of 3D shape acquisition device, 3D body scanner is becoming more and more popular in today's modelling applications. The most common shape representation that outputs by 3D shape acquisition device are point cloud. Therefore, it is important to be able to directly extract measurements data from point cloud model.

Despite the form of geometrical representation of the data, to calculate a geodesic, a source point must be specified in advance. The challenges of handling point clouds data is that, "Continue Dijkstra" propagation expand the "wavefront" from inward towards outward based on the connectivity between points. On polyhedral surface, this connectivity is represented by edges that connect vertices. However, on point clouds data, all the points are scattered and unordered. Therefore the "Continue Dijkstra" propagation cannot be performed on point clouds data directly.

In order to facilitate the connectivity of the points, a regular grid that covers the point cloud data set is employed. Within each cell that belongs to the grid, the mean point of the points that contained by this cell is regarded as the destination for all the points within this cell. This regular grid distributes destinations evenly over the point cloud data. Here, the concept of destination represents a location in a cell that from source point, all the geodesic that connected to the points in this cell are end at. Therefore, one destination

represented all the points contained by a cell, the number of destinations n is much smaller than the actual number of scattered points in point cloud data. Moreover, because all the cell in the grid are identical in term of size and shape, it is very easy to index a cell and its neighbour by using Equation.3.19.

$$index = \frac{(c_x - min_x)}{\Delta c} + x_{seg} \times \frac{c_y - min_y}{\Delta c} + x_{seg} \times y_{seg} \times \frac{c_z - min_z}{\Delta c} \quad (3.19)$$

where c_x, c_y, c_z are the 3D coordinate of the centre point c of cell C , min_x, min_y, min_z are the 3D coordinate of the minimum point of the bounding box of the point cloud data. δc is the interval of regular grid. Note that, to build a whole grid requires an additional $O(n^3)$ space. However, because each cell can be indexed directly by the space it covers, therefore no searching method is required to determine the 1-ring neighbour of a given cell. In the implementation of this algorithm, to optimize the memory consumption, an array C_{im} with $O(n^3)$ space cost only stores the index of non-empty cell, the actual data of the non-empty cell are stored into an dense array C with $O(n)$ space cost. Because the number of non-empty cells are much less than the number of the cell in a whole regular grid. The actual memory cost is close to $O(n)$. This process is summarized below.

After regular grid G has been built, the “Continue Dijkstra” is employed here to form the initial path for solving Equation.3.12. This process is illustrate below.

In Algorithm.5, in order to determine the neighbourhood for each sample point on a geodesic path, ANN searching method(Arya et al. 1998) is used on the scattered points. Assume that there are d nearest scattered points to some sample point on a geodesic. The normal vector of the tangent plane is calculated by performing PCA(Jolliffe 2002) to the d scattered points that within a fixed radius. In addition, causality of the propagation can also benefit

Algorithm 4 Build Regular Grid for Point Cloud

```
1: procedure BUILD REGULAR GRID FOR POINT CLOUD(A point cloud  
   data set  $S$ , a source  $p_s$  within the cell  $c_s$ , the intervals  $\Delta x$  of a grid, and  
   number of neighbours,  $d$ )  
2:   Build up a searching tree for ANN performing on  $S$   
3:   Select the minimum point  $p_{min}$  of the bounding box  $BBox$  of  $S$ .  $c$  is  
   the cell that contains the source point  $p_s$ .  
4:   while  $c \in BBox$  do  
5:     Get the closest point  $p \in S$  to the centre of  $c$   
6:     if  $p \in c$  then  
7:       Store  $c$  into regular grid  $G$   
8:       Use ANN fix radius search to get all the points  $C \in S$   
       that for every point  $p_c \in C$ , their distance to centre of  $c$   
       smaller than  $\Delta x$   
9:       Compute mean of  $C$  as the destination of  $c$   
10:    end if  
11:    Move  $c$  orthogonally by  $\Delta x$   
12:  end while  
13: end procedure
```

from the isotropicity of the regular grid. The propagation process is illustrated in Figure.3.11.

Algorithm 5 Approximate Geodesics Algorithm on Point Cloud

```

1: procedure INITIALIZATION(A Point Cloud data set  $S$ , a Grid  $G$  and a
   source  $p_s$  and the cell  $c_s$  contains  $p_s$ )
2:   for  $c_i \in G$  do
3:      $c_i.geoDist \leftarrow \infty$ 
4:      $C_i.geoPath \leftarrow \emptyset$ 
5:   end for
6: end procedure

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

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

```

Algorithm 5 Approximate Geodesics Algorithm on Point Cloud(continued)

```
31: procedure CONTINUES DIJKSTRA PROPAGATION( Calculate geodesic
    for every  $c_i \in G$ )
32:   while  $W \neq \emptyset$  do
33:     for  $c_i \in W$  do
34:       for  $c'_i \in oneRingNeighbourOf(c_i)$  do
35:         if  $c_i \in W$  OR  $c'_i.geoDist \neq \infty$  then
36:           continue
37:         else
38:            $parentNode \leftarrow GETPARENTNODE(W, c'_i)$ 
39:            $d_{est} \leftarrow distanceBetween(parentNode, c'_i) +$ 
              $parentNode.geoDist$ 
40:           if  $d_{est} < r_{max}$  then
41:             Perform Equation.3.12 on initialPath re-
               sults geodesic path from source to  $c'_i$ 
42:             Update the neighbourhoods for the samples
               point on  $c'_i.geoPath$  separately
43:              $W' \leftarrow c'_i$ 
44:           else
45:             Update  $d_{maxEst}$  if  $d_{est} > d_{maxEst}$ 
46:              $W_{next} \leftarrow parentNode$ 
47:           end if
48:         end if
49:       end for
50:     end for
51:      $W \leftarrow W'$ 
52:      $W' \leftarrow \emptyset$ 
53:     if  $W = \emptyset$  then
54:        $W \leftarrow W_{next}$ 
55:        $W_{next} \leftarrow \emptyset$ 
56:        $r_{max} \leftarrow d_{maxEst}$ 
57:     end if
58:   end while
59: end procedure
```

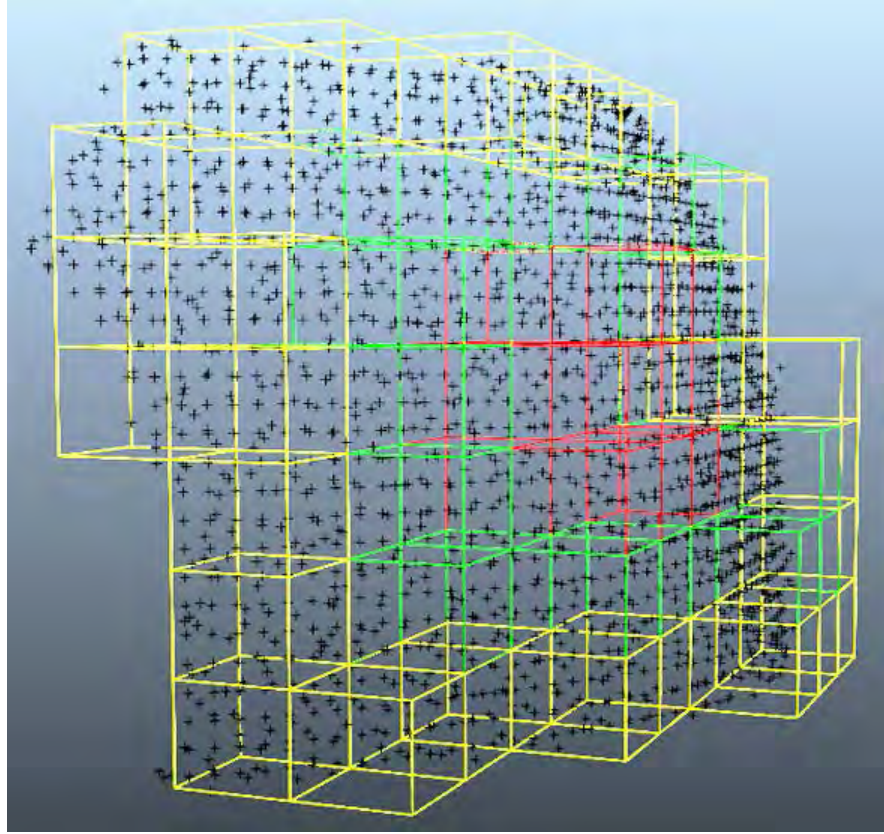


Figure 3.11: Regular grid covering a point cloud. Propagation is from the red cells to the yellow ones. In practice, we store the nonempty cells in an array. No need to store the whole grid.

3.6 Performance Analysis

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

3.6.1 Efficiency

MMP algorithm (Mitchell et al. 1987b; Surazhsky et al. 2005) defines the “Exact geodesic algorithm” as an algorithm that is able to produce geodesics that on a flattened planner, the geodesic path is a straight line. Because the

geodesics results from Alg.2 is done by minimizing the geodesic curvature on the smooth surface that the polyhedral surface is approximated from, therefore when Equation.3.12 converged, the geodesic curvature reach to the minimum(Butcher 2008; Hairer 2010). However, this curve is on the smooth with minimum geodesic curvature, (Polthier & Schmies 2006) point out that, if the projection of this curve does not pass any hyperbolic vertex on the polyhedron, the projected curve is a straight and shortest geodesic on the polyhedron. Therefore, in this thesis, Algorithm.2 is denoted as “accurate geodesic algorithm”.

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.3.17 is not a square matrix, it is necessary to left-multiply A^T on both side of the Equation.3.17 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.3.17, according to Bunch & Hopcroft (1974); Copper-smith & Winograd (1987), the LU solver has a time complexity if $O(m^{2.379})$. 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, the computation of the projection of each 3 successive sample points on a geodesic path onto the mesh faces costs a constant time to denotes at h , the projection of the whole geodesic path costs $O(kh)$ where k denotes the number of sample point 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 the matrices in Equation.3.17 is constant. Therefore, solving Equation.3.17 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.

For Algorithm.5, it employs ANN search on the point cloud data set. Let the number of the points in the point cloud data is N and the number of the non-empty cells in the regular grid is n . The ANN search for a given query point costs $O(\log n)$ time (Arya et al. 1998). Assume that there are at most d nearest neighbours for one query point by the ANN search. As a result, the ANN searching costs $O(d \log N)$ time on a point cloud. Thus, each geodesic thus costs $O(d \log Nn)$ time. Moreover, the time complexity for a single source geodesic computation is $O(d \log Nn^2)$.

Among the above presented algorithms, the most noteworthy achievement is the ability of computing geodesic paths in linear time in Alg.3 with a bounded error. This is especially significant for tracing a large number of geodesic paths and large 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 a 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 the MMP(Surazhsky et al. 2005), improved CH algorithms(Xin & Wang 2009) and FMM(Kimmel & Sethian 1998). In order to compare the time consumption for acquire the geodesic, “backtracing” is added into MMP, CH/ICH and FMM. The algorithm used for “backtracin” is presented in Surazhsky et al. (2005).

The source codes of MMP and ICH algorithms are available on the Internet¹. These algorithms are performed respectively on the Stanford Bunny’s model with 8 different resolutions as shown in Table.3.1. The running times are plotted in Figure.3.12.

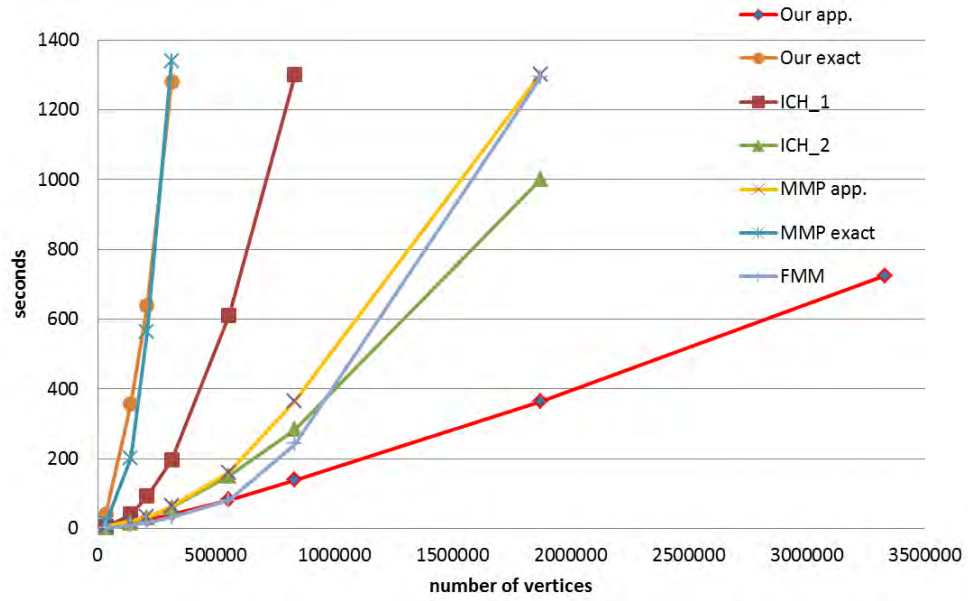


Figure 3.12: The comparison of running times. The times of the ICH and MMP algorithms include the cost of “backtracing”. The window size of our Algorithm.3, $w=30$.

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

Table 3.1: The resolution of the bunny models used in the experiment in Figure.3.12

For human model that used in this thesis for dressing, Algorithm.3 also

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>.

shows great advantage in terms of efficiency.

Furthermore, Algorithm.3, MMP approximation and ICH_2) are also performed on many large models for further comparison. We believe that these experiments provide some insight into the linear time complexity of our 1st order approximation algorithm against the MMP and ICH algorithms. The detail of this experiment please see in Appendix.A.

3.6.2 Accuracy

The accuracy of Algorithm.3 depends on the following parameters, the edge length e , the window size w , and the number of samples on each geodesic m . 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 the geodesic curvature flow of Equation.3.12 to this patch instead of the whole geodesic.

Let C_g denotes a geodesic on polyhedron that does not pass through any vertex p of the polyhedron unless p is the source p_s or a destination p_d . It can be observed that the C goes across a set of faces. Unfolding this set of faces into a plane, the C should be the straight line linking the p_s and p_d . For a new vertex q , we may combine the edge $\overline{p_d q}$ with the C as the estimation of the geodesic $p_s q$. Without loss of generality, let a window cover this set of faces and the number of faces be w . The error estimation for any geodesic computed by Algorithm.3 is illustrated in Figure.3.13,

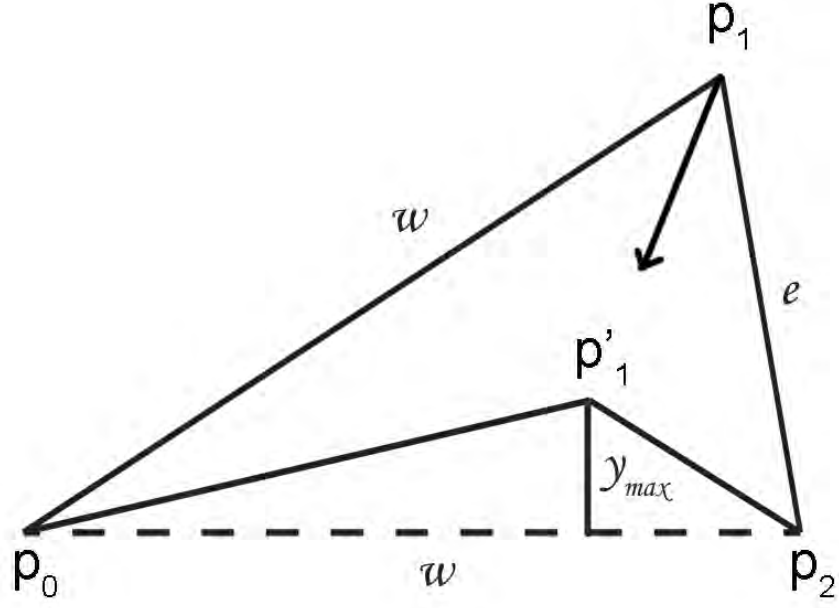


Figure 3.13: The estimation and desired geodesics. The $\overline{p_0p_2}$ is the desired geodesic while $\overline{p_0p_1}$ plus $\overline{p_1p_2}$ being the estimation.

Let p_0, p_1, p_2 are three point on a plane as shown in Figure.3.13. The initial path for solving Equation.3.12 is $\widetilde{p_0p_1p_2}$ where p_0 is source point and p_2 is destination point. Obviously, the true geodesic from p_0 to p_2 is $\overline{p_0p_2}$. Let $\overline{p_0p_1}$ is an established geodesic path, p_2 is an unvisited vertex. the length of edge $\overline{p_0p_1}$ is always smaller or equal than the length of $\overline{p_0p_2}$ since the propagation only performed on outward direction. Therefore, in the worst situation, as in Figure.3.13, $\overline{p_0p_2}$ and $\overline{p_0p_1}$ are equal. p'_1 is the updated point of Equation.3.12, 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, the based on the Heron's formula, the area of $\triangle p_0p'_1p_2$ can be written as,

$$T_{\triangle p_0p'_1p_2} = \frac{\sqrt{(L^2 - w^2)w^2}}{4} = \frac{wy_{max}}{2} \quad (3.20)$$

therefore,

$$L = \sqrt{4y_{max}^2 + w^2} \quad (3.21)$$

Therefore, let err denote the difference between true geodesic length and the solution of Equation.3.12, where err can be written as,

$$err = L - \|p_0 p_2\| = \sqrt{4y_{max}^2 + w^2} - w \quad (3.22)$$

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

$$err_{ave} = \frac{\sqrt{4y_{max}^2 + w^2} - w}{w} \quad (3.23)$$

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

$$err = m \left(\sqrt{1 + 4\left(\frac{y_{max}}{we}\right)^2} - 1 \right) < 2m \frac{y_{max}}{w} \sim O(\varepsilon m) \quad (3.24)$$

