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

December, 2013

Abstract

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

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

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

There are two main contributions in this research. Firstly, a geodesic curvature flow based geodesic computation scheme is presented which mimics the tape measuring process in tailoring. This scheme is capable of calculating geodesics on the character models in the form of both polyhedron and point cloud. In computer animation, the number of the virtual characters that can be handled simultaneously is increasing rapidly. In order to reduce the cost for modelling cloth for multiple different characters, a fast geodesic algorithm that has linear time complexity with a small bounded error is also presented. Secondly, a cloth pattern adjusting genetic algorithm is developed for automatic cloth fitting. This thesis considers the cloth fitting process as an optimization procedure. It optimizes both the shape and size of each cloth pattern automatically, the integrity, design and fit of the cloth are evaluated in order to generate a fit cloth while preserve the original cloth design. This is the first attempt to utilize the genetic algorithm in 2D cloth pattern adjusting problem for dressing different characters.

By automating the cloth modelling process, the tedious modelling work required by current cloth modelling method will be avoided. It empowers the creativity of animation artists and improves their productivity by allowing them to use the large amount of existing cloth design patterns in the fashion industry to create various clothes that fit multiple characters with different body shapes and proportions.

Contents

List of Figures

List of Tables

Abbreviations

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

List of Symbols

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

$\vec{n_p}$	Principle normal vector at a point on curve C
$\vec{n_b}$	Binormal vector at a point on curve C
$\overrightarrow{p_i p_{i+1}}$	A vector from p_i to p_{i+1}
$\langle A, B \rangle$	Dot product of vector A and B
$\times$	Cross product of two vectors
$p_{i-1}\widetilde{p_i p_{i+1}}$	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 Tp
fp_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

Clothing technique is essential to people's daily life. It has been developed for more than hundreds years. In order to produce a cloth, normally there are five steps involved in this process(Margolis 1964), choose the style, measure the body of the customer, adjust the cloth patterns, assemble patterns, and final try-on. For bespoke clothing, before final assembling, patterns will be loosely stitched together and put onto the customer to further trim off the excess to achieve better fit. In 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 easier to be accessed by the rapidly expanding lower middle class. This innovation also provides a standard for fashion designs to be preserved and spread with much lower cost, moreover, a design can be adapted to different human figures with ease.

Cloth patterns consist of a set of textile pieces which are the breakdowns of a complete garment. It is the most important medium in today's fashion industry. Almost all the cloth designs are preserved or distributed in the form of cloth pattern. Cloth pattern also provides an intuitive approach for user to recreate the original design of the cloth. By altering certain part of

a cloth pattern, the cloth can be adjusted into any desired size for any wearer.

Followed with the fast development of computer hardware and computer graphic techniques, simulating clothes and clothing in computer simulated virtual environment has become a hot research topic for decades. By constructing the cloth object in a computer simulated virtual environment, it can reproduces both visual features and physical behaviours of the textile objects. Virtual clothing is the generic term of this process(Volino & Thalmann 2000).

In general, there are two areas utilise the virtual clothing techniques, today's fashion industry and film or game industry.

1.1 Virtual clothing in fashion industry

In the fashion industry, cloth pattern is the basic item that consists a functional cloth. In order to implement a cloth design, firstly, the design needs to be broken down into a set of textile pieces, then by applying the assembling techniques such as stitch or adhesive, all the pieces are assembled together in certain sequence to form a complete cloth. The cloth design pattern can only be visualised after the completion of the entire cloth production process. Virtual clothing techniques in the fashion industry mainly focused on operating the cloth pattern in order to improve the efficiency of the cloth designing. By following the procedure of the real cloth production, the cloth patterns can be created, edited, graded and pre-visualised in the computer simulated virtual environment without actually implementing the cloth design into a real cloth. Furthermore, the physical simulation is also widely used for analysing the mechanical properties of the cloth.

1.2 Virtual clothing in computer animation

Virtual clothing in computer animation mainly focuses on reproducing the appearance of the cloth. It reproduces both the static shape and the dynamic behaviour of the cloth for the virtual character. In general, there are two steps involved in this process, cloth modelling and cloth simulation

1.2.1 Cloth modelling

Cloth modelling is the process of constructing the geometric shape of the cloth in the virtual environment. Due to the vast variety of the character designs, the intuitiveness and controllability of the cloth modelling method is crucial to the animation artist. The general-purpose geometric modelling method is the most popular method used for cloth modelling in computer animation. This type of method operates directly on the basic element of the geometrical representation of the 3D object, therefore, it gives the user maximum controllability throughout the modelling process and the appearance of the 3D cloth can be viewed throughout the entire modelling process. Pattern based cloth modelling method is rarely used in this area, because producing cloth pattern for a cloth design requires lots of expertises in tailoring but few animation artists possess such a skill. Moreover, because cloth model can only be visualised after completing the entire process of the pattern based cloth modelling, this makes editing a cloth notoriously tedious due to the repetition of the modelling process for modifying the cloth patterns.

1.2.2 Cloth simulation

Cloth simulation is the process of reproducing the dynamic behaviour of the cloth object in a virtual environment. Based on the method used for driving

the deformation of the cloth, in general, there are two types of cloth simulation methods used in the computer animation.

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

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

1.3 Motivation

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

The latest advances of computer hardware resulting an increase of the

computational power. The physical simulation of the cloth became possible. Although the cloth can be modelled and simulated separately from character skin, the current methods for constructing cloth for a virtual character still requires large amount of manual operation and computation power. The modelled and simulated cloth can only be applied to a particular character. In order to model clothes for different characters, a substantial amount of work need to be carried out. Animation artist usually needs to start from scratch to model a cloth for a new character. The duplication of effort cannot be eliminated when dressing different characters with different body shapes and proportions.

Pattern based cloth modelling techniques are widely used in the fashion industry. In film and gaming industry, most artists still rely on the general-purpose modelling software packages such as Maya, Softimage, 3D Studio MAX to construct cloth for their characters, the pattern based cloth modelling methods are rarely used due to the unintuitiveness and tediousness of its workflow.

In general, the workflow of the pattern based cloth modelling technique consists of three steps, 2D pattern design, 3D pattern generation and 3D pattern assembling. The first step correlates to the patternmaking process in the real world cloth production. The patterns that are created in this step determines the final shape of the cloth. Patternmaking in fashion industry is a highly skilled task, that only the well trained experts are able to produce the correct patterns for a cloth design. However, few animation artists possess this skill. In fact, in most cases, animation artists rely on the existing cloth pattern in fashion industry as the reference for the modelling cloth. However, in the fashion industry, cloth patterns are designed based on the measurements of either the ideal figure or a particular customer in order to produce the cloth that fits them. For the vast variety of the body shape and proportion of the animation characters, the pattern comes from the fashion industry does not fit

to those characters. To make matters worse, animation character usually has a very exaggerated body proportion which is far away from the normal human body proportion. The traditional pattern grading technique that is widely used in fashion industry can not be applied here. Moreover, without the knowledge of the pattern adjustment that is used in fashion industry, it is very difficult for an artist to adjust each pattern to fit the cloth to a new character. In order to adjust the design of the cloth, artists must either start from 2D pattern generation and repeat the rest of the workflow in which usually requires several iterations to construct the desired cloth, or perform the general-purpose modelling techniques to amend the ill-fitted cloth. When patterns are adjusted manually by artists, the lack of patternmaking knowledge makes preserving the original cloth design difficult, the appearance of the cloth largely depends on the modelling skill of the artists.

There are two key factors that are directly related to the appearance of the modelled cloth, the preservation of the cloth design and the fit of the cloth. Several techniques have been developed for transferring a cloth from one character to a different character. Cloth geometric morphing techniques is one of the most popular method(SmithMicro 2012). However, this type of methods operates directly on the geometry of the 3D cloth, it can fit cloth to different characters, but the design of the cloth can be largely different from its original when characters are largely different. Furthermore, Brouet et al. (2012) have proposed series of methods that directly operate the geometry of the 3D cloth on the new character in order to fit the cloth while maintaining the cloth design. However, the final 2D cloth design patterns need to be extracted by using surface flattening techniques(Sheffer et al. 2005). It not only requires an excess computation, but also the shape of the pattern can not be preserved from the surface flattening process.

With the fast development of computer power, more and more characters can be handled simultaneously. Multiple character application such as

crowd system is widely used in the production of film, TV and games. The shortcomings of the current cloth modelling methods such as tediousness and unintuitiveness, have become the bottleneck of the pipeline in these field. The research presented in this thesis aims at providing an efficient and easy-to-use solution for virtual clothing. With the help of this solution, not only the animation artists are able to model cloth for different characters automatically, but also the large amount of cloth designs exist in the fashion industry will become usable for the construction of the virtual cloth.

1.4 Research aims and objective

Although the geometric cloth modelling method provides lots of controllability to the user and it is very easy to use, the diversity of the cloth is largely limited by the artistic skill of the animation artist. The current pattern based cloth modelling method requires deep knowledge in tailoring and lots of repetition for editing the cloth. However, its ability of using the large amount of existing cloth design patterns in the fashion industry is still a very attractive advantage to the animation artist.

The research presented in this thesis focuses at bridging this gap by improving the tediousness and unintuitiveness of the current pattern based cloth modelling method. This work aims at developing an easy-to-use automatic pattern based cloth modelling method for the animation artist that allows them to directly use the existing cloth design patterns in the fashion industry.

This method includes several techniques: automatic character measuring, automatic cloth pattern adjusting and cloth pattern assembling. Given a character model and patterns of a cloth design from fashion industry, this method is able to fit the cloth to any character while maintaining the original cloth design. The objective of this thesis are to:

1. Review recent work on different techniques for virtual clothing and analyse their advantages and disadvantages for modelling cloth for characters with different body shapes and proportion.

2. Develop an efficient measuring method for the character regardless of their posture.

3. Develop a measuring method for the point cloud character model to cope with the increasing number of 3D scanned character models.

4. Develop a method to adjust each cloth pattern automatically based on the measurements and preserve the original design of the cloth.

1.5 Contributions

The virtual clothing method presented in this thesis consists two major parts, character measuring and cloth pattern adjustment. For character measuring, a geodesic based measuring method has been developed. Due to the fast development of 3D shape acquisition techniques, the character model is not only limited to polygon or smooth surface, but also 3D scanned point cloud data. Three geodesic algorithms have been developed for this part in order to cope with different geometrical representations of the character model. For cloth pattern resizing, a pattern adjustment genetic algorithm have been developed. The contributions of this thesis are as follows

1. A detailed review of work covering traditional cloth making techniques, recent research progress on computer-aided cloth design, and existing cloth modelling and simulation techniques.