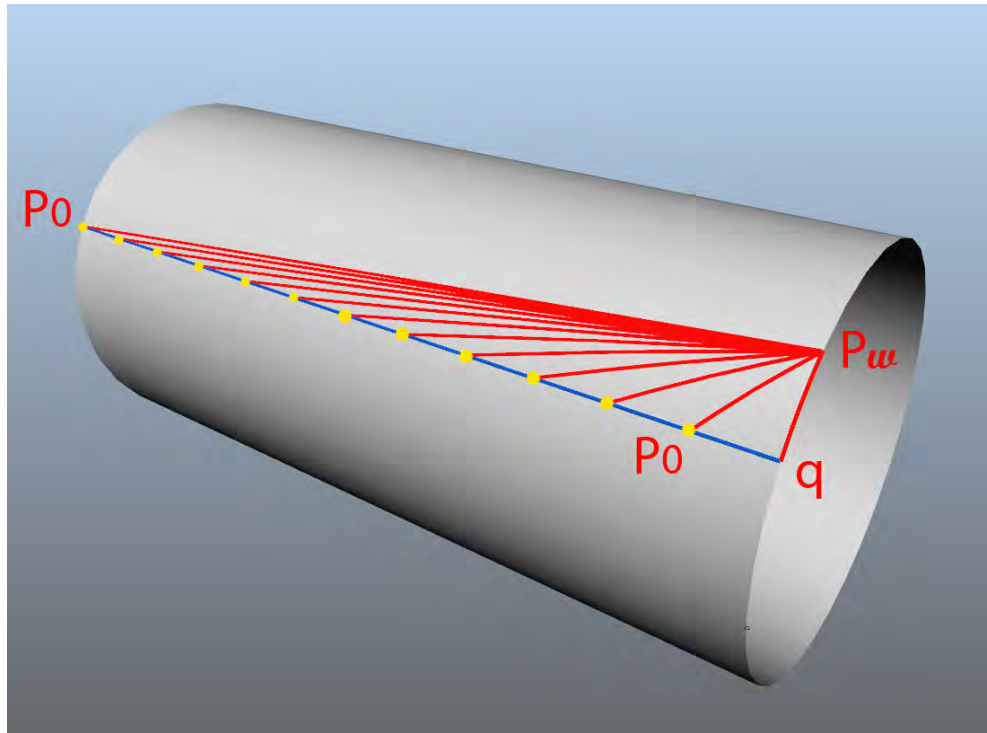
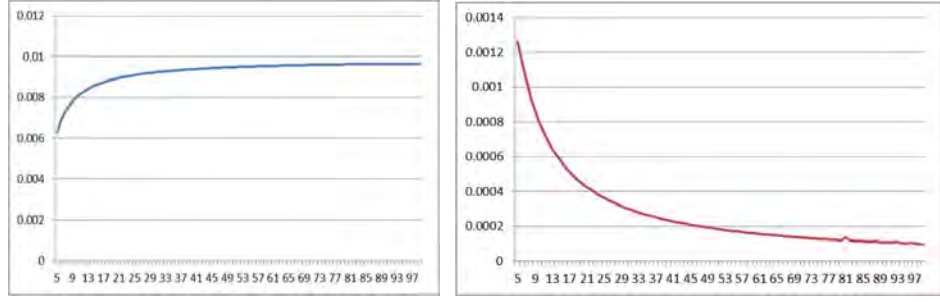


Figure 3.14: Error estimation of Algorithm.3 with different window size



(a) The convergence of y_{max} with varying window size (b) The convergence of ϵ with varying window size

Figure 3.15: Error of Algorithm.3

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 the y_{max} , we performed the Algorithm.3 on a cylinder surface with the edge length $e = 1$ and the window size w varying from 5 to 100 in Figure.3.14. The numerical results are shown in Figure.3.15a. It can be noted that when $w = 5$, the y_{max} tends to zero. But, the larger the window, the bigger the y_{max} . This can be explained that on the plane shown in Figure.3.13, the curvature of the geodesic estimation $\widetilde{p_0 p_1 p_2}$ is becoming small when increasing the window size w . The system of Equation.3.12 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.3.14, the p_0 is viewed as the start point of the window. If it is not the source, there must exist a geodesic from the source to the p_0 . It is ideal that the desired geodesic from the source to the p_2 passes through the geodesic from the source to the p_0 . Thus, it is natural to enlarge the window size as much as possible. Figure.3.15a and Figure.3.15b shows that both the y_{max} and the ratio ϵ convergence to small values. This means that the large w does not decrease the error significantly. The choice of the window size w should take into account the time of solving Equation.3.12 rather than the computational error.

		Hand #V:80089 #F:160000	Camel #V:34546 #F:69088	Jar #V:13358 #F:26681	Knot #V:28800 #F:57600
Ours Alg.2 (offset)	ave abs	0.00309	0.00098	0.00786	0.005218
	ave rel	0.5054%	0.1911%	1.3454%	0.4852%
Ours Alg.2 (proj)	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 3.2: *The average absolute and relative errors of Algorithm.3, MMP app. and ICH.2. The window size of our Algorithm.3, $w = 30$.*

In order to evaluate the accuracy of Algorithm.3 the MMP approximation and ICH.2 algorithms are also performed on 4 models, hand, camel, Jar and knot, which are representative models with complex topological and geometric structures.

MMP exact algorithm is performed on these 4 models and its solution are considered as the ground truth for the experiment and then calculated the absolute errors by the differences between the exact geodesic distances and approximate ones and the relative errors by the ratios of the absolute errors over the exact distances.

Note that for Algorithm.3 two results are kept as one is the offset geodesic paths and the other is the projections of the offset geodesics onto the meshes. Table.3.2 gives the average absolute and relative errors of these 3 algorithms. It can be noted that the offset solution of Algorithm.3 obviously outperforms the others, including the ICH.2, even though ICH.2 is regarded as the exact algorithm. However, the projections of the offset geodesics to the meshes

have no distinct difference compared to MMP approximation and ICH_2 algorithms. Moreover, to further validate the error estimation for Algorithm.3, MMP exact algorithm is performed on the lowest resolution bunny model in Figure.3.1 and use the length of geodesics result from MMP exact algorithm as ground truth.

The ratio ε in Equation.3.24 is usually a very small number. Figure.3.16 shows the histogram of the ratio ε over the bunny model, where the horizontal axis indicate the value of ε and vertical axis indicate the number of point in a geodesic.

Figure.3.17 further shows the distribution of the real absolute errors and the estimated ones in the bunny test. It can be noted that the error estimation of Equation.3.12 can accurately reflect the upper error bound of Algorithm.3

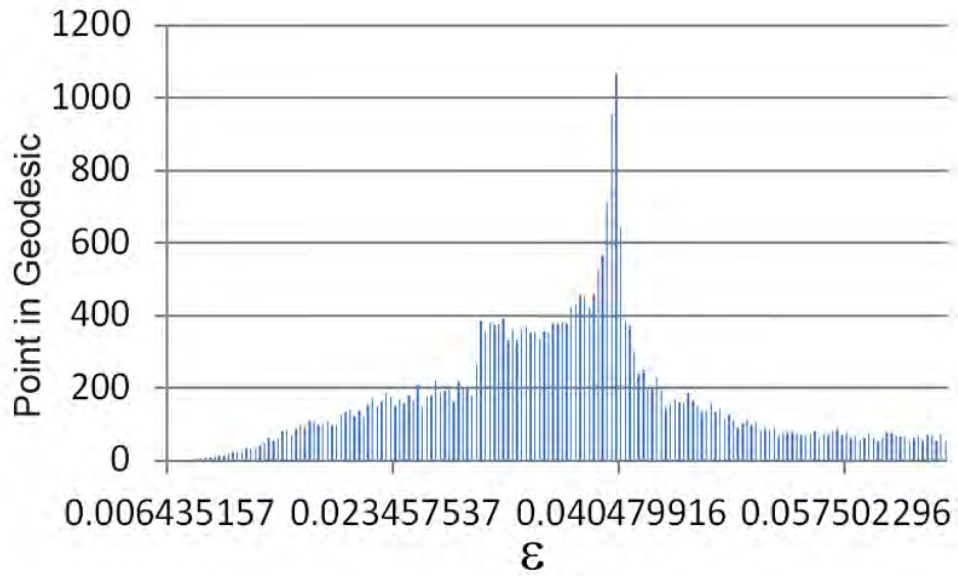


Figure 3.16: *Histogram of the ratio ε*

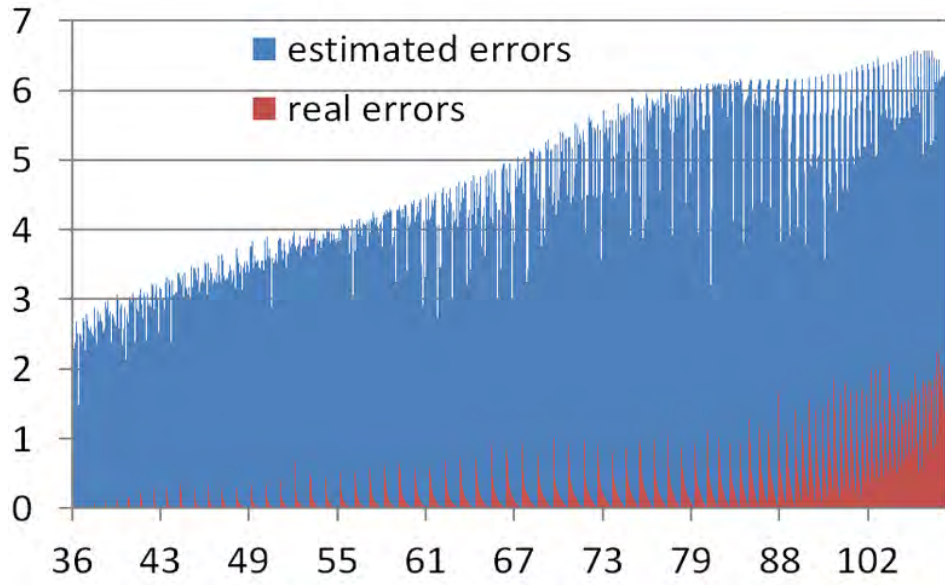


Figure 3.17: The distribution of the estimated errors and real errors. The window size of our Alg.3, $w = 30$. The x-axis indicates the number of the faces covered by a geodesic path.

Thirdly, the Algorithm.5 is preformed on three point cloud model, which are Stanford bunny, Buddha and sculpture, respectively for geodesic computation. The experimental results are shown in Table.3.3. Destinations are the means of the patches that are cropped by cells separately here. Obviously, they are fewer than the scattered points. Although the ANN searching leads to a $O(n \log n)$ searching time and requires extra space to store the searching tree in theory, it does not result in large time and space complexities in practice. Figure.3.18 shows shortest paths and isolines on the point clouds in Table.3.3.

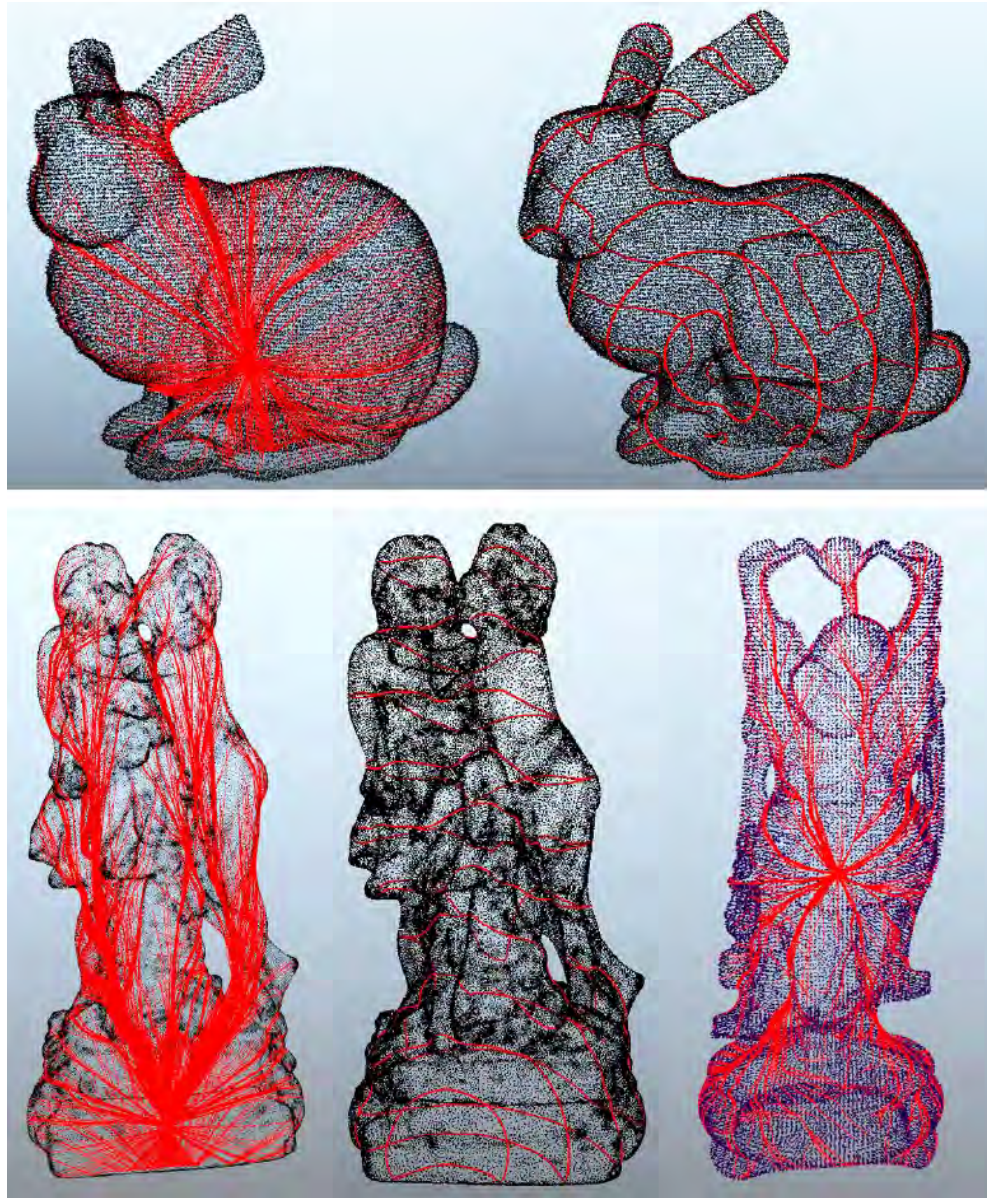
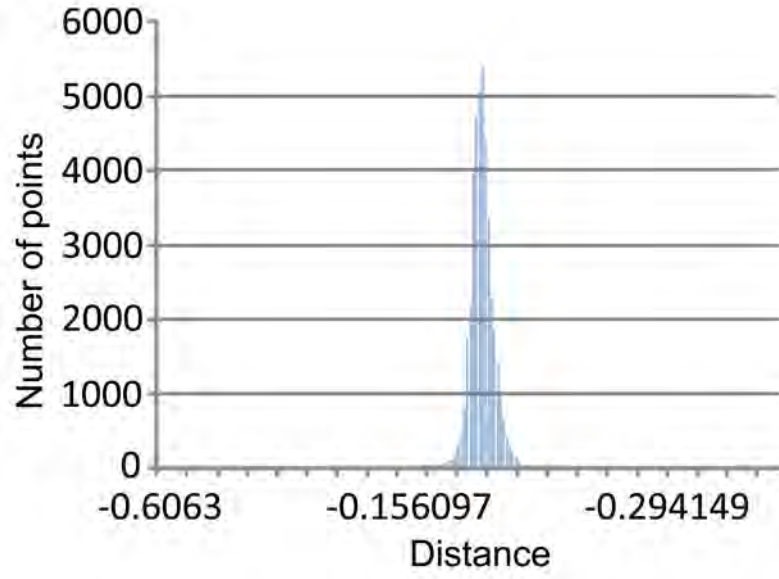


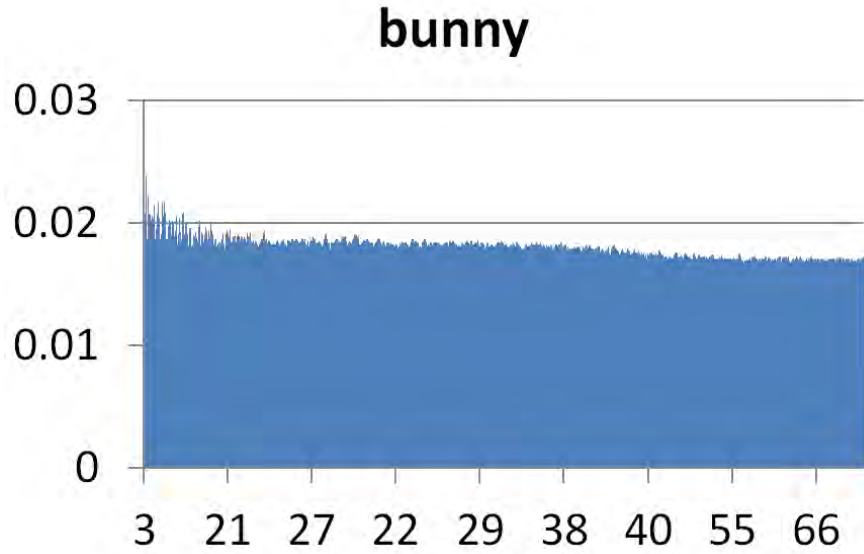
Figure 3.18: *Models used for tests in Table.3.3*

	Bunny	Sculpture	Buddha
#Points	69668	406224	841151
#Destinations	3748	10878	30177
Time(sec)	32.924	73.019	226.772
Memory(Mb)*	7.668	24.336	101.221

Table 3.3: *Performance of Algorithm.5. *Memory indicates the peak memory cost used in performing Algorithm.5, excluding the storage of the model itself.*



(a) The histogram of the offset distances of the sample points to the implicit surface of the point cloud bunny



(b) The distribution of the average relative errors of our Algorithm.5 on bunny. The x-axis indicates the number of the sample point in a geodesic path

Figure 3.19: Accuracy of Algorithm.5.

The accuracy of the resulting geodesic paths can be evaluated by the offset distances of the sample points on geodesics to the implicit surface of a point cloud. Figure.3.19b shows the histogram of the offset distances of the sample points to the reconstructed polyhedral surface of the bunny model.

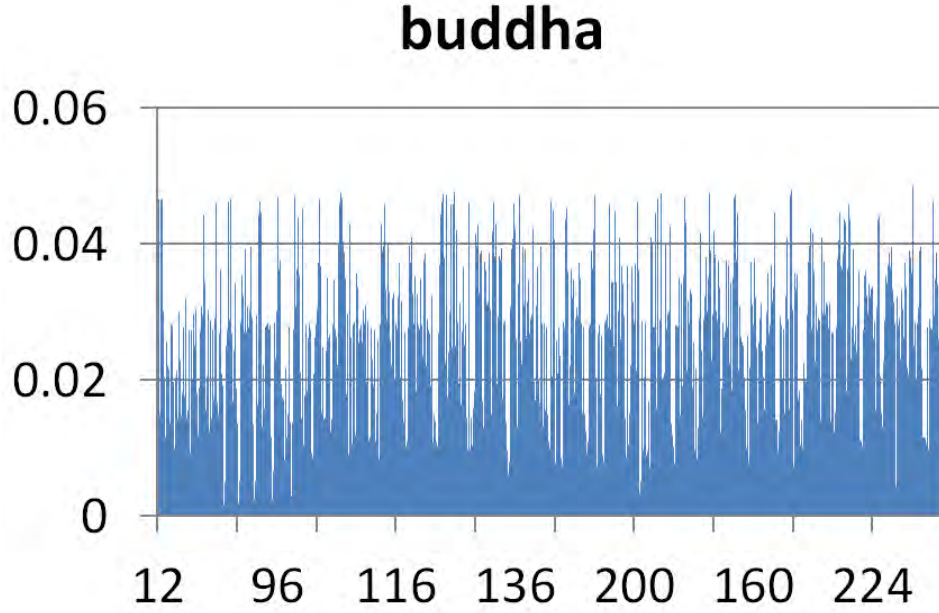


Figure 3.20: *The distribution of the average relative errors of Algorithm.5 on buddha. The x-axis indicates the number of the sample point in a geodesic path*

According to the Figure.3.19b, Algorithm.5 can guarantee the resulting geodesic paths close to the implicit surfaces of point clouds. Moreover, MMP exact algorithm is preformed respectively on the reconstructed Bunny and Buddha meshes and the results geodesics are regarded as the ground truth for this experiment. Figure.3.19a and Figure.3.19b show the distribution of the average relative errors of the Algorithm.5 on the two point clouds. It can be noted that the error on Buddha is obviously higher than that on Bunny. This is due to the fact that the model of Buddha' has more details than the model of Bunny. Further increasing the number of cells in the grid can decrease error.