2. This thesis describes an automatic pattern based cloth modelling method which bridges the gap between traditional tailoring techniques and virtual cloth modelling method. This enables the usage of the large amount of existing cloth design in fashion industry to animation artists.

3. This thesis also describes a geodesic curvature flow based geodesic scheme for the measurements extraction. This scheme consists of two algorithms, one with high measuring accuracy and the other incorporates a small bounded error into the geodesic calculation to achieve faster measurement with linear time complexity.

4. A point cloud geodesic algorithm is introduced in order to cope with the increasing number of the 3D scanned character models. This algorithm allows the geodesic to be calculated directly on the scattered point data outputs by 3D scanner.

5. A pattern adjusting genetic algorithm is proposed. Considering the measurement, seam-line among the patterns as well as the shape of each pattern, the original design of the cloth can be preserved through out the fitting process. This is the first attempt to utilize the genetic algorithm in 2D cloth pattern adjusting problem for dressing different characters.

1.6 Thesis outline

The reminder of the thesis is organised as follow:

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

Chapter 3 introduces a geodesic algorithm that inspired by the measuring method used in tailoring. Three geodesic algorithm that suit for dif-

ferent error bounds and geometrical representations are described in detail. Finally, the experiments that performed on many different model are demonstrated. The comparison between the presented algorithms and two most popular geodesic algorithms are presented.

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

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

Chapter 2

Literature Review

Clothing is one of the most distinctive features of human being that differs us from other creatures. 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 the cloth remains, that is to provide protection to the wearer from environment. Cloth also performs a wide range of cultural and social functions which expresses the personality, occupation, sexual differentiation and social status of the wearer (Harms 1938).

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

The cloth making techniques have been developed for hundreds years, a well developed cloth production pipeline is widely used in today's cloth manufacturing. This thesis adapts the cloth production techniques in fashion industry into the cloth modelling process in computer animation to improve the efficiency and reduce the tediousness of current cloth modelling techniques. Cloth pattern is the most common cloth design representation in the fashion industry. In order to model cloth based on cloth pattern efficiently, a general overview of the cloth pattern in fashion industry is presented in this chapter. Because the measurements acquisition for cloth making is crucial to the fit of the cloth, therefore, a detailed introduction of the research achievements in anthropometry study is presented. After that, the state of the art virtual clothing techniques are introduced. This thesis utilises geodesic to mimic the tape measuring in fashion industry for the measurements extraction from the character model and uses genetic algorithm to adjust cloth patterns automatically, therefore, finally, a brief review on geodesic algorithm and genetic algorithm is presented.

2.1 Pattern-making in fashion design

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

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

fabric pieces in order to minimize waste (Vout 1996).

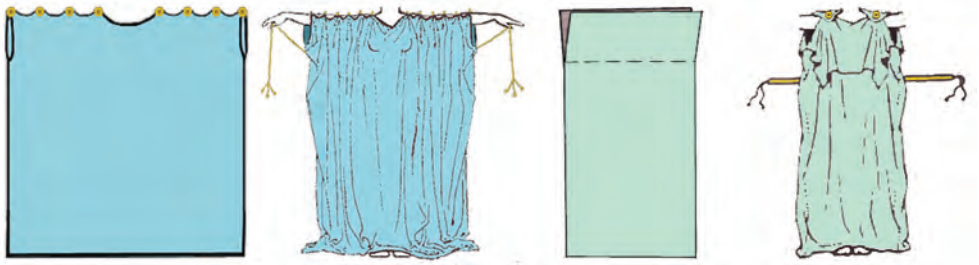


Figure 2.1: *The peplos(left) and the chiton(right) was the common cloth wear by woman in ancient Roman (McManus 2003).*

In the 15th century, the seminal art of patternmaking began. The fabric was carefully trimmed to fit the contour of the body(MacDonald 2009). The foundation of the 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. Cloth made by tailors was elaborate and relegated only to the very rich. With the onset of the industrial revolution, standardization of the cloth pattern has become essential to the success of the ready-to-wear cloth. Initial attempts to create standardized cloth pattern resulted in a poorly fitted garments with little detail. Men's suits were boxy, plain, ill-fitting sacks. After lengthy experimentation and standardization of sizing regulation, patternmaking made a triumphant transformation from customization to standardization (MacDonald 2009). During that time, cloth patterns are made into one size, the tailor had to grade the cloth pattern to the size that wearer was needed. Because pattern-grading requires lots of tailoring expertise and experiences, cloth pattern was not widely accepted in home sewing and the fashion was a phenomenon exclusive to ladies of high standing. At the end of 19th century, Butterick introduced the revolutionary graded cloth patterns(Company 1994). The effects of this idea was significant as it opens up the market of home sewing to the general public. With the advent of Butt-

erick's pattern, not only did dressmaking become much easier, but also did the fashion became available to men, women, and children of all classes all over the world. During the WWII, with the help of the development of anthropometry, the graded pattern could fit the general public better than before. After nearly a century, patternmaking has been widely accepted by today's fashion industry(Noke & of Technology. Patternmaking Dept 1987; Rosen 2004).