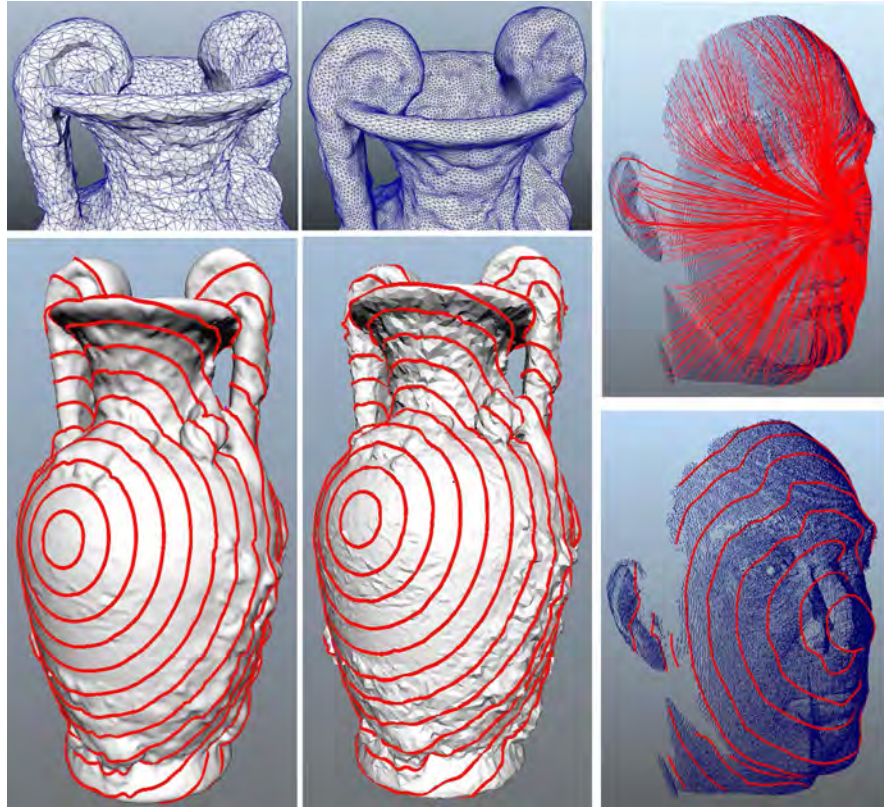


Figure 3.21: *Algorithm.3 performs on low quality mesh(left), high quality mesh(middle) and Algorithm.5 on point cloud data with small gaps*

3.7 Geodesic in body measurement

During the process of measuring, two types of measurements is used for describing the body dimension of the subject, circumference and length.

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

Table.3.4 listed out the major measurements are taken from the body of the subject. and illustrate the datum points for general measuring on a female figure.

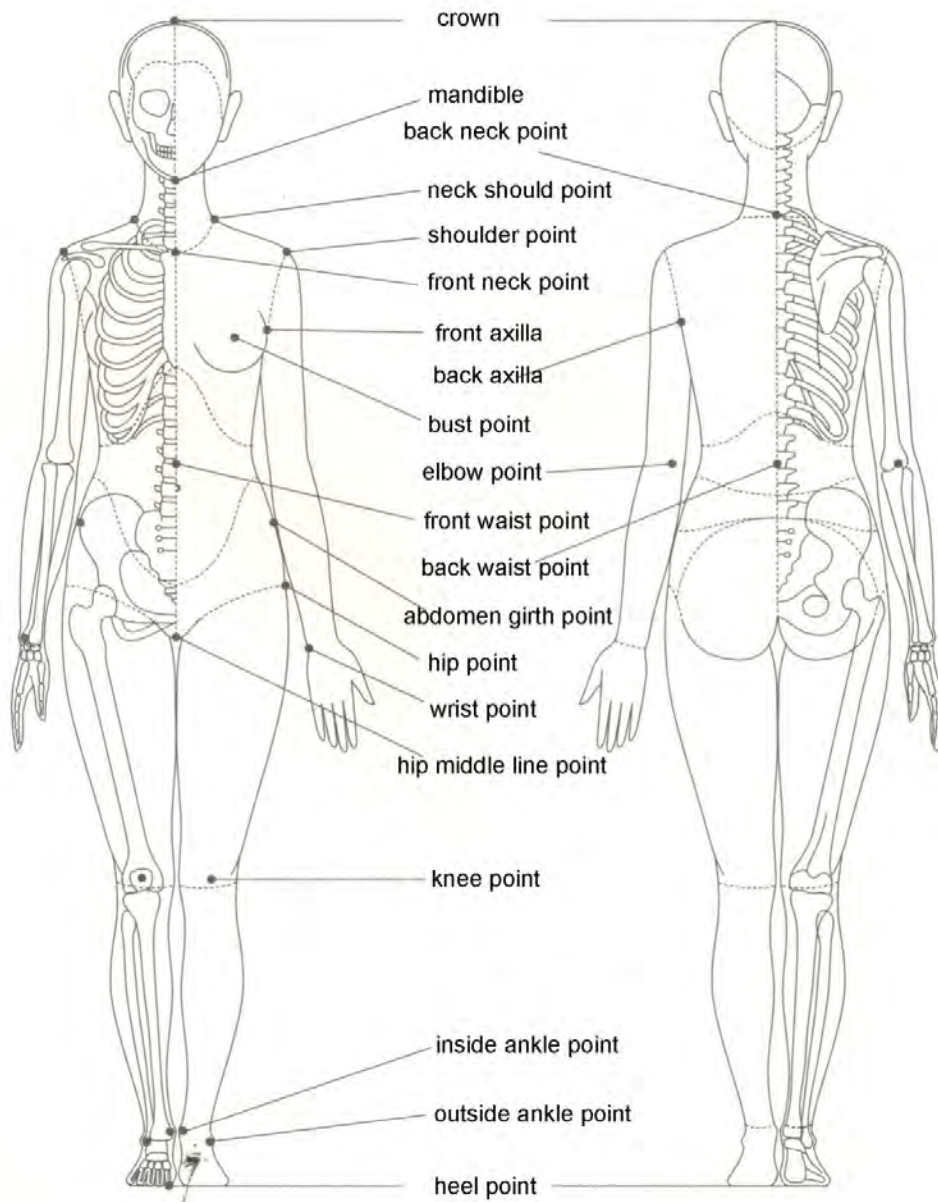


Figure 3.22: Datum point on a female body(Xiong 2008)

In this thesis, the circumference is measured by applying convex hull

algorithm(Graham 1972; De Berg et al. 2008) to the intersection of a cutting plane and the body part that are measured, then the arc length of the curve that connect all the intersect point is the circumference of the desired body part. Figure.3.23 demonstrate the bust girth and waist girth that are required for fitting a shirt on to the character.

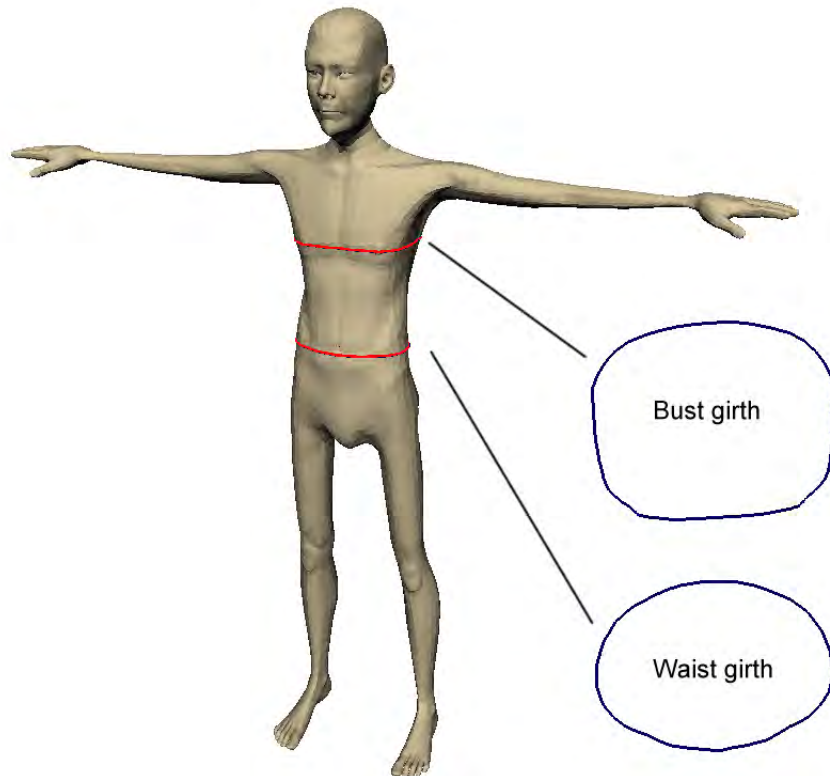


Figure 3.23: *Two circumferences measurement on character, two blue close curve indicate the convex hull of its corresponding measurements*

For length measurement, the geodesic algorithm presented in this chapter is applied to the body of the subject. With each required measurement, two datum that associated with the measurement is used as the source point and the destination point for the geodesic algorithm respectively. Figure.?? demonstrate the length measurements that required for fitting a shirt on to the character.

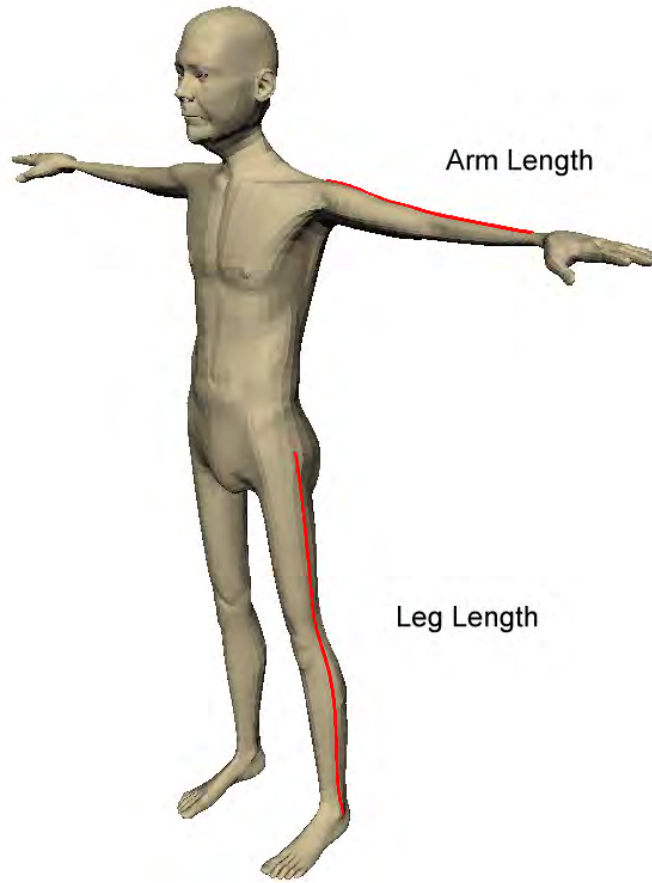


Figure 3.24: Two length measurement on character, the “Arm Length” is the length of the geodesic path from “Shoulder Point” to “Wrist Point”, the “Leg Length” is the length of the geodesic path from “Hip Point” to “Outside Ankle Point”, for the datum points, see Figure.3.22.

3.8 Conclusions

In this chapter, three algorithm respectively for accurate and approximate geodesic computation on triangulated manifolds as well as approximate geodesic computation on point cloud data set are presented. The most important contribution is that the proposed approximation algorithm can reach linear time complexity with an bounded error on triangulated manifolds. Numerical com-

parisons with existing algorithms (i.e. MMP, ICH_1 and ICH_2) have further demonstrated the advantages of our algorithms in terms of both speed and accuracy. By integrating Algorithm.3 into the measuring system, the time consumed on solve the geodesic path between multiple pair of source and destination has largely reduced, moreover, the accuracy of the solution is maintained. In the final chapter , more experiments will be presented to illustrate the advance of the algorithm presented in this chapter.

Chapter 4

Virtual Cloth modelling and Re-targeting

This chapter presents a method for automatically modelling clothes based on the measurements of subject. The required measurements are acquired by technique presented in chapter.3. Based on the measurements, the shape of the cloth patterns are automatically adjusted for the subject. The presented method is able to create cloth for different subjects with different body proportion without involving the tedious process that used in the traditional modelling and generates well fitted clothes efficiently.

4.1 Introduction

Modelling virtual cloth is a well known time consuming process that requires many expertises to achieve. It usually took an well trained modeller several hours to model a complete set of cloth. Normally, when creating a cloth from patterns, the 2D cloth patterns have to be created, then they must be place onto the 3D human model. To stitch patterns together, cloth simulation need to be performed. If some part of cloth need adjustments, this process

need to be performed iteratively till the cloth fit to the subject. Moreover, if this cloth need to be dressed onto other subject who has very different body proportions, this tedious process would need to be repeated again from scratch. When using geometrical modelling method to create cloth, that is, to model cloth polygon by polygon. The whole process looks easier than pattern based modelling technique because the cloth can be modelled directly onto the subject therefore the fit is guaranteed. However, this advantage only exist when dealing with single character, the geometrical modelling method also suffers from same drawback when dealing with multiple character. The whole modelling process must be repeated to fit to a subject with different body proportion.

Followed by the fast development of computer hardware and cheap storage, more and more characters can be handled by computer at the same time. Consequently limited by current cloth modelling technique, dressing a large number of different subject is still considered as a very tedious and time consuming process.

However, In garment manufacturing, through generations of development, cloth pattern has become an important medium that transfer cloth design into wearable object. Cloth patterns are the basic items that consists a functional garment. Based on the cloth pattern, a complex process called pattern grading is developed to adapt the need for dressing general public. Cloth pattern grading aims at adjusting the cloth to fit different wearer while preserve the original design of the cloth Moore et al. (2001). Moreover, modern cloth pattern is firstly designed to follow certain measurements of the ideal figure to maximize the generality. Then by providing measurements of the body of an specific subject, the shape of every pattern is adjusted respectively to fit the new body. This is the general procedure of made-to-measure tailoring. The automatic cloth resizing and re-targeting method presented in this chapter is inspired by this process.

To implement a cloth design, the design needs to be broken down into a set of textile pieces first, then through the assembling technique such as stitch and adhesive, all the pieces are assembled together in a certain sequence to form a complete cloth. The process that breaks down a cloth design into pattern is called "patternmaking" (Armstrong 2000). Through this process, cloth design is transferred from design concept into the combination of a set of patterns that are easy to be cut from the raw textile material. Therefore, the same cloth design can be reproduced easily (Hannah 1919).

4.1.1 Cloth Patternmaking

Armstrong (2000) defines the cloth patterns as the template from which the parts of a garment are traced onto fabric before cutting out and assembled. Pattern grading is the process of systematically increasing and decreasing the dimensions of a pattern into a range of sizes for production. In garment industry, only one size pattern is developed and fitted for one cloth design, then other sizes are graded from these patterns. This process not only retains the original design of the cloth during the distribution but also very cost effective during the process of manufacturing (Moore et al. 2001).

When creating patterns, blocks (also known as slopers) are created first. Blocks are two-dimensional templates that consist the basic design of a type of garment and they are constructed based on the measurements taken from wearer (Armstrong 2000). In the massive production of garment industry, blocks for a cloth design are usually created by using the measurements from a size chart which is usually based on a particular ethnic or group of people who share the similar body proportion, for example, US standard clothing size chart (ISO/TR-10652 1991) contains the body proportion for the general public of 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 to the blocks. Figure.4.1 demonstrate the block for a woman's shirt.

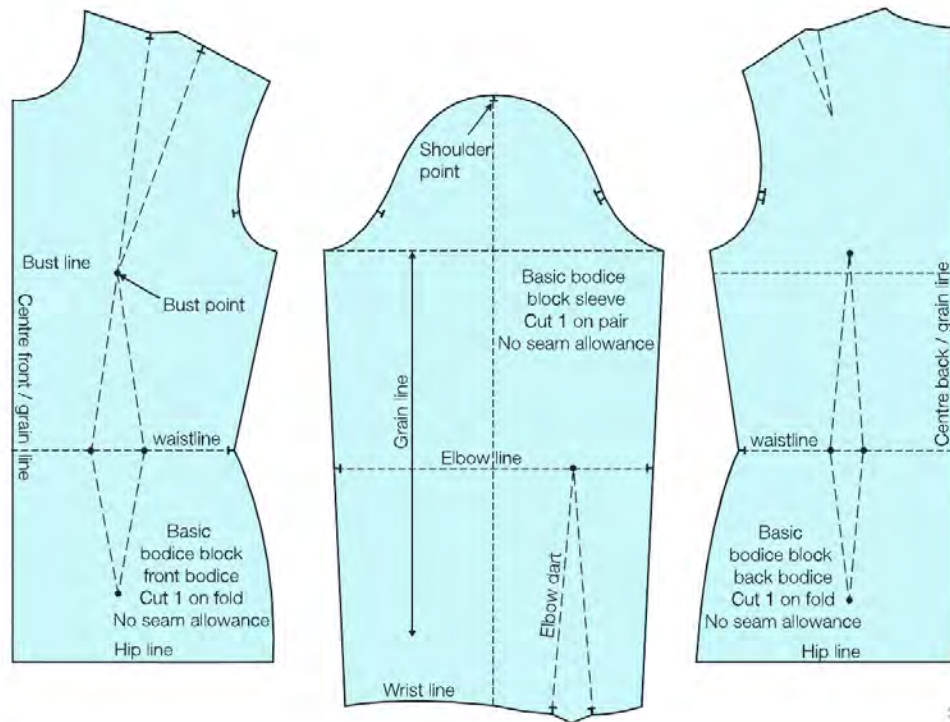


Figure 4.1: *Garment block for woman shirt(Rosen 2004).*

When measurements are taken from an individual, it provides a good indication of the body dimension that the cloth design is intend to fit. In made-to-measure tailoring industry, measurements acquisition is particularly important since it directly determines the fit of the garment. Therefore, this process are usually performed by an experienced tailor.

Normally, only the standard measurements are used for block construction. During pattern grading, patterns that derived from blocks are scaled proportionally into different size with predefined intervals introduced by a size chart. This method has been proven very cost effective for making cloth for general public due to the people that fit a predefined size are normal distributed into an interval introduced by a size chart(Schofield & LaBat 2005).

However, in the virtual world the situation is much complex from what it is in reality. In reality, for each ethnic or certain group of people, the similarity of the body proportion can be determined. In virtual world, the body proportion is only limited by the imagination of the artist, therefore to use the size chart to perform proportional scaling onto patterns to fit the subject no longer applies to the virtual character. Nonetheless, as shown in Figure.4.1, on every block, there are several important landmarks that associate with the datum point on the body of subject. By matching the distance that in-between the landmarks on the block with their associated measurements that acquired from subject, the cloth that made from the block is able to fit to the subject with any body proportion.

4.1.2 Block Resizing Criteria

In pattern grading, all the pattern are scaled proportionally(Moore et al. 2001) to preserve the shape of pattern respectively. However, as aforementioned, the body proportion of virtual character can be largely different than human. Therefore, proportional scale all the pattern might leads to an undesired result.

Figure.4.2 indicates same shirt design being dressed onto two character with same height and lime length using proportional scaling method presented by(Moore et al. 2001). For the reason of the length of their corresponding body parts are similar, both of the character falls into the same grading intervals. However, the character on the right is much stronger than the one on the left, his body supports shirt much better than the one on the left. On the contrary, the skinny character has much thinner body in which results a baggy shirt. During made-to-measure cloth tailoring, extra measurements from the major measurements are taken from the body of an subject in order to produce the block with much higher accuracy such as "Arm Hole Length" or "Cuff Girth". With the correction brought by extra measurements, differ-

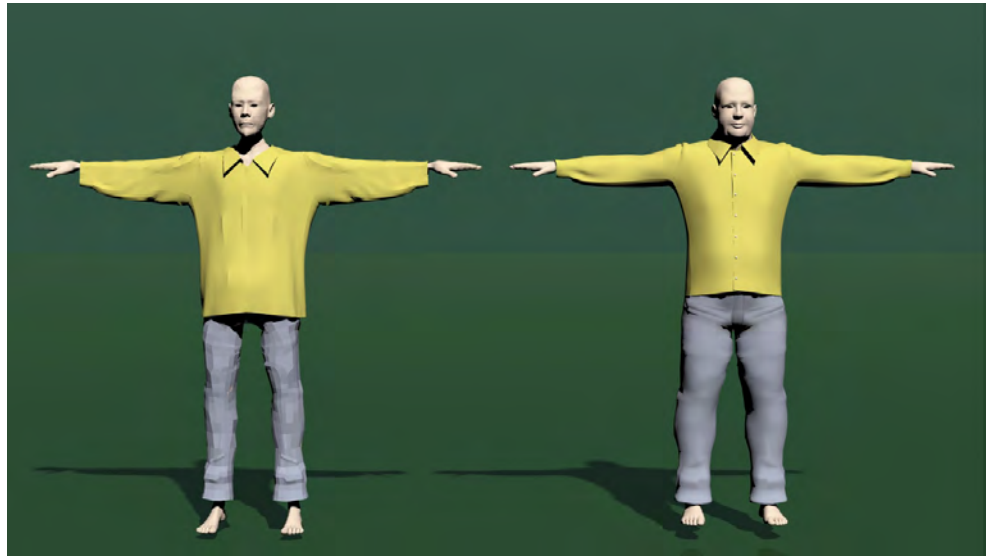


Figure 4.2: *Same shirt design proportional scaled onto two character with similar body proportion*

ent part of a block might scale in a different manner. This non-proportional scaling to the block is the biggest different between bespoke cloth making and garment massive production.

Because the body proportion of a virtual character can be vastly different. The traditional pattern grading method no longer meet the needs of dressing any character in virtual world. Therefore, inspired by made-to-measure cloth tailoring technique, a method that is able to produce bespoke cloth for virtual character is presented in this chapter. In order to generate block based on the measurements taken from virtual character, a few geometric criteria need to be evaluated in order to ensure the fit of the cloth.

Target Measurements One of the major goal of this research is to construct cloth for multiple characters. Because cloth pattern represents different part of garment corresponds to different part of human body, therefore all patterns need to be deformed to a manner that that match to the measurement data of is corresponding body part and further achieve the garment fit. Because all the patterns are derived from blocks,during

the pattern resizing, for each block that associated with patterns, the distance between all the landmarks that associated with the datum on the body of the subject need to satisfy the requirement set by measurements taken from the subject. Especially when the subject has an unusual body proportion, different part of a block might scale in different fashion. For the instance of the sleeve for demonstrated in Figure.4.2, the arm of two character have the same length, however circumference of the arm are much different, Moreover, despite the difference between the circumference the arms, the wrist girth are same for both character. Therefore, in order to satisfy the all the measurement requirements, different part of sleeve pattern need to be adjusted in different manners.

shape Cloth pattern is the most basic component that consists a complete garment, the shape of patterns defines the design of cloth. The pattern grading is the considered as the most difficult step during tailoring in which only few people is able to master. For professional, the overall shape of patterns can be altered significantly to accommodate the difference in proportions. However, the details of the pattern such as slops, or 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 one cloth onto different character without altering the design of the cloth. However, each cloth pattern is deformed to match the measurements data from a part of body. Therefore a shape evaluation need to be performed after the adjustment of 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 stitch point that located on two adjacent patterns. During the adjustment of pattern, either the target measurements matching or patter shape maintaining is per-

formed individually, therefore there is no guarantee that after the adjustment of the patterns, the seam-line remains consistence. Because seam-line determines the location of the patterns after it constructed into 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, the automated cloth resizing method is explained in detail. This method operates directly on blocks and optimize each block to ensure all the patterns satisfies all the criteria introduced in previous section.

Two kinds of input are required for this method. The first input is a group of patterns represents a cloth design is required, within each pattern, the landmarks that associates with the datum point on the subject are defined. Moreover, the sewing information in which consist of seam-lines are presented along with the pattern. Note that the size of the patterns is irrelevant to the final modelling result. Figure.4.3 demonstrates an example of the patterns.

The second input is the 3D model of the subject. This method requires the body and limes of the subject stays in a stretched and relaxed posture such as the “T-pose”. Because this method involves physical simulation for the the sewing process, the contact of the limb will enlarge the probability of the penetration between cloth and the skin of the subject. Moreover, the measurements of the subject are also provided by the method presented in the previous chapter.

In order to properly determines the size of patterns, the block for each pattern are generated at first. Because block is the “initial pattern” of all the cloth patterns, all the landmarks that associates with datum point on the

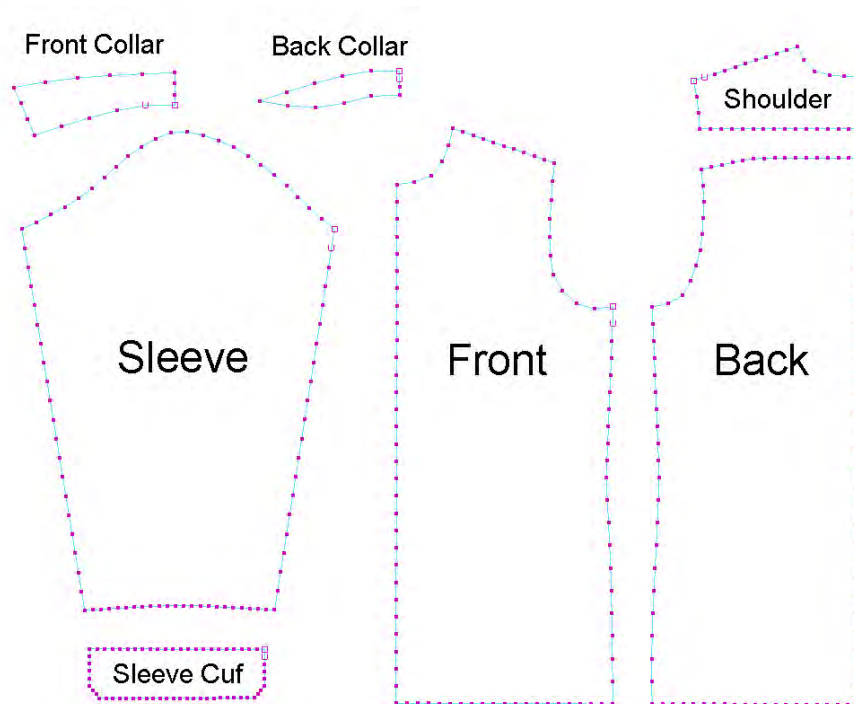


Figure 4.3: *An example of the cloth pattern*

body of the subject are defined on the blocks. Moreover, because block and pattern has strict one to one association which means for any pattern it only generated from one unique block. Also, because block contains much less details than pattern, performing resize algorithm on blocks will cost much less computational time than performing this algorithm on pattern directly.

When constructing a block from a pattern, firstly, the “critical points” of the pattern is selected. The “critical point” is defined by points which are most capable of representing the geometrical feature of a pattern such as the sharp turning point on the contour of the pattern. Figure.4.4 illustrates the “critical points” on sleeve pattern. After that, a bounding box is created for the pattern, within the bounding box, for every “critical point”, two orthogonal line are created to form a grid. This is demonstrated in Figure.4.4.

In next step, based on the grid, a nurbs plane is generated in which the control points of the nurbs plane is the intersection point on the grid(this is

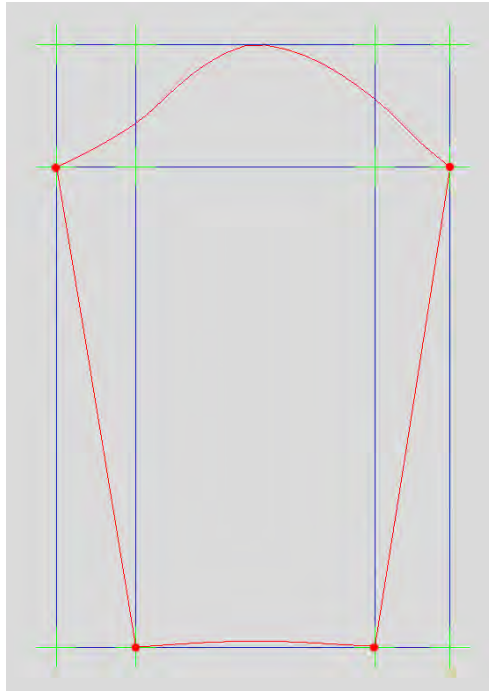


Figure 4.4: Block generated from sleeve pattern(red closed curve). The red points are "critical points" of sleeve pattern.

demonstrated by the green cross in Figure.4.4). All the points on each pattern can be represented by a parametric coordinate on nurbs plane. For every pattern, a unique nurbs plane is created, therefore, this nurbs plane can be reckoned as the block of the given pattern. The traditional cloth grading technique applies proportional scaling to the pattern. In general, the measurement that associated with the length of the limes are used as the reference of the scaling such as "Height", "Back Length" and "Arm Length". Figure.4.5 demonstrate the result of performing proportional scaling to sleeve block based on the "Arm Length" measurement.

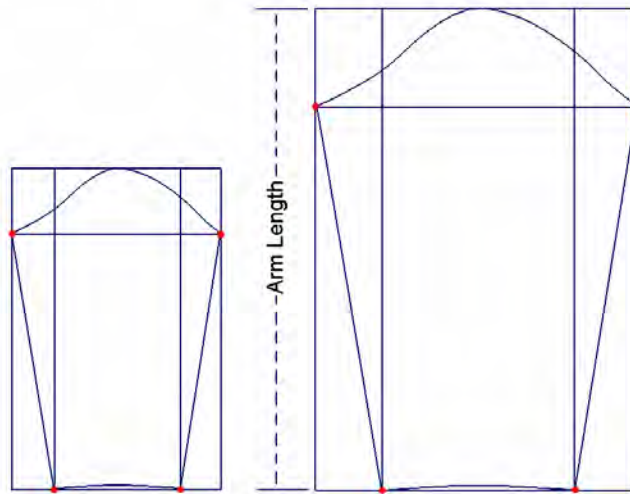


Figure 4.5: Block before(left) and after(right) the proportional scaling. The height of the sleeve pattern after scaling has reached the measurement “Arm Length”

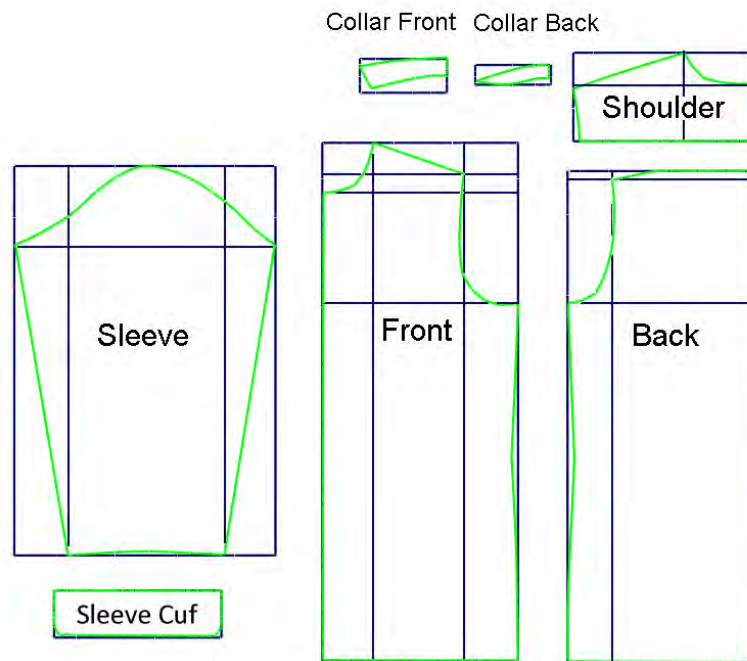


Figure 4.6: Blocks after proportional scaling is performed.

The proportional scaling has several drawbacks that need to be improved. Firstly, in order to perform the proportional scaling, at least one

measurement must be provided as scaler. However, for some patterns, such as “collar”, it does not have measurements directly associated with therefore proportional scaling can not be performed. As depicted in Figure.4.6, “Collar Front” and “Collar Back” does not associate with any measurement taken from the character, Therefore, they cannot be resized by proportional scaling method. Moreover, for the pattern who cannot be resized based on the measurements, their size usually depend on the patterns that they are stitched to. For example, the “Collar Front” is stitch to “Front” and “Collar Back” is stitched to “Shoulder”, therefore, the size of “Collar Front” is depends on the size of “Front” and the size of “Collar Back” is depends on the size of “Shoulder”.

Because “Front”, “Back” and “Shoulder” are scaled by using different measurements, different proportional scaling is applied on them. This might leads to inconsistency of the seam-line between “Front”, “Back” and “Shoulder”. Moreover, because the size of “Collar Front” and “Collar Back” depends on the size of “Front” and “Shoulder”, the inconsistency of the seam-line will be enlarged when “Collar Front” and “Collar Back” is resized based on the “Front” and “Shoulder”. Based on the experiment of this research, all most every pair of patterns that need to be stitched together has the same issue as ”Collar Fount” and “Collar Back”, that is, given a pattern P_b that stitches to P_a , if there is no measurement that can determine the size of P_b directly. Then the size of P_b can only be determined by the length of the seam-line $S_{a,b}$ that connect P_b with P_a . Moreover, the length of the seam-line $S_{a,b}$ can only be determined once the size of P_a is confirmed by M_a . However, if there is another pattern P_c that stitched to both P_a and P_b by $S_{a,c}$ and $S_{b,c}$. Furthermore, there is a measurement M_c are directly affects the size of P_c . Under this circumstance, the size of P_c is determined by multiple conditions simultaneously. In many cases those condition may not concord and in some cases even conflict to each other.

4.2.1 Genetic Algorithm

The genetic algorithm(GA) is an optimization and search method that based on the principles of natural selection(Haupt & Haupt 2004; Deb 2001). During the natural selection, effected by the environment, the biological traits of the organism becomes more or less common in a population through the generations of reproduction. A genetic algorithm allows a population that consisted of many individuals to evolve under certain rules and to a state that minimizes the cost function(Holland 1992). It has many advantages over the traditional optimization method,

1. Has the ability to search through a wide sampling of the cost surface simultaneously.
2. Has the ability to deal with a large number of variables.
3. Has the ability to deal with very complex cost surfaces and avoid local minimum.
4. Is well suited for distributed system.

A genetic algorithm starts from a group of random generated solution called “initial population”. Within the population, every individual is a set of variables that represents a solution of the problem. Then by evaluating every individual using the cost function, a ranking is assigned to the individual in terms of cost. After that, a selection method is applied to selection a group of individuals to perform the mating and mutation. Finally, the convergence of current generation is checked. If the current generation of population did not reach the minimum of the cost surface, the process goes back to step one and executed the algorithm iteratively till the minimum cost is reached, this procedure is demonstrated in Figure.4.7

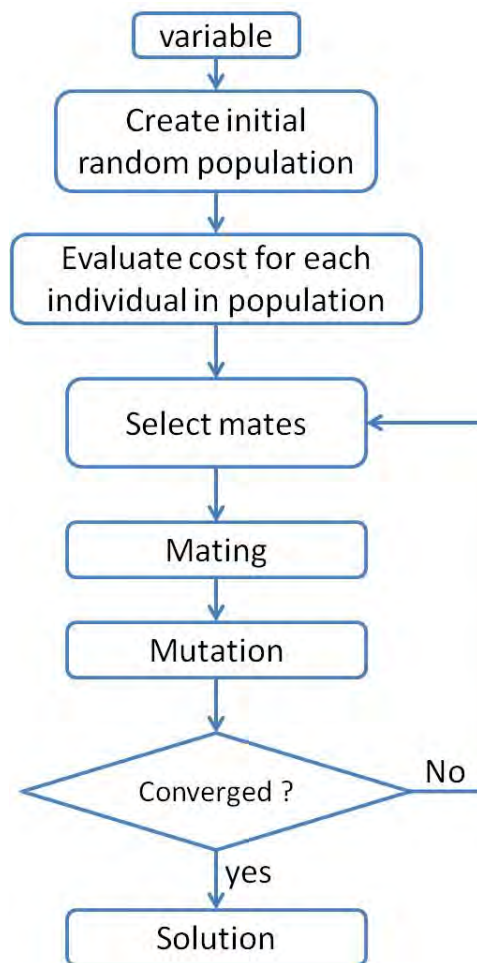


Figure 4.7: *The general flowchart for single objective genetic algorithm*

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

optimal solutions. After the Pareto-optimal solutions has been found, usually, a higher-level objective is used to select one solution out of the Pareto-optimal solutions.

In the rest part of the chapter, a multi-objective genetic algorithm designed for pattern resizing is presented in detail. This algorithm uses multi-objective optimization to find the best combination of the patterns that best fits the measurements and the seam-line while maintaining the shape of pattern.