Today's patternmaking techniques mainly consist of two general categorises, patternmaking for massive production(Nugent 2008; Shoben & Ward 1987; Stecker 1996; Staff 2007), and patternmaking for custom tailoring(Margolis 1964; Cabrera & Meyers 1983; Browne 2011). The patternmeking in massive production mainly focus at the ease of distribution and standardization. This type of patternmaking techniques grade patterns into different predetermined size in order to fit most of the body size in general public. In custom tailoring, patterns are distributed as an one size design concept, the final cloth requires experienced tailor to adjust the size and shape of the cloth patterns based on the measurement taken from the customer to create fit cloth.

The research presented in this thesis brings the idea of custom tailoring into the construction of virtual cloth. Cloth are created from cloth patterns that are adjusted based on the measurements of the character. This approach allows a cloth design to be fitted onto any character.

2.2 Anthropometry

Fit is one of the essential factors that directly determines the functionality of a cloth. In order to achieve fit, measurements of the wearer's body need to be acquired. Anthropometry is the branch of human science that study the measurements of the body size, shape, mobility, flexibility and strength(Gupta & Zakaria 2014). Human body dimension, as ours personality, is largely varied among the population. Many user-centred applications require understanding

of this variability. Especially for garment industry, as cloth is an object that its functionality is determined by its coverage and sealability. Both the coverage and the sealability need to be ensured by obtaining wearer's body measurements. This section will debrief the research achievements in anthropometry and introduces the method of anthropometry data acquisitions.

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

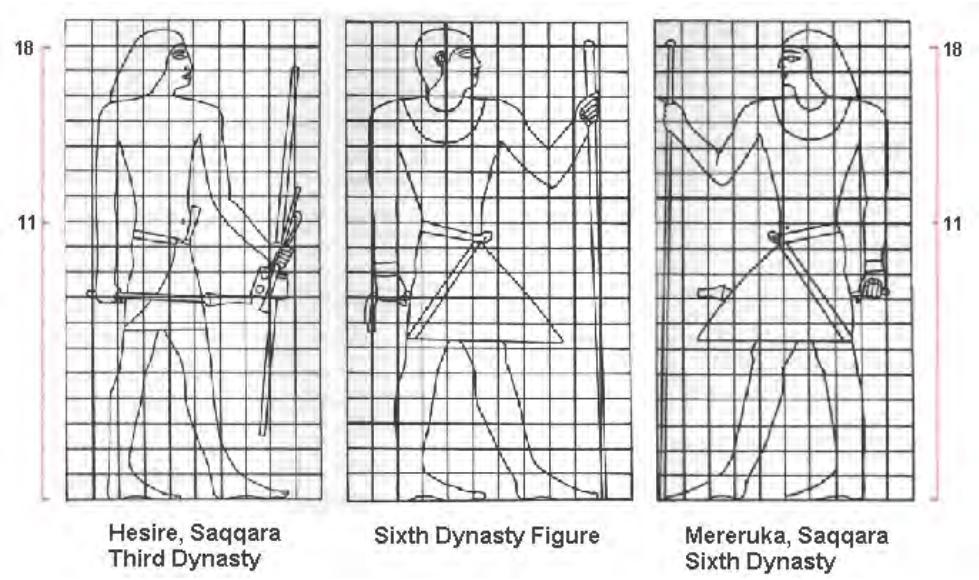


Figure 2.2: *Body proportion in ancient Egypt*

This modular system evolved initially as a drawing standard is still used

today. The most detailed system of human proportions that today's anthropometry researches built on is from the Roman architectural theorist Vitruvius in 15 B.C(Selin 2008). His theory of human proportions is well known as the "well-shaped man" (Arnheim 1955), in which the height of the human stature are equal to arm span. Vitruvius also employs this human proportion as a fundamental principle in his building design as he claims that "no Temple can have a rational composition without symmetry and proportion, that is, if it has not an exact calculation of members like a well-shaped man" (Pollio et al. 1914; Frings 2002).

The most recognized visualization of Vitruvius's human proportion is the drawing created by Leonardo da Vinci(Stemp 2006). This piece is accompanied by the nodes that are based on the theory of human body proportion developed by Vitruvius. It depicts a male figure in two superimposed postures that the arms and legs of the male are circumscribed by a cycle and a square respectively. After this piece was created, the theory of human proportion had become bound up with the "Golden ratio". "Golden ratio" refers to the umbilicus divides of the stature of 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 the 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

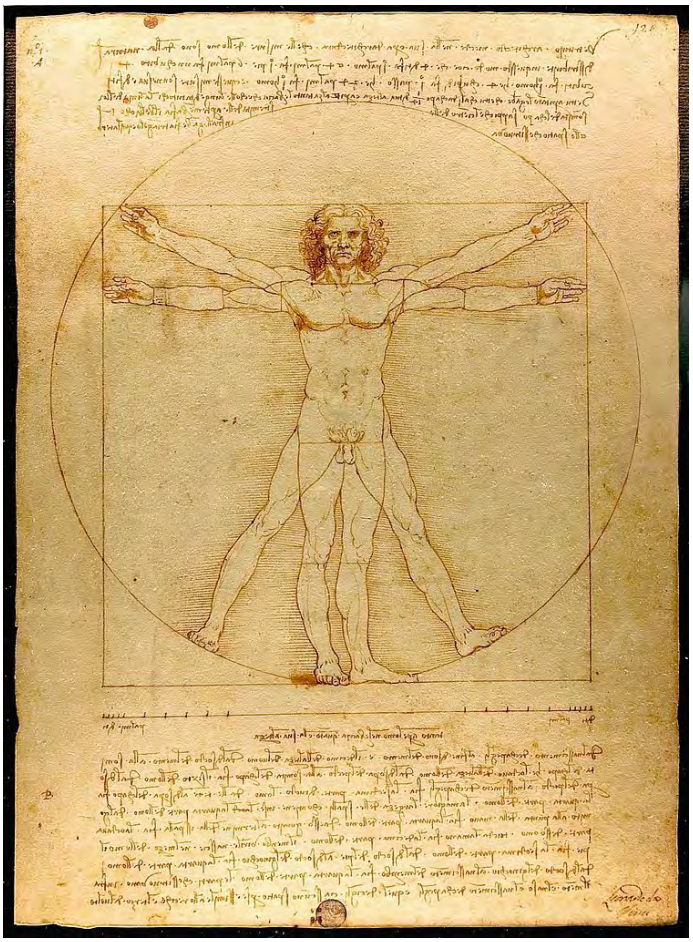


Figure 2.3: *The Vitruvian Man created by Leonardo da Vinci in 1490 (Stemp 2006)*

cockpits.

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

Driven by the industrialization of the early 19th century, the development of standardization reached to an unpretentiously speed. In order to massively 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 predetermined sizes. Grading

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*

is a standard method of applying increases and decreases to cloth patterns to make the cloth larger or smaller.

Generally, there are three steps involved in defining a sizing system (Schofield & LaBat 2005): Firstly divide the general population into several categories 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 fashion industry is to produce cloth that fits customer properly. Therefore, for massive cloth production, anthropometry is crucial to the design of the sizing system(Norton et al. 1996; Samaras et al. 2007; Mehta 2009).

2.3 Virtual clothing

Virtual clothing is a research topic in the field of computer graphics for a very long time. It is the generic terms of simulating clothes and clothing in computer simulated virtual environment. It reproduces both visual features and physical behaviours of textile objects in computer simulated virtual reality (Volino & Thalmann 2000). In general, today's virtual clothing methods

can be classified into three types based on the core technique that used for the construction of cloth shape or its deformation. They are geometric virtual clothing, physical virtual clothing and hybrid virtual clothing.

2.3.1 Geometrical Based Virtual Clothing

Geometrical virtual clothing technique 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 that utilises geometrical modelling techniques. The cloth was constructed by a grid consists of vertices, then the shape of cloth was generated from catenary curves between its hanging points, see Figure 2.4. This technique creates an underlying shape out of several hanging points. Then it passes through each set of these points and maps a catenary curve to the set. Finally it takes the lowest out of each overlapping set and uses it for the render. By using this method, the stationary hanging cloth can be generated efficiently, however, limited by the shape of the catenary curve, this method can only generates the hanging cloth, it cannot generates more complex shape of the cloth.

T. Agui & Nakajima (1990) introduced a geometrical method for modelling a sleeve on a bending arm. The sleeve 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 of bending sleeve and can only be implemented in the stationary cloth simulation.

Hinds & McCartney (1990) presented a method for interactive garments designing based on mannequins represented by bicubic B-Spline surface. The garment is represented by a group of 3D panels defined by its edges, and these are created as a surface offset from the body with various distances. later on, Hinds et al. (1991) presented a method of translate 3D pattern into 2D patterns

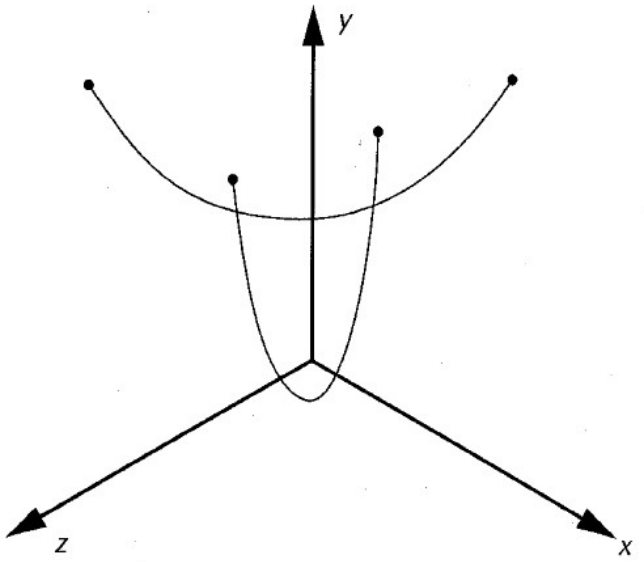


Figure 2.4: *Catenary curves presented in (Weil 1986)*

by using the method presented by Calladine (1986).

Miller et al. (1991) introduced a geometrically deformed model. This model 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 such as a sphere or a cube. Then based on a set of constraints, this simple object is expanded to fit the object within a volume. At the same time, the deformed object maintains its closed and non-self-intersect property. The major advantage of this method is that the sampled data are aggregated by placing geometrical relationship on the model. This allows a initial convex model to be transformed into a concave object. Since the computational cost is associated with the number of the constrain that is required during the deformation, the level of detail can be easily controlled. Later on, (Thomas Stumpp 2008) extended this method into cloth modelling. This method advanced itself with a higher efficiency. According to the experiment section in this paper, their method reached linear

time complexity in terms of number of the mesh vertices. However, since physical property of the cloth is related to the size of the clusters, the physical property of the cloth is determined by the topology of the cloth mesh and it cannot be controlled explicitly by the user.