4.2.2 Definition of Population

Within a population, individual is the basic element represents one solution of the problem (Haupt & Haupt 2004). In our case, all the pattern are described by a set of points located on the contour of the 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, $Block_i = [cp_0, cp_0 \dots cp_m]$. After a block is created based on the a pattern, all the point of the pattern can be interpolated by the parametric coordinate on the block plane. Moreover, every pattern has its unique block, for each pattern, it can be represent uniquely by the control point of corresponding block plane. Note that $m \ll n$, that is the number of the control points is much less than the actual number of the points on the pattern, particularly, in the experiments, m ranging from 4 to 30 and n ranging from hundreds to a thousand. Consequently, a pattern can be represented by the control points of the block very efficiently. Therefore, the gene in the this algorithm is consisted by a $N_p \times N_{cmax}$ size matrix and its chromosomes are illustrated in Table.4.1,

$$\left\{ \begin{array}{l} p_0 = cp_0 \quad cp_1 \quad cp_2 \quad \cdots \quad cp_m \\ p_1 = cp_0 \quad cp_1 \quad cp_2 \quad \cdots \quad cp_n \\ p_2 = cp_0 \quad cp_1 \quad cp_2 \quad \cdots \quad cp_o \\ \vdots \\ p_n = cp_0 \quad cp_1 \quad cp_2 \quad \cdots \quad cp_p \end{array} \right. \quad (4.1)$$

Table 4.1: *The structure of gene*

In Table.4.1 the gene of an individual. cp_i denotes a control point of block p_i , an individual consists of a group of block $ind = p_0, \dots, p_n$ which a complete cloth are made from.

The initial population is the first sample over the cost surface. Therefore, its distribution is very important because an adequate sampling distribution many reduce the time of finding the solution and prevent premature convergence in a local minima. Moreover, If the initial population can be defined as close as possible to the desired result, searching range of the cost surface can be largely reduced and further leads to the improvement of the performance of the algorithm. Therefore, before performing the genetic algorithm the 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 $M = [m_{i_1}, \dots, m_{i_n}]$. For each stored measurement, it contains the 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.4.8

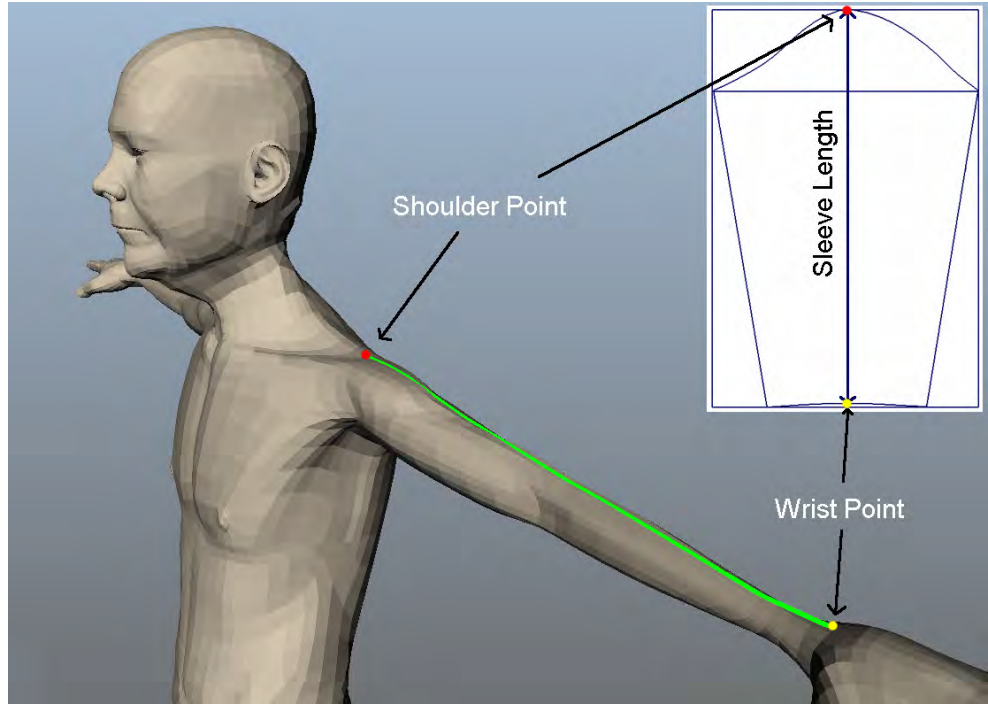


Figure 4.8: Association between body datum point and patter landmarks. The green line is the geodesic from shoulder point(red point) to wrist point(yellow point). The red and yellow point on the sleeve pattern is the pattern landmarks that associated with the datum points on the body of character. When sleeve is well fitted, the "Sleeve length" should match the length of this geodesic.

The scaling factor can be calculated by Algorithm.6.

Algorithm 6 Initial Pattern

- 1: **procedure** PROPORTIONALLY SCALE PATTERN(A pattern P , The measurements of Character M)
 - 2: **for** measurement $m_i \in M$ **do**
 - 3: p_s and p_e is the two datum points of m_i
 - 4: lm_i and lm_j is two landmarks on the pattern that $lm_i \sim p_s$
and $lm_j \sim p_e$
 - 5: $d \leftarrow$ distance between lm_i and lm_j
 - 6: $Sr \leftarrow m_i/d$
 - 7: **end for**
 - 8: Select largest Sr as the scaling factor for P
 - 9: **end procedure**
-

By using Sr as the scaling factor, the input patterns are proportional scaled. Although the proportional scaling cannot produced a well fit cloth for

subject, it is able to resize the patterns to a certain degree so that the cloth is roughly fit to the character. Therefore 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 unit Gaussian distributed random value is added to every control points of block.

In order to improve the efficiency of the evolution, the individual can age during the evolution. In nature, the size of population changes over generations because the individual who carries a fit gene has better mating opportunity than those who carries the less fit gene. Therefore, a fit gene is able to live longer during evolving than 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 the bilinear method that introduced by Michalewicz (1996) for calculating the 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 a gene in the population. $cost_e$ denotes the expected value of the cost for current gene and $cost_i$ is the actual cost value for current gene. When a gene reached its life span, it dies when evaluating the cost. During evaluation, if lots of individuals died in the same 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, the dead individuals are replaced by newly mutated individuals to maintain the coverage over the cost surface.

4.2.3 Crossover and Mutation

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

Crossover is a exploitations to a certain area of cost surface. it requires two individuals to participate the process and generates two offspring. In order to perform crossover, two parents need to be selected from the population first, the difference between two parents need to be large enough to create a effective offspring, because two similar parent will results a very similar offspring. This will result over dense sampling in a small region near the parents, and leads to the premature convergence. When selecting parents, an interval δi is defined for the index of individual. For example, let an individual ind_i is selected as the mother, the father will be select from $ind_{i+\delta i}$. For the full crossover, $\delta i = 1$, for partial crossover, a random integer is assigned to the interval. After the parents are selected the recombination of 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 the father side and cm_i is the chromosomes from mother side. Normally, crossover generate two offspring, in which both child carries both part of gene from their parents.

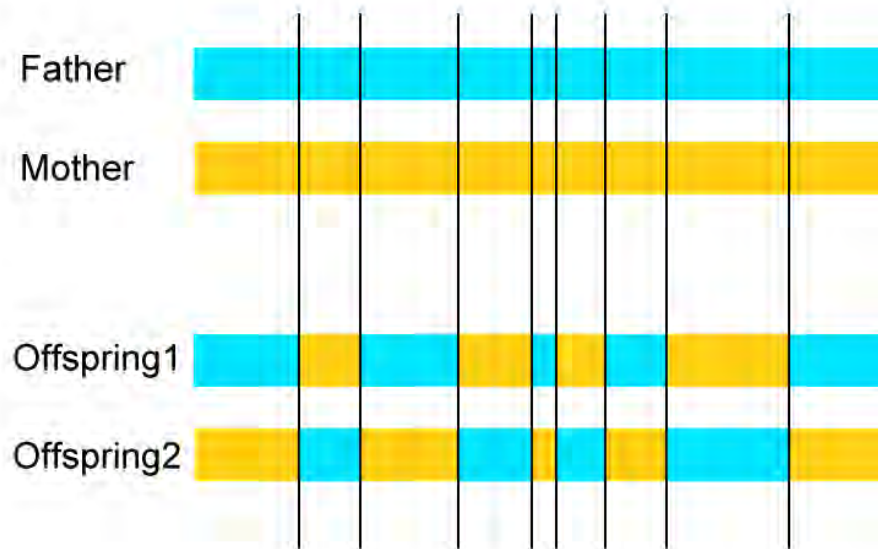


Figure 4.9: *Crossover of parents*

Figure.4.9 demonstrate the uniform crossover method(Spears & Anand 1991; Gwiazda 2006; Ghosh & Tsutsui 2003), it uses a fixed ratio exchange rate between two parents. the black line indicate the exchange point in gene that are randomly chosen by a random number generator.

In this algorithm, every individual consists of a group of different patterns, when two individual mates, the crossover only performed on the same pattern in both side of parents. The crossover method is summarized in Algorithm.7.

Because crossover operates on the existing gene pool that consists from two parents. It samples the area near the parents much more denser than mutation. Therefore, a higher crossover rate will increase the speed for convergence of the genetic algorithm. In real world, a species will extinct without introducing new gene because the limited gene pool will exhausts during the new born of individual. In genetic algorithm, this phenomenon appears as the

Algorithm 7 Crossover

```
1: procedure CROSSOVER(Two individuals  $ind_1$ ,  $ind_2$ , crossover rate  
    $cxb$ )  
2:   if  $cxb < random()$  then  
3:     for  $block1_i \in ind_1$  and  $block2_i \in ind_2$  do  
4:        $N \leftarrow$  number of control point in current block  
5:       for  $i = 0; i < N; i++$  do  
6:         if  $rand() < 0.5$  then  
7:            $offspring[i] \leftarrow block1_i$   
8:         else  
9:            $offspring[i] \leftarrow block2_i$   
10:        end if  
11:      end for  
12:    end for  
13:  end if  
14: end procedure
```

over dense sampling to a local area around the initial population. This often results premature termination of the evolution.

Mutation is a process that new chromosomes 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 the new are on 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 mutation. Grefenstette (1986); Srinivas & Patnaik (1994) point out that the choice of the mutation rate is heavily problem specified. For different problems, best mutation rate varies significantly.

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 stage of the the evolution, the genetic algorithm usually requires different exploration-exploitation ability. In this

algorithm, the non-uniform mutation, 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 the seed, a very wide distribution over the sampling cost surface is preferred, therefore, large mutation rate is set. Before evolution comes to the end, more exploitations is preferred rather 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 evolved. Assume that Gen_{max} is the predefined maximum number of generations of genetic algorithm. Then, for each individual, the randomly chosen chromosome cp_i is replaces by on 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 current generation, γ is a normal distributed random number, $\Delta(gen, t) \in [0, x]$ is a random variable that mutates cp_i with 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 uniformly distributed random value from 0 to 1, Gen_{max} is the number of generations for 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.8

Algorithm 8 Mutation

```
1: procedure MUTATION(An individuals  $ind_i$ , Generation index  $gen$ , Mu-
   mutation rate  $mutpb$ )
2:   if  $mutpb < random()$  then
3:     for  $block_i \in ind_1$  do
4:        $dpb \leftarrow random()$ 
5:        $U \leftarrow$  the number of control point on  $U$  direction
6:        $V \leftarrow$  the number of control point on  $V$  direction
7:       if  $dpb < 0.5$  then  $\triangleright$  Variate  $cp_i$  in  $U$  direction
8:         for Each row of  $cp_j$  on  $block_i$  do
9:           Append  $u$  value of  $cp_j$  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  $cp_j$  on  $block_i$  do
16:          Assign  $u$  value of  $cp_j$  with corresponding element
          in  $Variables$ 
17:        end for
18:      else  $\triangleright$  Variate  $cp_j$  in  $V$  direction
19:        for Each column of  $cp_j$  on  $block_i$  do
20:          Append  $v$  value of  $cp_j$  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  $block_i$  do
27:          Assign  $v$  value of  $cp_j$  with corresponding element
          in  $Variables$ 
28:        end for
29:      end if
30:    end for
31:  end if
32: end procedure
```

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 control points or on V direction only for a selected column of control points. The choice between two directions are randomly chosen for each block. Assume a row of control points is selected from an block. The U value of each control points is stored into an array denotes as U_{row} . The same row of control points on the original patter is also selected and U value of every control points is also stored into an array denotes as U'_{row} . For each pattern, its block is created as a rectangle nurbs plane that the pattern is inscribed to. Every row or column of control points of the block are in the same incremental or decremental order. this is demonstrated in Figure.4.10.

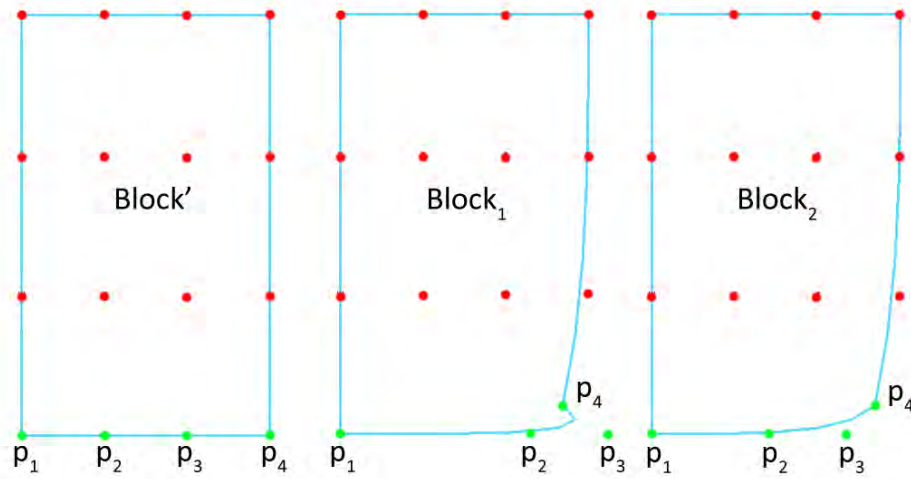


Figure 4.10: *Block'* is at its initial status, *Block₁* and *Block₂* are the results of mutation operation applied on the bottom row of control points(green points).

When a block is at its initial status, such as *Block'* indicated in Figure.4.10, the order of the control points in selected row is denoted by $R' = [p_1, p_2, p_3, p_4]$. *Block₁* and *Block₂* are two possible result generated by mutation operation on *Block'*. Sorted by the U coordinate, the order of control point in the corresponding row can be written as $R_1 = [p_1, p_2, p_4, p_3]$ and

$R_2 = [p_1, p_2, p_4, p_3]$. In order to avoid the overlapping of the mesh in the pattern that generates from the block, the control points in any row or column need to be maintained during the mutation. Since $R' \neq R_1$ and $R' = R_2$, $Block_1$ is an invalid gene of current individual. In order to model cloth, each pattern need to be trianglized first to from the polyhedral surface from the out-line of the pattern. In the case of $Block_1$, because the order in R_1 is changed, therefore, it will cause the face overlap during the triangulation of the pattern. therefore, the mutation need to be reapplied till $R' = R_1$, such as $Block_2$ is a valid gene of current individual.

4.2.4 Evaluation and Selection

In genetic algorithm, fitness function describes the objective of the algorithm. Given a solution, the fitness function is used to measure how far the current solution to the desire goal of the convergence. When a population evolves into a new generation, the selection operator delete the n worst individuals and breed 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 goal of the algorithm, this is generated by applying the fitness function to test each solution. In this algorithm, objectives for the evolution is to adjust the size and shape of the each pattern so that each of the pattern fulfil the requirement of the measurement data. Also, among patterns, the topology of the seam-line that two patterns joint together remains. Most importantly the consistency of cloth design must be retained after resizing. Therefore, three fitness functions are developed for evaluating each individual respectively.

Measurements evaluation

Given a block, landmarks are associate with the datum points on the body of the subject. 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.

$$Error_m = \frac{\sum_n \|M_n - Dist(lm_a, lm_b)\|}{n} \quad (4.5)$$

Where M_i denotes the measurements associate with the current block. lm_a and lm_b are two landmarks on the block that associate with M_i . n denotes the number of measurements that determines the size of current block.

Seam-lines evaluation

A cloth cannot be made into the right design without correct sewing. It is the most important method that joint 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 chapter, a stitch refers to a constrain that attaches the vertex on two pattern together and a seam-line is consisted with the 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 transition. Therefore, in this algorithm, the seam-line evaluation consists with two criteria, the angle at each point and the length of each edge on either side of seam-line.

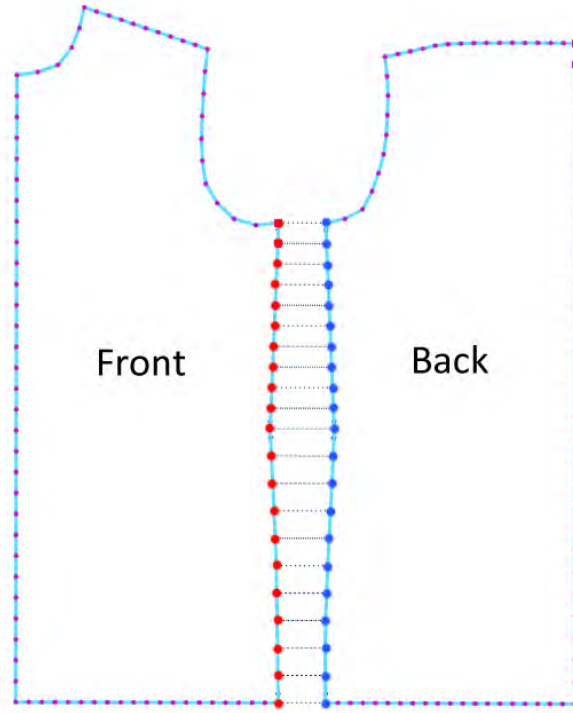


Figure 4.11: The seam-lines between “Front” pattern and “Back” pattern. The red points indicate the seam-line on the front pattern and blue are the seam-line on the back patter.

Let $sline_f$ denote the seam-line on “Front” pattern that depicted by the red points and $sline_b$ denotes the seam-line on “Back” pattern that depicted by the blue points in Figure.4.11. Both point in a point pair that connected by a constrain has the same index in its corresponding seam-line. Given a point pf_i on the $sline_f$, the greatest included angle θ between $\vec{pf_{i-1}pf_i}$ and $\vec{pf_{i+1}pf_i}$ is recorded. Next, the point at the same location on $sline_b$ is also recorded as θ' . Therefore, the angle criteria can be written as Equation.4.6.

$$Error_{angle} = \sum_{i=0}^n (\|\theta_i - \theta'_i\|) \quad (4.6)$$

Where n denotes the number of point pairs in the seam-line. One of the other criteria is the edge length in seam-line. For a pair of seam-lines with the same total length, different point distribution will results undesired

tension along the seam-line in which causes wrinkle to occur around seam-line. Within every seam-line, all the point are stored in an predefined order, thus in order to ensure the consistency of the distribution of the point in the seam-line pairs, the edge that connected two successive points in a seam-line need to be identical to the one in the other seam-line. Therefore, the edge length criteria can be written as Equation.4.7.

$$Error_{edge} = \sum_{i=0}^{n-1} (\|l_i - l'_i\|) \quad (4.7)$$

Where l_i denotes the length of the edge $\overline{p_i p_{i+1}}$ in a seam-line and l'_i denotes the length of the edge $\overline{p'_i p'_{i+1}}$ in another seam-line. n denotes the number of the point in a seam-line. Therefore, the seam-line evaluation outputs the cost at 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. Thus, the consistency of the shape for each pattern need to be kept through out the evolution to ensure the cloth that fit to a subject remains the original design. The shape of a geometry can be described via many methods. In this thesis, the definition that introduced in Kendall (1977) is used which is “Shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object”.