Geometrical cloth modelling method may be fast on generating the static cloth shape, but cloth is a soft object whose shape is closely associated with the shape of body underneath. The physical simulation can generates the shape of the cloth easily and reproduce the kinetics property of textile very well.

2.3.2 Physical Based Virtual Clothing

For physical virtual clothing methods, cloth is represented by a set of particles that are inter-connected to each other by springs. This representation replicates the behaviour of the soft flexible object such as fabric. In general, there are two types of models for the physical cloth modelling technique, energy-based techniques(Terzopoulos et al. 1987) and the force-based techniques(Volino & Magnenat-Thalmann 2005). 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 particles in order to achieve a minimum energy state. This type 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. For this reason, the Force-based technique has been brought in to virtual clothing in order to describe the reaction of the cloth to the external forces. This type of techniques usually use differential equations to represent the force among each particle and perform a numerical integration to solve the differential equation and locates the position of each particle at every time step.

This type of methods is normally used in dynamic cloth simulation.

Terzopoulos et al. (1987) introduced a method utilises the physically-based model to construct the shape of a cloth object. Their method is based on the elasticity theory to describe the shape and motion of the deformable materials. It also has the ability to interact with other physically-based models. The simulation model introduced in this paper is based on the simplifications of elasticity theory to deformable curves, surfaces, and solid objects. It has the ability to generate static shapes by simulating their physical properties such as tension and rigidity (Figure 2.5). Moreover, by bring the physical properties such as mass and damping in to the physical simulation, the dynamic of objects can be simulated. *"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, it is very time consuming for solving the energy equation for a complex object, limited by the computational power, this method can only simulates the interaction between simple shaped objects.

Volino & Magnenat-Thalmann (2005) introduced a general mechanical model for cloth simulation. Because the mass-spring system cannot represents the anisotropic nonlinear mechanical behaviour which often appears on the fabric, This model utilises an accurate particle system for dynamic simulation instead of mass-spring system. A triangle face of the cloth mesh is described by three 2D coordinates correspond to three mechanical properties, weft, warp elongation and shear. These 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, see Figure 2.6. This particle system is able to simulate the anisotropic nonlinear mechanical behaviour of a cloth object by using polynomial spline approximations of the strain-stress curves. Furthermore, the aforementioned three mechanical properties are physically measured from real cloth piece using Kawabata Evaluation

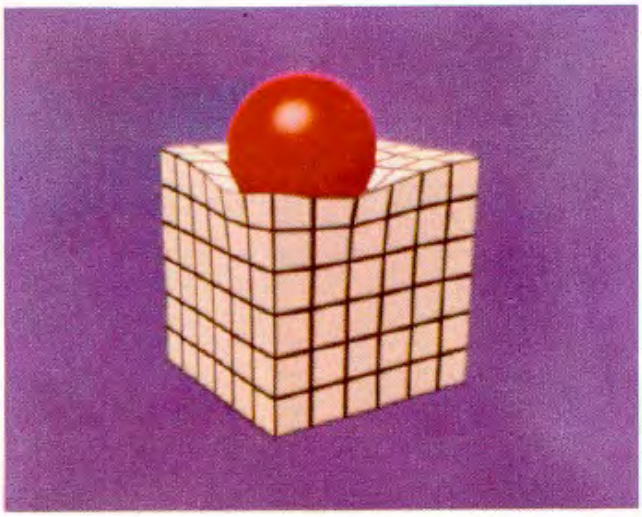


Figure 2.5: *The simulation result presented in (Terzopoulos et al. 1987), this method established the foundation of the modern energy-based cloth simulation method.*

System (KES) and the SiroFAST method.

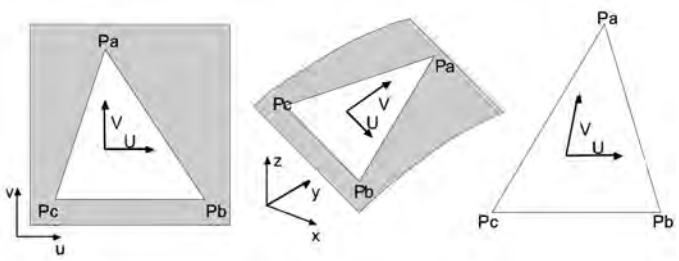


Figure 2.6: *A triangle face of cloth surface defined on the 2D cloth surface(left). The deformed triangle in 3D space(middle). The deformation of the weft-warp coordinate system of this triangle face(right).*

Volino et al. (2009) proposed a simulation model for the large deformations of textile. It can generates the nonlinear tensile behaviour of textile with accuracy and robustness meanwhile the process of the simulation remains simple. Differing from the majority of the existing cloth simulation systems, this model uses accurate strain-stress curves to represent the weft, warp and

shear tensile behaviour. 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.

Chen & Tang (2010) introduced a method for simulating inextensible cloth 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 cloth algorithm cannot preform correctly if some of the stiffness coefficients are set close to infinite. Thus they provide a method such that the physical-based simulation process is transferred into a deformation process of an initial developable mesh surface to a final mesh surface. Their method deforms cloth mesh by using energy minimization. Gauss-Newton iteration is used for the minimization of the energy function. However, this method can only handles the conservative force, it cannot handles 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 states therefore it is not suitable for dynamic use.

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

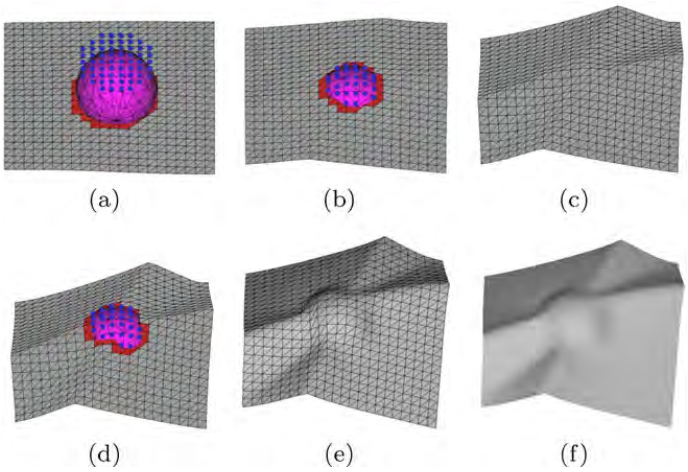


Figure 2.7: *blue points are the dynamic anchor points, (a), (b), and (d) are three intermediate states in sequence.The number of the dynamic anchor points are reduced gradually; (c) and (e) are the two sequent intermediate states where the potential energy of the mesh is lowered gradually. (f) is the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

cloth is largely constrained.

2.3.3 Hybrid Virtual Clothing

Hybrid virtual clothing method combines both physical simulation method and other types of deformation methods together such as data-driven method or geometrical method in order to compensate the shortcomings of each techniques to produce a fine result meanwhile minimize the calculation.

In order to reduce the computation complexity of the physical cloth simulation techniques, Rudomin (1990) presented a method uses geometrical approximation to reduce the computation time of physical virtual method technique introduced by Terzopoulos et al. (1987). Later on, Kunii & Gotoda (1990); Tsopelas (1991) proposed a series of hybrid virtual clothing methods that map geometrical modelled fine wrinkle details onto the physical simulated cloth mesh in order to avoid the heavy computation for simulating the

high resolution mesh. Hadap et al. (1999) introduced a texture based wrinkle modelling method for cloth simulation. This method generates the deformation detail based on the bump map that is created by user on a physical simulated coarse cloth object. However, because there can be a limited number of the pre-defined shape details or bump maps created by user, the variety of the deformation is consequently limited.

Data-driven method is widely used in combining with physical based simulation method to improve the efficiency of the cloth simulation. Cutler et al. (2005) introduced a kinematic method for generating wrinkles on cloth for CG characters. Winkles are created by artists using the geometric sculpture software and are stored into the database. The final cloth deformation details are generated by matching the similar stretch distribution of current cloth to the referencing posture of the character. However, the similarity between the reference posture and the coarse physical simulated cloth only exists on the tight fit cloth, the wrinkle database can not be used to generate shape detail on the loose fit cloth. Popa et al. (2009) introduced a data-driven method for generating the fine folds on the captured cloth model. In this method, the shape and the position of the wrinkle are captured from video footage based on the wrinkle's distinguishing shape characteristics, and the wrinkles are created by using stretch-minimizing deformation, which produces believable wrinkle shapes. However, the details of the captured wrinkles are limited by the resolution of cloth mesh therefore, significant amount of detail can be lost during the capturing process.

For pattern based cloth modelling techniques, cloth is consisted by a set of cloth patterns. In order to fit a cloth to a character, correct size of the patterns need to be determined. However, in the real world, altering the size of the cloth pattern is done by a knowledge intensive and time consuming process called "patter grading". Without the tailoring expertise among the animation artists, this process can only be done by iteratively performing the

entire cloth modelling process to try out the correct size of the patterns. In order to tackle this issue, Brouet et al. (2012) proposed an automatic method for cloth transferring between characters with different body shapes. This method fits a cloth onto a different character by adjusting the 3D cloth directly and patterns are extracted after the 3D cloth is fitted to the new character. The 3D cloth adjusting process is performed by combining the proportional scaling method with a constrained gradient based optimization process. The constrains are defined by associating the user selected point on the cloth mesh with a pair of points on the relevant bone and character skin. The distances between the selected vertices on cloth mesh to their associated point pairs are used as the parameters of the optimization. The optimization is performed by moving each vertex of the cloth mesh on its tangent space to minimize the distance changes. After 3D cloth is fitted onto the new character, the cloth patterns are extracted by using surface flattening method(Sheffer et al. 2005). Because the constrains of the optimization is defined by user and the selected reference points are varies on the different character model, the convergence of the optimization can be heavily affected by the user inputs. This method requires cross-parametrization between source and target models, the fit of the cloth can also be affected by the accuracy of the parametrization. Moreover, extracting the resized 2D cloth patterns requires extra computation and the distortion of the pattern shape can also be introduced by the surface flattening process.

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

on the visual result more than the physical accuracy, such as computer games and animations.