In this algorithm, two shape fitness function are developed for evaluate pattern shape at different stage of the evolution. 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 see Figure.4.12.

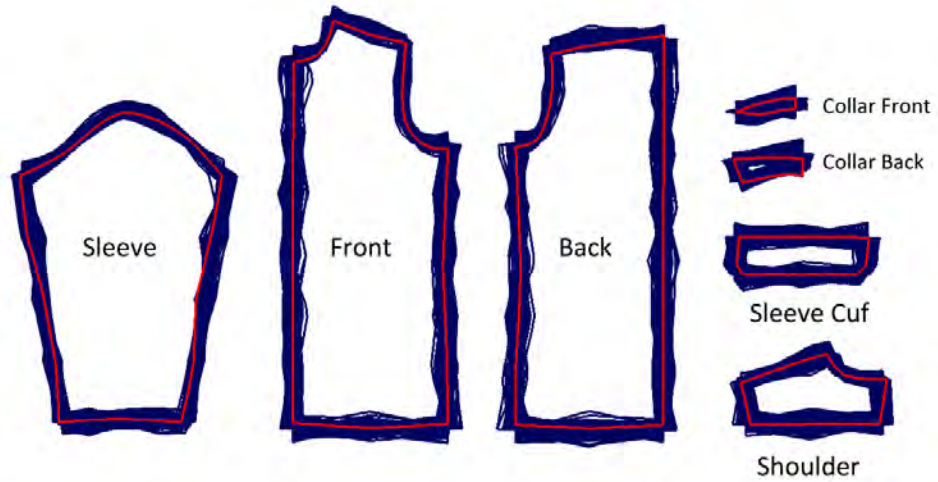


Figure 4.12: *Shape evaluation for the first generation patterns, Blue lines indicate the variation of the patterns, and red line has the best shape preservation.*

Because this algorithm solves a multi-objectives optimization problem in which the objective usually conflict to each other and results a set of solutions called “Pareto front” instead of single best solution. Within this set, each solution is not dominated by the rest of the solution of this set (Michalewicz 1996). Lots of real problem in real world are multi-objective problem that requires 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 constructing the “Pareto front”, then a higher-level objective is used for find the “best” solution among the “Pareto front”.

When resize a cloth, given a subject, only one size of the cloth is needed for dressing the subject. To select one solution from “Pareto front”, the shape similarity between original pattern and resized pattern is used as the higher-

level objective to select one solution from the “Pareto front”. To perform such a process, the angle at each “critical point” is calculated before the evolution starts. After the form of “Pareto front”, the angle difference between resized block and original block at each “critical point” is the criteria for selecting one solution from “Pareto front”.

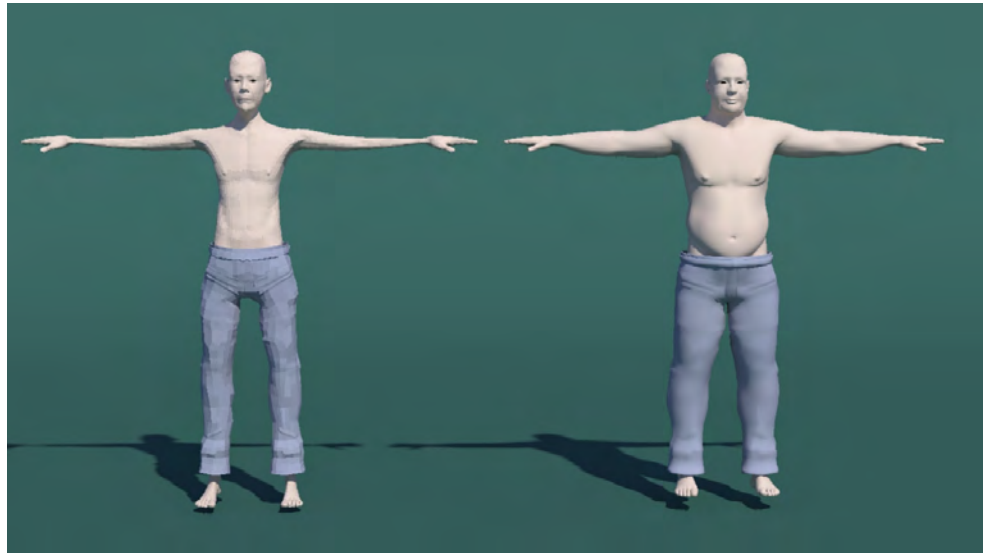


Figure 4.13: *Character on the left has much thinner arms than character on the right.*

For the character on the left, because the arm is much thinner than the one on the right, “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.4.14 demonstrate the “Front” pattern resized for the character in Figure.4.13. The pattern is selected based on the minimum changes of inner angle of the “critical point” of solutions. The design of the shirt is preserved. Figure.4.15 demonstrate the fit shirt modelled by the method presented in this chapter

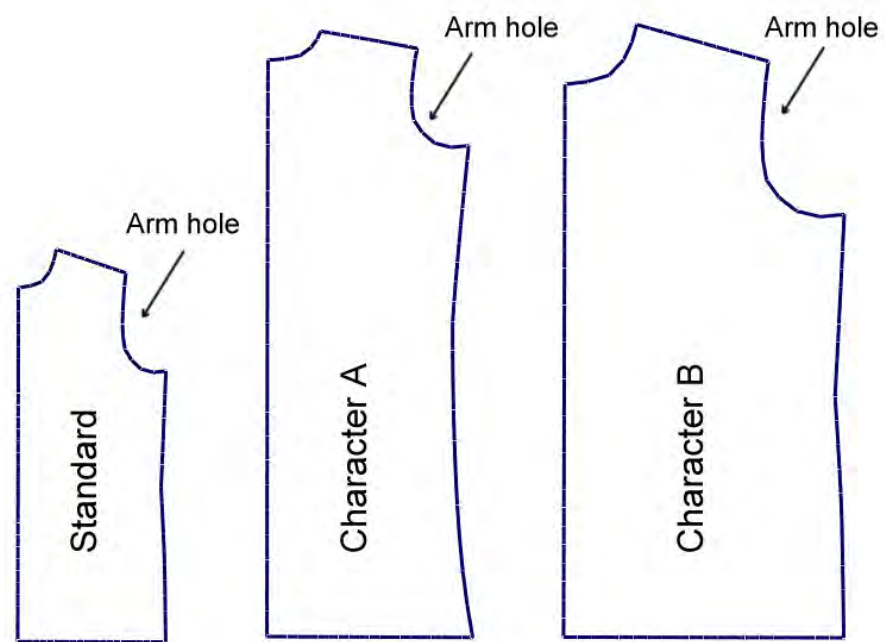


Figure 4.14: *The standard pattern(left), the “Front” pattern for the character on the left of Figure.4.13(middle), the “Front” pattern for the character on the right of Figure.4.13(middle)*

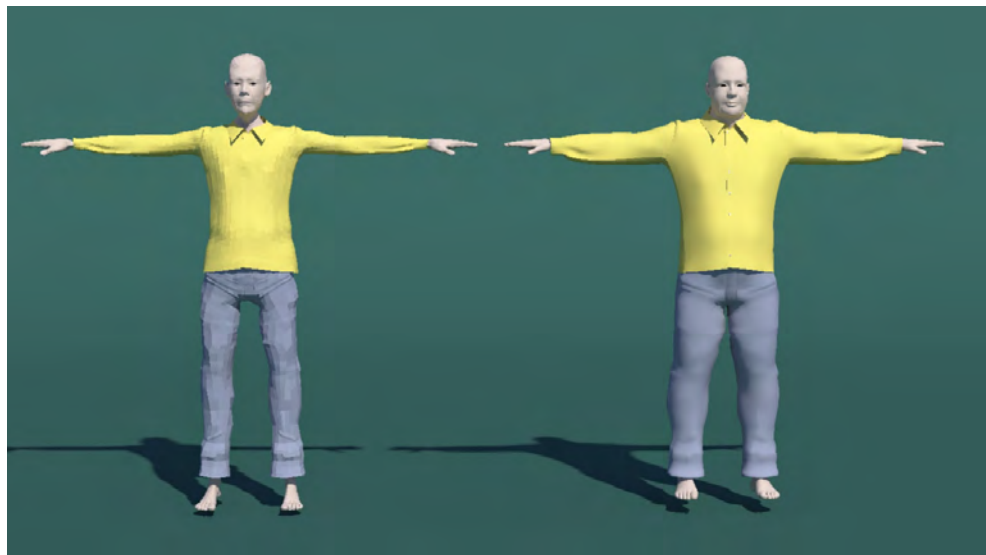


Figure 4.15: *Fit shirt modelled for characters*

Selection

Selection decides the individuals that are fit enough to survive the natural selection and reproduce child in the next generation. In single objective optimization, solutions can be sorted by fitness to the objective. However In multi-objective optimization, because the 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 objective 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 the 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 than solution B in at least one objective.

To demonstrate the concept of domination, Figure.4.16 illustrate an optimization problem with five solutions $s_1...s_5$ and two objectives f_1 and f_2 .

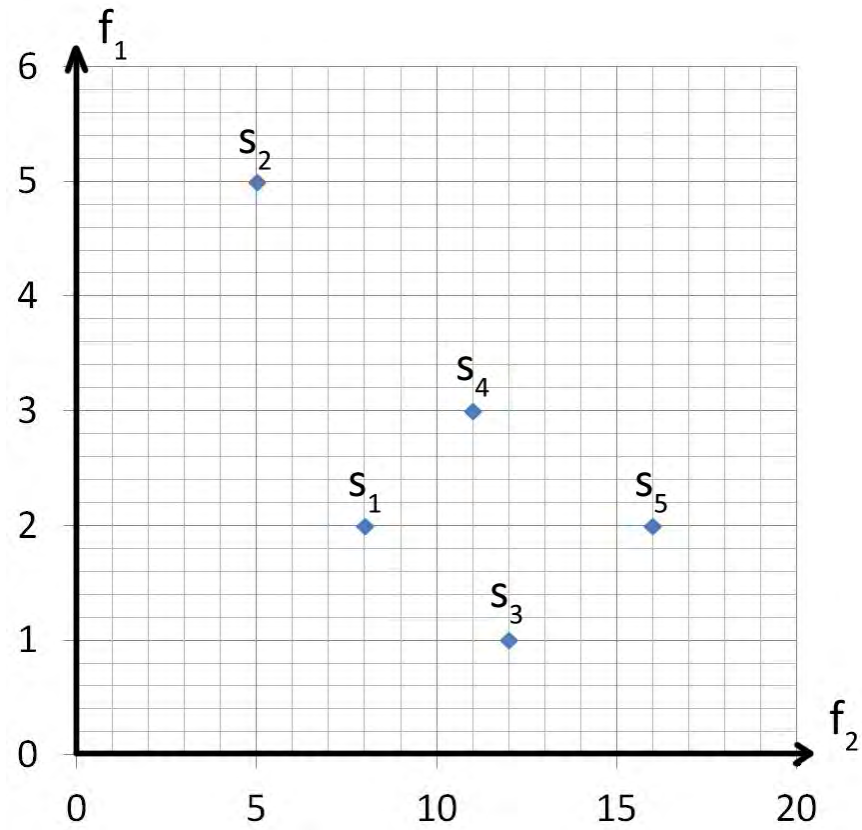


Figure 4.16: *Domination between five solutions*

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.4.16, when comparing s_1 and s_2 , s_1 is better than s_2 in both f_1 and f_2 . Thus, s_1 satisfy 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 two conditions of the definition of domination, therefore, s_1 is dominated by s_5 .

Deb et al. (2002) introduced non-dominating sorting algorithm that uses this method to select all the solution in a population that does not dominated

by the rest of the population to construct “Pareto front”. 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 s_3 and s_5 consist “Pareto front”.

In this chapter, all the solutions are compared according to the definition of domination in terms of three fitness functions, that is, measurements, seam-line and shape to form the “Pareto front”. In the final step, the shape objective is used for selecting one solution with less shape distortion as the final solution of the algorithm.

4.3 Pattern assembling

After every pattern are adjusted into the correct shape and size based on the measurements from the subject, they are ready to be placed on the subject and stitched together to form a complete cloth. In order to achieve such a task, all the patterns are positioned onto the 3D character simultaneously before sewing take place. Three steps are involved for the construction of the 3D cloth. Firstly, the bounding surface of the body part of subject are created, then, patterns are triangulated and then based on the parametric coordinate of the bounding surface, all the vertices of the 2D pattern are transferred into 3D space. Finally, according to the seam-line information, constraints are created to every point pairs in the seam-line and physical simulation is performed to give the final shape of the cloth.

The concept of the bounding surface is based on the ideal of cross-section of the human body can be approximately reckoned round in shape. For this reason, the cylindrical surface is chosen as the bounding surface. Cloth patterns are developable surface because they are cut out from a flat-

tened big piece of textile materials. Therefore, to construct a developable surface around the body of the subject first they transfer the patterns that are also developable onto it can minimize the distortion during the pre-positioning.

To create bounding surface, the body of the subject are divided into six parts in which are head, torso and for limbs. In this thesis, since all the input character are rigged character, the body part can be determined by the skin weight corresponding to the bones. After the segmentation of the body 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, this is demonstrated in Figure.4.17

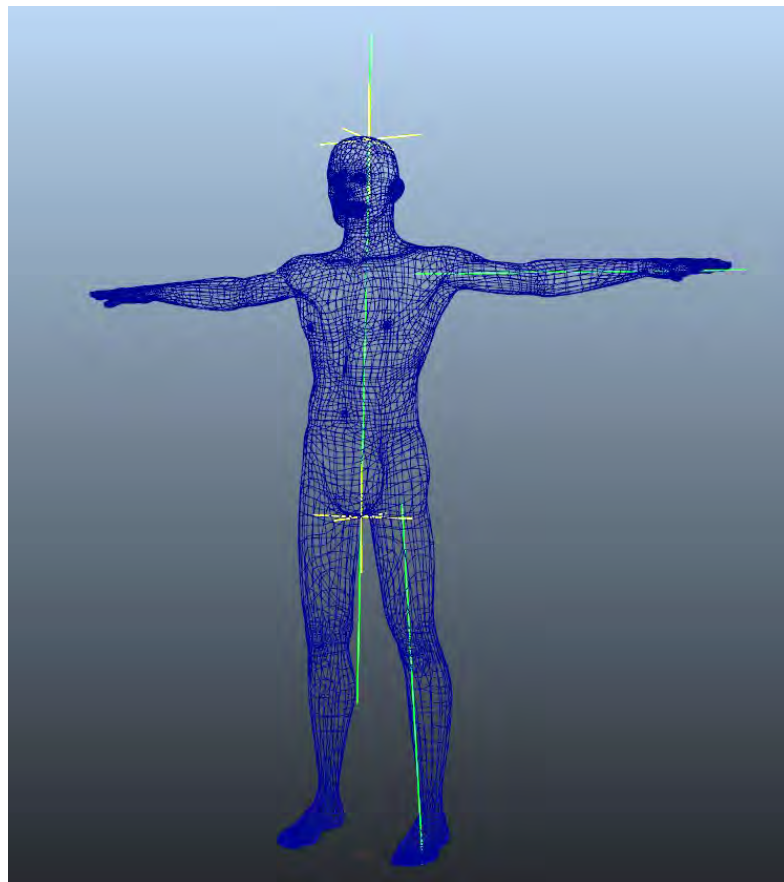


Figure 4.17: *Centre axis for the body part on the left side of the subject.*

The radius of the cylindrical bounding surface is determined by the

number of pattern associated with this body part. At this step, patterns are pattern are arranged on the flattened bounding surface side by side according to position of the sewing relationship as showed in Figure.4.18. The width on the U direction of all the pattern is the circumference of the bounding surface. Figure.4.19 demonstrates all the bounding surface for a subject.

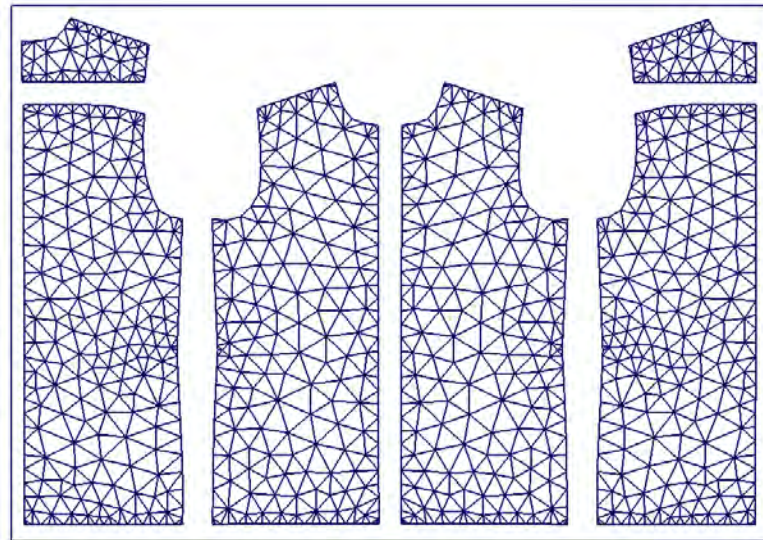


Figure 4.18: *Patterns arranged side by side within the range of a box, this box is the flattened bounding surface.*

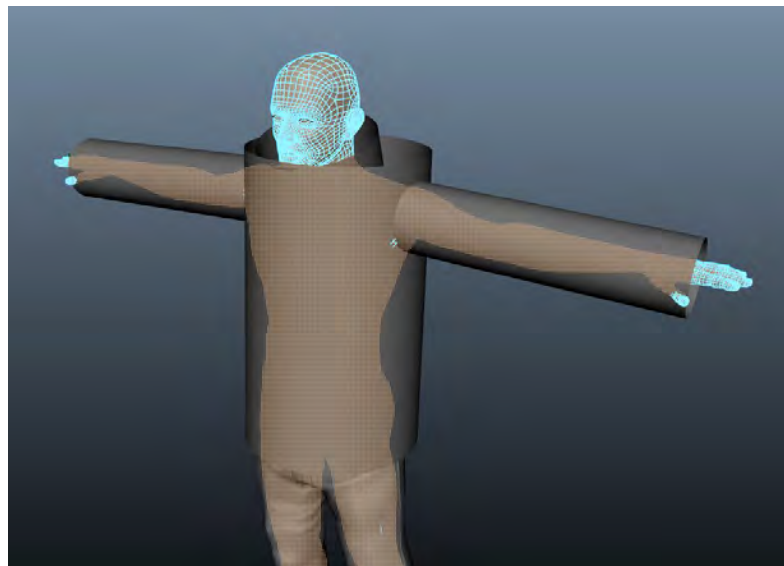


Figure 4.19: *Bounding surfaces on character.*

The next step is transfer the vertices of 2D pattern onto the bounding surface. At first, on the bounding surface, a area which has the same height and width of the bounding box of the patterns are marked out. Then, simply by enquiring the parametric coordinate of each point of pattern on the bounding surface, each point can be located at the bounding surface. After, by applying the same connectivity of the point from pattern to the points on bounding surface, the pattern is transferred from 2D space into 3D space. Figure.4.20 demonstrate the fully positioned shirt pattern to a subject.



Figure 4.20: *3D patterns*

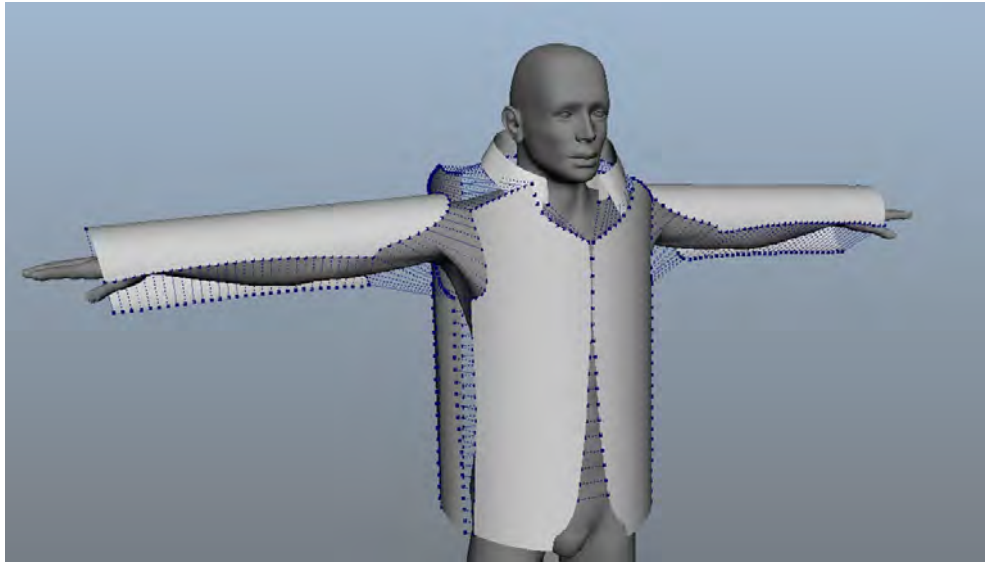


Figure 4.21: *3D pattern sewing*

To form the final cloth, constraints between sewing points need to be applied and physical simulation needs to be performed to pull all the patterns together. This assembling method is implemented in Maya2012 using the nCloth module. Figure 4.21 illustrates all the sewing constraints.

To validate the algorithm presented in this chapter, four characters are used for cloth modelling, see Figure 4.22. A shirt design (Xiong 2008) and a trousers design (Xiong 2008) are used for the cloth modelling test. This experiment is performed on an Intel Xeon 3.33GHz PC with 8GB RAM running Windows 7 (64-bit) operating system. For cloth resizing, 400 initial solutions are generated at the beginning of the algorithm and 200 generations are evolved for every character. Figure 4.24, 4.25, 4.26 and 4.27 demonstrate the final solution for each character respectively. After pattern resizing is complete, all the patterns are transferred onto the bounding surface of the character, this is shown in Figure 4.28. Figure 4.29 and 4.30 demonstrate the cloth modelled by the method presented in this chapter. Finally, Table 4.2 lists out the fitness value of the final solution of all characters. Notice that for character A and D, because the body proportion is largely different from the standard hu-

man body proportion that the shirt pattern and trousers pattern are based on. Therefore, cloth pattern requires larger deformation to fit the character A and D than character B and C. This is shown in Table.4.2 as the error in Shape function of character A and D is larger than the character B and C.

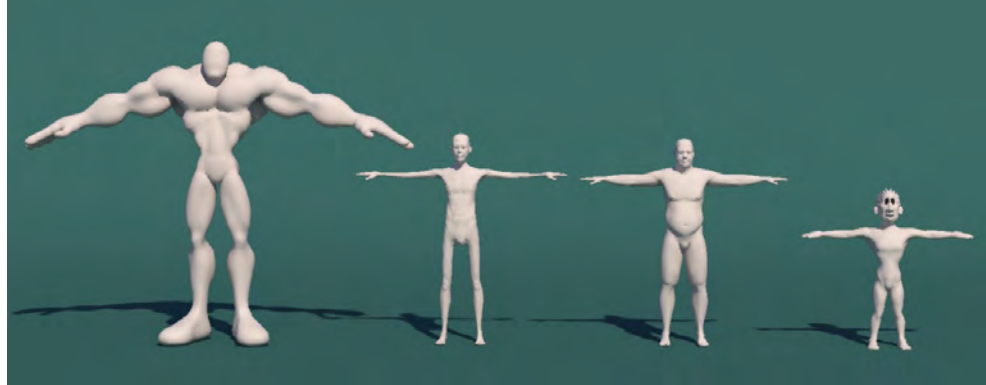


Figure 4.22: Characters need to be dressed up with the cloth design in Figure.4.23, from left, Character A, B, C and D.

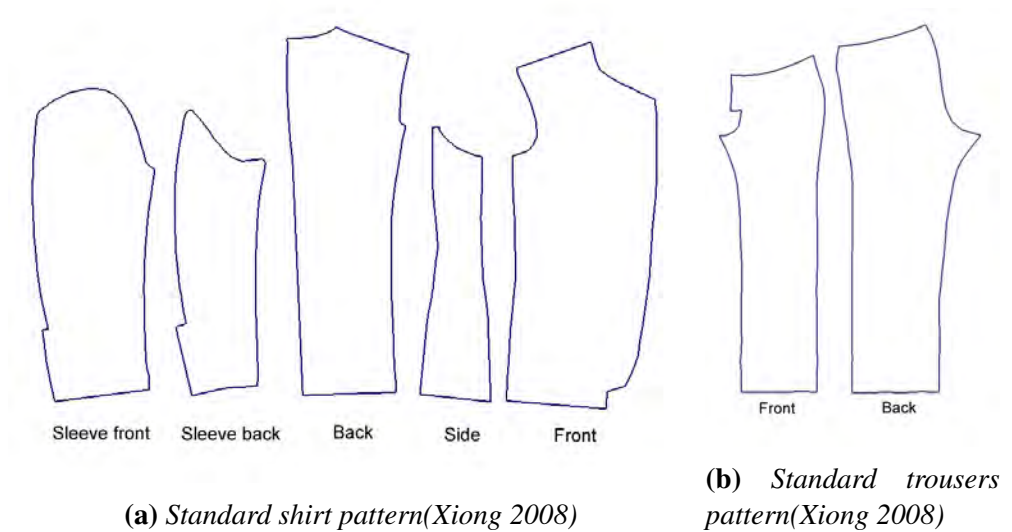


Figure 4.23: Pictures of animals

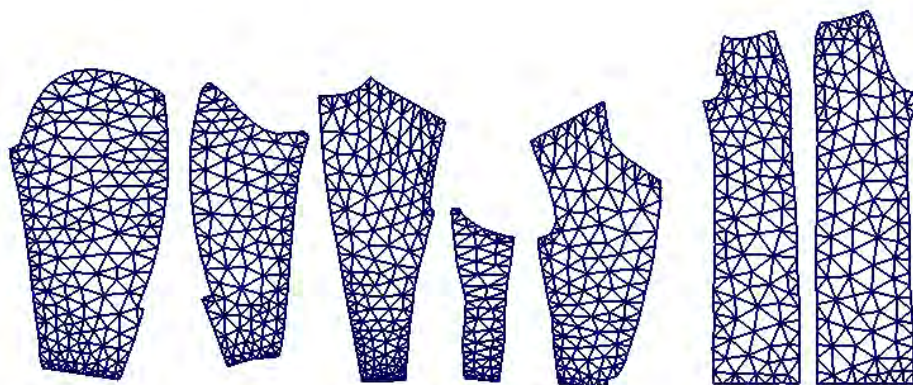


Figure 4.24: *Patterns for Characters A*

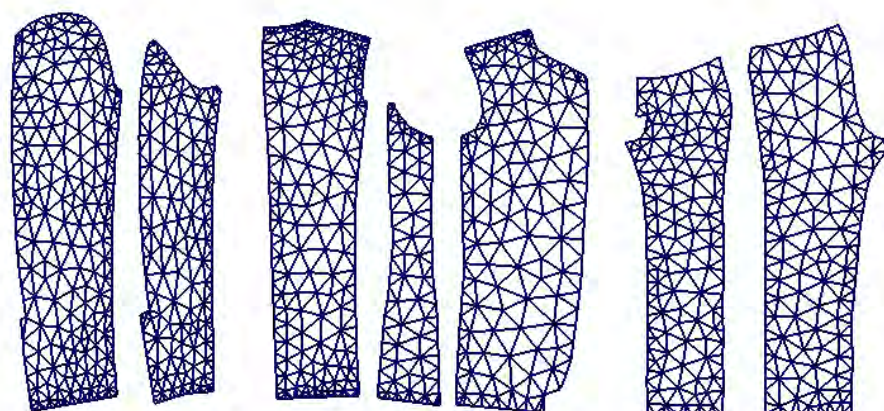


Figure 4.25: *Patterns for Characters B*

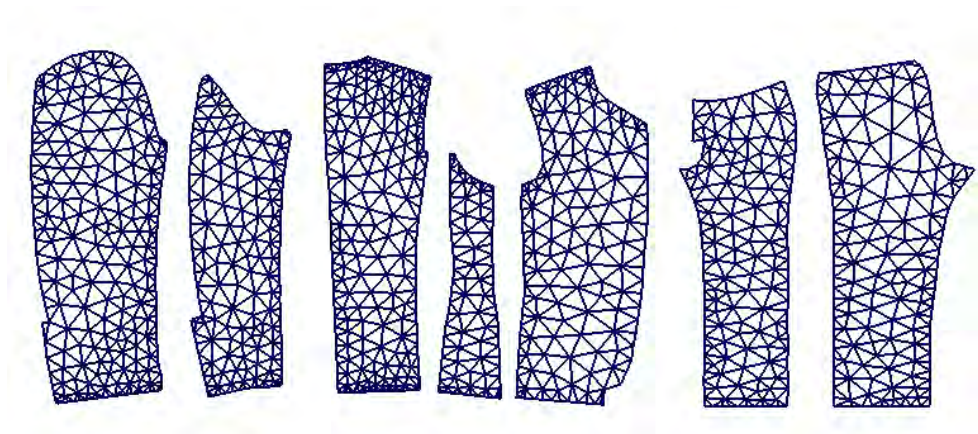


Figure 4.26: *Patterns for Characters C*



Figure 4.27: *Patterns for Characters D*



(a) *Character A*



(b) *Character B*



(c) *Character A*



(d) *Character B*

Figure 4.28: *3D Patterns on the bounding surface of characters*

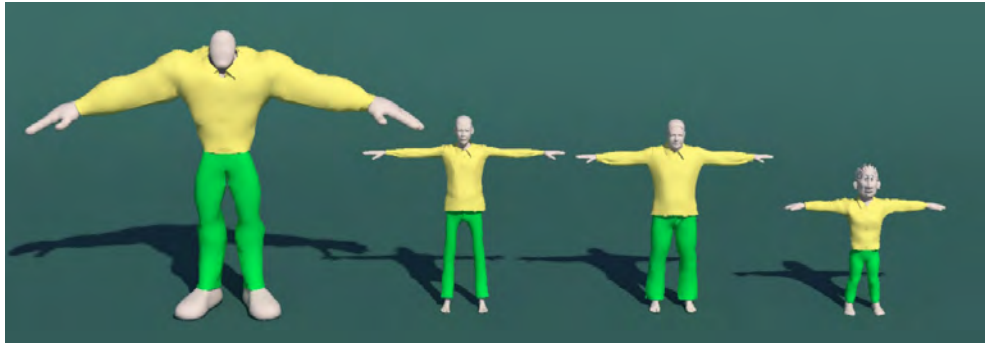


Figure 4.29: *Front view of the same cloth design being dressed on four characters with different body shape.*

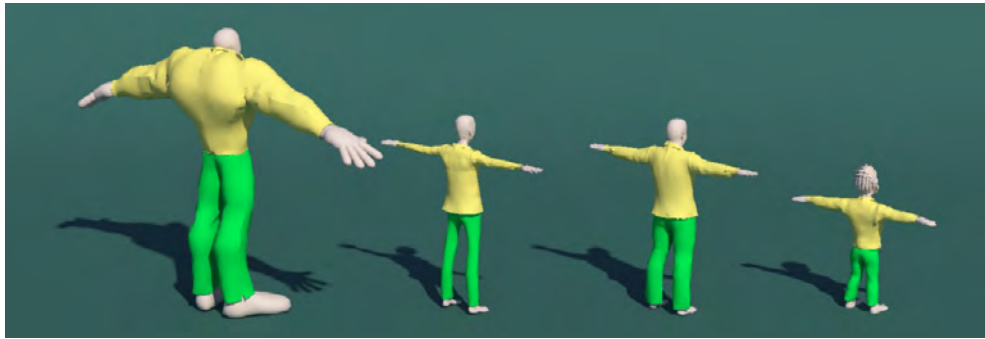


Figure 4.30: *Rear view of the same cloth design being dressed on four characters with different body shape.*

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 value 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.*

4.4 Conclusion

In this chapter, a method for dressing any character from a standard cloth design pattern is presented. This method combines the traditional clothes making techniques with modern computer graphic techniques to model the cloth from the measurements of the virtual character. This method takes 2D cloth patterns as inputs. And then key measurements are obtained from the body of the subject. After that the patterns are adjusted based on the key measurements to achieve fit to the body of the subject. Then all the patterns are transformed into mesh and positioned around the body of the subject. The vertex constraints are used to stitch all the patterns together based on the sewing information predefined by user in order to construct the final cloth. Last but not least, physical simulation is performed to the cloth to complete the pattern assembly process.

One of the most significant advantage of the method presented in this chapter is the ability to dress subject with any body proportion. This method strictly follows the procedure of make-to-measure tailoring to ensure the fit of the cloth. By automating the process of cloth pattern resizing, it is able to transfer a cloth from one subject to another subject with different body shape whilst maintain the design of the cloth. Moreover, because this method operates directly on the pattern rather 3D cloth, it ensures the accuracy of implementation of the cloth design when 3D cloth is created from patterns. In cloth manufacturing, all the cloth pattern are flat, in the other world, all the patterns are developable surface. The curve profile of a cloth is crated by using different sewing method.

However, many method that developed for dressing character are operator directly on 3D cloth object such as Brouet et al. (2012). Directly manipulating vertex on 3D cloth mesh to achieve fit is able to produce compelling result. However, the propose of garment pattern previewing is to create the

3D visual preview of 2D patterns. If a cloth is resized in 3D space, all the pattern need to be remapped from 3D space onto 2D planer in order to produce the pattern for resized cloth. However, the patten in 3D space is hardly maintain developable surface ones its vertex is modified. Therefore, the mapping operation from fitted 3D cloth object to 2D pattern can introduce error into the resized cloth pattern. In the method that presented in this chapter, all the resizing are done in 2D pattern before the sewing, therefore, the cloth that formed onto 3D subject is the “actual” preview of the pattern design. This feature avoid the introduction of errors when mapping an undevelopable mesh onto a 2D planner.

Chapter 5

Conclusion and Future Works

5.1 Conclusion

Recreate world surrounds us in an interactive virtual manner has been a hot research topic studied intensively throughout the history of computer science. In virtual world of today, research on virtual clothing is a crucial factor that directly affects the fidelity of virtual human. Virtual clothing requires two major tasks, cloth modelling and cloth simulation, as the area of application differs, the focus point of virtual clothing various. For computer aided cloth design in fashion industry and computer aided cloth cutting in garment manufacturing field, the accuracy of cloth modelling or cloth simulation are crucial however, in computer animation and computer game, the visual fidelity and simulation efficiency are the objectives.

The research proposed in this thesis focus at the virtual cloth modelling problem in computer animation area. Followed with high speed development of computer hardware and cheaper storage, more and more characters are needed in a virtual world. However, creating virtual cloth for a virtual character involves both textile engineering knowledge, artistic expertise as well as computer graphic knowledge, therefore, it is considered as a challenging and

time-consuming task. Especially when dressing multiple characters, the cloth modelling process needed to be repeated for every character. This repetitive still required when dressing multiple character with different body proportion with one cloth design.

Many techniques have been developed for this area, Chapter.2 briefly reviewed the current virtual clothing methods, however, the major problem of the current cloth modelling method is that, most of the method are not designed to model from fit, a cloth is treated as a part of virtual character and can only be wear on a particular character. One cloth cannot be putting on another character without involving large amount of manual adjustment to the cloth model. Therefore, in order to model cloth for multiple characters, repetitive manual operation can not be avoid.

The cloth modelling method presented in this thesis adapted pattern grading technique and made-to-measure tailoring technique from the garment manufacturing industry into computer graphic area. By using this well-developed techniques from garment manufacturing industry, the method presented in this thesis able to tailor made cloth for every character automatically regardless of their body proportion. Once the cloth is created by a set of patterns, the measurements are taken from the character and the cloth is resized based on the measurements automatically to ensure the cloth fits character, therefore largely improves the efficiency of dressing multiple characters. Because In the apparel industry there is a huge number of cloth designs that represented by patterns. However, in computer animation, an animator could not use those cloth patterns and resizing techniques directly to assist them to model the cloth for characters. In fact, as mentioned above, the major methods of cloth modelling in animation still require the animator to model every part of the cloth manually. The topology structure and modelling accuracy for cloth are seriously constrained by the workload of the animator. The research presented in this thesis is driven by such needs that is, to develop a automatic

cloth modelling method that is able to takes measurements from the body of the subject and adjust all patterns respectively in order to model the cloth for an individual character. By automating the process of pattern based cloth modelling, dressing multiple character no longer requires manual adjustment to every character. In fact, only the measurements need to be extracted from every character and the cloth modelled by the method presented in this thesis is able to generate cloth that fit to the character.

In order to accomplish such a task, two steps need to be involved, firstly, measuring subject and secondly adjusting cloth pattern. For the first step, in order to model cloth for any character from fit, the measurement of the body of the subject need to be extracted. In computer animation, the “T-Pose” is the standard posture for modelling character. Unfortunately, the concept of “T-pose” are ambiguous because different studios have different definition of the “T-Pose”. This problem is solved by using geodesic distance instead of Euclidean distance. Chapter.3 presented a method for calculating geodesic on polyhedron and point cloud data with high efficiency. For the second step, because a Cloth is consisted by a group patterns. Because the body proportion of the character in virtual world does not like human body in which the body proportion follows certain statistical regularity Pheasant & Haslegrave (2006); ISO/TR-10652 (1991); EN:13402 (2001) therefore, patterns that adjusted based on body part can be largely different. However, for the pattern assembling, the seam-lines between each pattern need to be consistent and the original design of the cloth need to be preserved after the pattern adjustment. In order to tackled this problem, a cloth pattern resizing genetic algorithm is presented in Chapter.3 that able to find the best combination of adjusted patterns that fit the character.

The main contributions of this thesis is,

1. **A novel geodesic algorithm scheme**

In chapter.3, three algorithm respectively for accurate and approximate

geodesic computation on triangulated manifolds as well as approximate geodesic computation on point cloud data set are presented. The most important contribution is that the proposed approximation algorithm can reach linear time complexity with an bounded error on triangulated manifolds. Numerical comparisons with existing algorithms (i.e. MMP, ICH_1 and ICH_2) have further demonstrated the advantages of our algorithms in terms of both speed and accuracy. By integrating Algorithm.3 into the measuring system, the time consumed on solve the geodesic path between multiple pair of source and destination has largely reduced, moreover, the accuracy of the solution is maintained. In the final chapter , more experiments will be presented to illustrate the advance of the algorithm presented in this chapter.

2. Cloth pattern grading

In chapter.4, a method for dressing any character from a standard cloth design pattern is presented. This method combines the traditional clothes making techniques with modern computer graphic techniques to model the cloth from the measurements of the virtual character. This method takes 2D cloth patterns as inputs. And then key measurements are obtained from the body of the subject. After that the patterns are adjusted based on the key measurements to achieve fit to the body of the subject. Then all the patterns are transformed into mesh and positioned around the body of the subject. The vertex constrains are used to stitch all the patterns together based on the sewing information predefined by user in order to construct the final cloth. Last but not least, physical simulation is performed to the cloth to complete the pattern assembly process. One of the most significant advantage of the method presented in this chapter is the ability to dress subject with any body proportion. This method strictly follows the procedure of make-to-measure tailoring to ensure the fit of the cloth. By automating the process of cloth pattern resizing, it is able to transfer a cloth from one subject to another sub-

ject with different body shape whilst maintain the design of the cloth. Moreover, because this method operates directly on the pattern rather than 3D cloth, it ensures the accuracy of implementation of the cloth design when 3D cloth is created from patterns. In cloth manufacturing, all the cloth patterns are flat, in the other world, all the patterns are developable surface. The curve profile of a cloth is created by using different sewing methods.

5.2 Future work

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

1. Cloth animation

The method presented in this thesis only applies to the cloth modelling, it cannot generate the motion of the cloth. The full cycle of virtual clothing involves the reproduction of the dynamic property of the cloth, therefore, the proposed method could be improved by adding the simulation when modelling is complete.

2. Weighted geodesic

The geodesic presented in this chapter cannot calculate the geodesic on an even weighted graph. Equation 3.15 involves an identical matrix. By adjusting this matrix, this geodesic scheme can be extended to calculate geodesic on a weighted graph. Another advantage for extending the geodesic scheme presented in this thesis to a weighted graph problem is that, when the measuring subject has a skeleton attached to it, the vertices on the mesh of the subject have different skin weights. Therefore, the geodesic could travel through the datum near joints, resulting in more accurate measurements when the posture of the subject is different.

3. Interactive cloth design

By providing measurement, the cloth modelling method presented in this thesis can generate fit cloth for any subject. However, sometimes, during the fashion design, for a particular kind of body proportion, the design of the cloth needs to be changed a bit for the best appealing. Therefore, the research in this thesis can be improved by adding a pattern editing interface to allow pattern to be adjusted by the user.

Bibliography

- 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. Arya, et al. (1998). ‘An optimal algorithm for approximate nearest neighbor searching fixed dimensions’. *J. ACM* **45**(6):891–923.
- 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.
- N. Badler (1997). ‘Virtual Humans for Animation, Ergonomics, and Simulation’. In *Proceedings of the 1997 IEEE Workshop on Motion of Non-Rigid and Articulated Objects (NAM ’97)*, NAM ’97, pp. 28–, Washington, DC, USA. IEEE Computer Society.
- 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.

- R. Brouet, et al. (2012). ‘Design Preserving Garment Transfer’. *ACM Transactions on Graphics* .
- 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.
- 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.
- M. Cavazza, et al. (1998). ‘Motion Control of Virtual Humans’. *IEEE Comput. Graph. Appl.* **18**(5):24–31.
- 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.
- B. P. Company (1994). *Butterick's 1892 metropolitan fashions*. Dover Publications.
- 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.
- 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.
- J. daniel Boissonnat & F. Cazals (2001). 'Coarse-to-fine surface simplification with geometric guarantees'. *Computer Graphics Forum* **20**:490–499.
- M. de Berg, et al. (2008). *Computational Geometry: Algorithms and Applications*. Springer.
- 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.
- 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. 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. Gorsline (1993). *A History of Fashion: A Visual Survey of Costume from Ancient Times*. B.T. Batsford.
- 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.
- J. Hershberger & S. Suri (1999). ‘An Optimal Algorithm for Euclidean Shortest Paths in the Plane’. *SIAM J. Comput.* **28**(6):2215–2256.
- 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.
- M. Hofer & H. Pottmann (2004). ‘Energy-minimizing splines in manifolds’. *ACM Trans. Graph.* **23**(3):284–293.
- J. H. Holland (1992). *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA.
- W. Hundsdorfer & J. Verwer (2003). *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer Series in Computational Mathematics. Springer.
- 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.

- J. M. Kaldor, et al. (2010). 'Efficient yarn-based cloth with adaptive contact linearization'. *ACM Trans. Graph.* **29**(4):105:1–105:10.
- T. Kanai & H. Suzuki (2000). 'Approximate Shortest Path on Polyhedral Surface Based on Selective Refinement of the Discrete Graph and Its Applications'. In *Proceedings of the Geometric Modeling and Processing 2000*, GMP '00, pp. 241–, Washington, DC, USA. IEEE Computer Society.
- S. Katz & A. Tal (2003). 'Hierarchical mesh decomposition using fuzzy clustering and cuts'. *ACM Trans. Graph.* **22**(3):954–961.
- 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.
- 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.
- N. M. MacDonald (2009). *Principles of Flat Pattern Design 4th Edition*. Bloomsbury Academic.
- A. Margolis (1964). *The complete book of tailoring*. Doubleday.
- B. F. McManus (2003). 'ROMAN CLOTHING: WOMEN'.
- S. Mehta (2009). *Human Body Measurements: Concepts And Applications*. Prentice-Hall Of India Pvt. Limited.
- F. Mémoli & G. Sapiro (2001). 'Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces: 730'. *J. Comput. Phys.* **173**(2):764–.

- 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.
- M. Muller, et al. (2004). ‘Physically-Based Simulation of Objects Represented by Surface Meshes’. In *Proceedings of the Computer Graphics International*, CGI ’04, pp. 26–33, Washington, DC, USA. IEEE Computer Society.
- J. Noke & R. M. I. of Technology. Patternmaking Dept (1987). *History of the Patternmaking Department, Royal Melbourne Institute of Technology 1909-1985*. R.M.I.T.
- K. Norton, et al. (1996). *Anthropometrica: 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.

- G. Peyré & L. D. Cohen (2006). ‘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.
- H. Pottmann, et al. (2010). ‘Geodesic patterns’. *ACM Trans. Graph.* **29**:43:1–43:10.
- 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.
- M. R. Ruggeri, et al. (2006). ‘Approximating geodesics on point set surfaces’. In *Proceedings of the 3rd Eurographics / IEEE VGTC conference on Point-Based Graphics*, SPBG’06, pp. 85–94, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

- 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.
- 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 thorie des courbes double courbure.'. *Journal de Mathematiques 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.
- M. Shoben & J. Ward (1987). *Pattern Cutting and Making Up*. Butterworth.
- 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 (1994). 'Adaptive probabilities of crossover and mutation in genetic algorithms'. *Systems, Man and Cybernetics, IEEE Transactions on* **24**(4):656–667.
- E. Staff (2007). *Hand Book of Garments Manufacturing Technology*. Engineers India Research Institute.
- P. Stecker (1996). *The Fashion Design Manual*. Macmillan Education Australia.
- 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.
- D. Thalmann (1992). 'Using Virtual Reality Techniques in the Animation Process'. In *Proc. Virtual Reality Systems, British Computer Society*.
- 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 Deform-

- mation 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. Tsopeles (1991). ‘Animating the Crumpling Behavior of Garments’. In *Proc. 2nd Eurographics Workshop on Animation and Simulation*, pp. 11–24.
- 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.
- H. Wang, et al. (2010). ‘Example-based wrinkle synthesis for clothing animation’. *ACM Trans. Graph.* **29**(4):107:1–107:8.
- J. Weil (1986). ‘The synthesis of cloth objects’. *SIGGRAPH Comput. Graph.* **20**(4):49–54.
- 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.

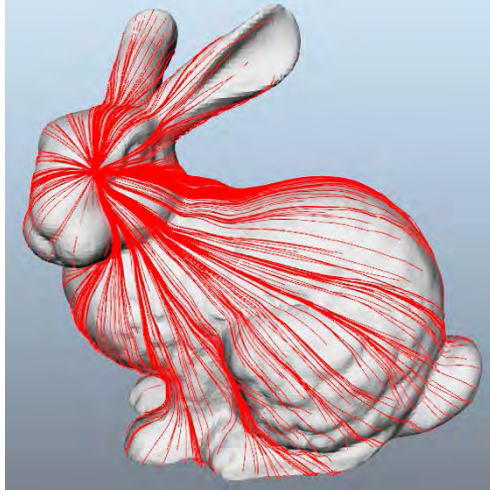
N. Xiong (2008). *World Classical fashion Design and Pattern*. Jiangxi Art Press.

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.

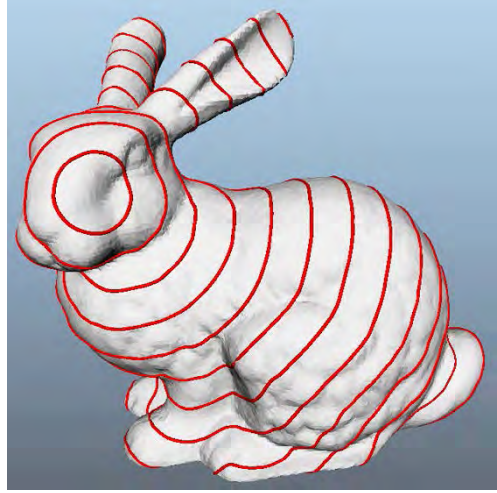
Appendix A

Appendix Title

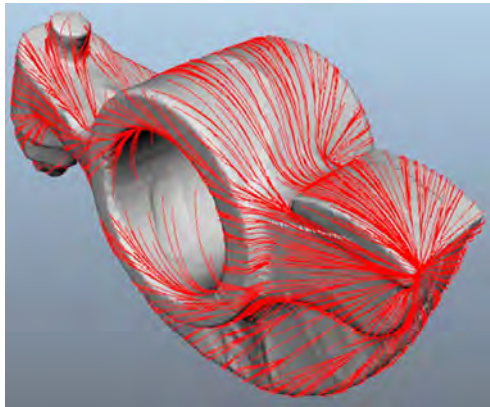
Experiment for comparing time complexity that performed on large models:



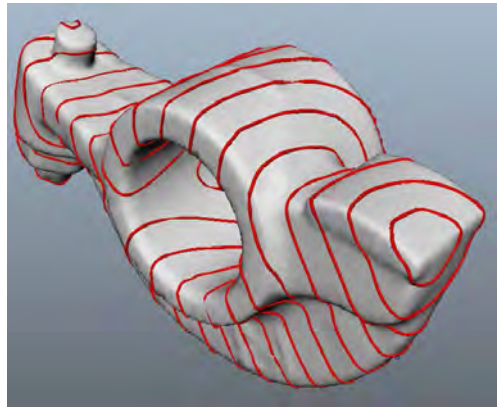
(a) *geodesics on bunny*



(b) *isoline on bunny*



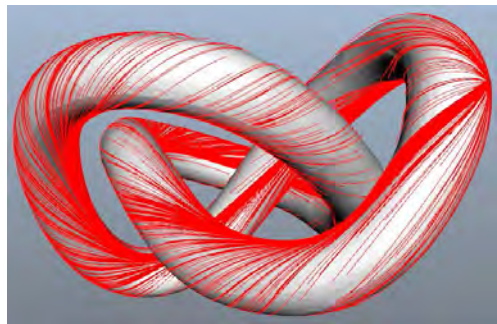
(c) *geodesics rockarm*



(d) *isoline on rockarm*

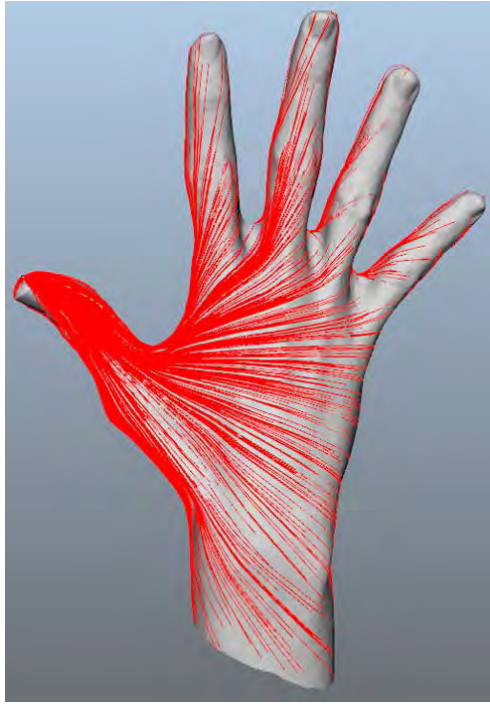


(e) *geodesics on asian dragon*

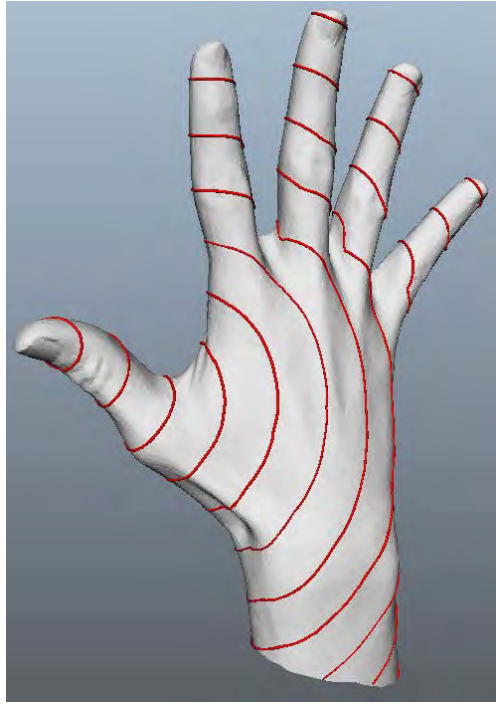


(f) *geodesics on knot*

Figure A.1: *Models used in geodesic computation experiments*



(a) *geodesics on hand*



(b) *isoline on hand*



(c) *geodesics on Neptune*



(d) *isoline on Neptune*

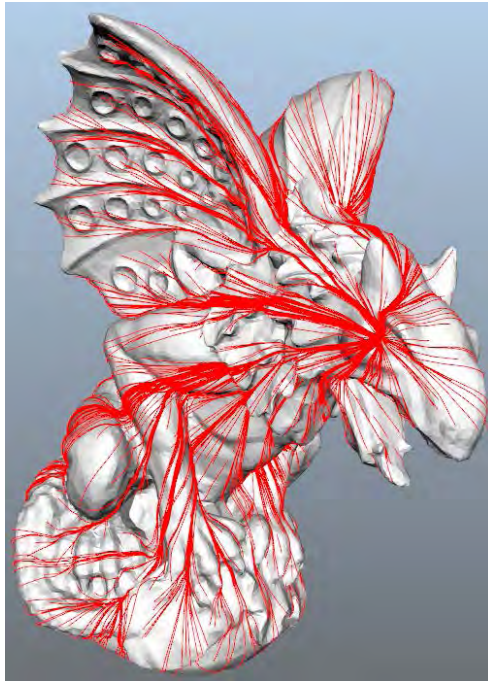
Figure A.2: *Models used in geodesic computation experiments.*



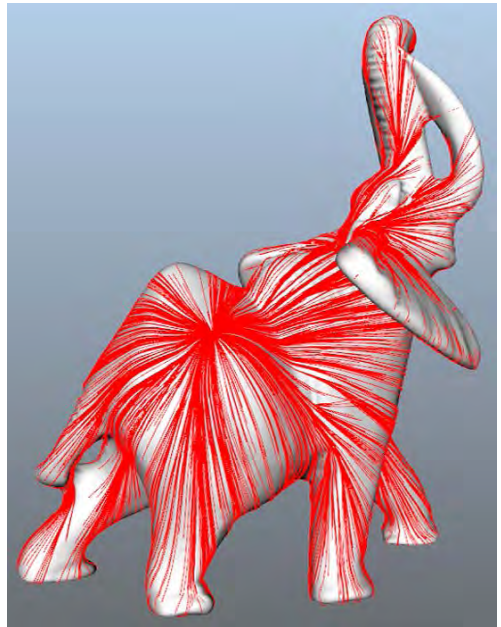
(a) *geodesics on Lucy*



(b) *isoline on Lucy*



(c) *geodesics on gargo*



(d) *isoline on elephant*

Figure A.3: *Models used in geodesic computation experiments.*

Times(sec)	Algorithm.3	MMP app	Improved CH.2
Hand #V:80089 #F:160000	6.627	18.762	13.39
Bunny #V:119602 #F:251704	14.922	19.018	20.161
Camel #V:207266 #F:414528	17.851	63.915	35.196
Rockerarm #V:241056 #F:482112	19.944	85.119	43.114
Jar #V:408378 #F:960516	56.861	195.09	93.068
Neptune #V:2003932 #F:4007872	315.358	970.927 (167.992)*	832.254 (689.307)*
Elephant #V:2396060 #F:4792128	357.435	1526.86 (194.28)*	1962.01 (1524.81)*
Dragon #V:3609455 #F:7218906	621.892	2013.92 (229.76)*	1281.363 (1008.221)*
Gargoyle #V:5391140 #F:10782288	1482.25	4649.39 (870.82)*	4386.76 (3182.91)*
Neptune #V:12023612 #F:24047232	2449.44	Out of mem- ory 24Gb	26822.91 (20175.29)*
Lucy #V:14027872 #F:28055742	3545.714	Non- manifold triangles	Non- manifold triangles

Table A.1: The comparison of running time of Algorithm.3, MMP app, and ICH.2.

**For several large models, the “backtracing” times is also showed for MMP approximation and improved ICH.2 algorithms in brackets respectively*

The experimental results are given in Table.A.1. Figure.A.1 A.2 and A.3 demonstrate the shortest paths or isolines on these models that are used

in Table.A.1. It can be noticed that the MMP approximation and ICH_2 algorithms can cost less time for small scale models. However, for large models, The approximate algorithm presented in this thesis is noticeably faster than others. Particularly, for large models such as Asian dragon in Figure.A.1, Lucy and Gargo in Figure.A.3 the approximate algorithm presented in this thesis is many times faster than the MMP approximation and ICH_2. This is consistent with Figure.3.12.

Moreover, It can be observed that the “backtracing” takes up too much time in MMP and ICH algorithms, in particular, the “backtracing” occupies most of the running times for the large models in ICH_2. This further verifies The approximate algorithm presented in this thesis is appropriate for tracing accurate geodesic paths on large models.