2.4 Geodesics

The standard posture used for modelling character differs from person to person. Traditional anthropomorphic data acquisition method requires character to stay in a standard posture in order to extract the correct measurement data. Therefore, applying traditional anthropomorphic measuring method to the same character in different posture will result different measurements. And the cloth that is adjusted based on those measurements will be adjusted into different sizes. This will cause the ill-fit of the modelled cloth. This thesis utilises the geodesic to mimic the tape measuring process in tailoring for the measurements extraction from different character in different posture. Geodesics are usually defined as the locally shortest path between two points on a curved space. Geodesic computation is a common operation in computer aided design, machine learning, medical image analysis and computer animation. Throughout years of research, many algorithms have been developed for various applications, such as surface-based brain flattening (Bartesaghi & Sapiro 2001; Wandell et al. 2000), mesh refining (Peyré & Cohen 2006), mesh segmentation (Katz & Tal 2003) and terrain navigation (Aleksandrov et al. 2005).

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 works of geodesic computation mainly focused on the convex polyhedrons. Sharir (& Schorr(1984) extend the Dijkstra's(Dijkstra 1959) algorithm into three dimension space, with a time complexity $O(n^3 logn)$. Later on,an exact solution of geodesics on a convex polyhedral surface was developed by Schreiber & Sharir (2006), with a reduced complexity of $O(n\log n)$.

For non-convex polyhedrons, a challenging issue is that, a geodesic path may go through the hyperbolic vertices of a polyhedron. Based on the definition of geodesic proposed by 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 sequences since these two edge sequences are all locally shortest around this hyperbolic vertex but not necessarily shortest in other local area, the computational result may not be correct.

In order to handle this problem, Mitchell et al. (1987b) (MMP algorithm) extended Dijkstras algorithm(Dijkstra 1959) and introduced the technique of "continuous Dijkstra" that propagates the distance information from the source outward in a Dijkstra-like manner. The "continuous Dijkstra" propagation performs as follows. Each edge is partitioned into several sections called "window", each window carries distance information through out the propagation. MMP algorithm costs $O(n^2 \log n)$ time to solve the "Single Source All Destinations" problem. Surazhsky et al. (2005) implemented MMP algorithm and furthermore, 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 $O(n \log n)$. However, the uniformity of the grid highly affects the approximation error. Therefore, performing FMM on an irregular and skinny triangulated grid will leads to significant high error. Bose et al. (2011) points out that the approximation error of FMM is unbounded.

Chen & Han (1990)(CH algorithm) provides an exact solution to the geodesic problem based on a key observation of "one angle one split" prin-

ciple with time complexity of $O(n^2)$. This method consists of a two parts. In the first part, the shortest path from a given source to each vertex on the mesh is computed and a set of windows is defined on each edge. Each window contains the information about the shortest path from the given source point to points on the current window. In the second part, 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 needs to be dealt with in the circumstance of a geodesic passes through a hyperbolic vertex because the planer unfolding to this type of vertex can be self-overlapped.

Xin & Wang (2009)(Improved CH algorithm) discovers that in CH 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 algorithms were developed specific for geodesic computation on polyhedron. Because "Continues Dijkstra" propagation method are used for propagating geodesic information outwards, the connectivity among the vertices is necessary. Therefore, they cannot handle the geodesic computation on scattered point cloud data. Moreover, the current geodesic algorithm usually requires "backtracing" method to retrieve the complete geodesic path information, it consumes large amount of time on the high resolution model.

For the geodesic computation on 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 in order to form the connectivity among the sampling points. For instance, Mémoli & Sapiro (2001) extends the FMM to be able to perform on point cloud data, where the resulting geodesic is an approximation over a coarse grid on the point cloud surface. Hofer & Pottmann (2004) employed moving least squares method to constrain the discrete geodesic curves onto the point cloud surface.

Ruggeri et al. (2006); Pottmann et al. (2010) proposed an approximate geodesic method on point set surface that employs an energy-minimization function to calculate geodesics. The initial curves are created by Dijkstras algorithm(Dijkstra 1959). Then all the initial curves are refined by performing energy minimization function to form the resulting linear approximation of the geodesics. Because the energy minimization consumes large amount of time, this method is not suitable for the multiple geodesic computation.

Calculating geodesics is a very time consuming process, the reviewed researches are mainly focus at improving the efficiency of the geodesic computation while maintain the calculation accuracy. However, because high resolution character model bas become more and more popular in computer animation, the current algorithms still consume large amount time to calculate geodesics on the high resolution model. Moreover, although technique such as fast marching method(Kimmel & Sethian 1998) is proposed to reduce the time complexity of the computation, its accuracy varies among the different character models with different topology.

2.5 Genetic algorithm

Modelling cloth from cloth patterns for a character requires each pattern to be adjusted individually. During the pattern adjustment, several criteria need to be satisfied in order to fit the cloth to the character and preserve the design of

the cloth. In this thesis, pattern adjustment is considered as an optimization process that has multiple objectives. For optimization algorithms, genetic algorithm dwarf others by its simplicity and efficiency.

In nature, individuals that suits the environment best survive. The ability of adapt to the changing environment is curial for the survival of individuals. Genetic algorithm are inspired by such mechanisms of evolution and natural selection in real world(Srinivas & Patnaik 1994b). The pioneering book of Holland (1992) demonstrated wide range of problems that can be solved by using evolutionary process in a highly parallel manner. The process utilise the concept of evolution is called genetic algorithm(Koza 1995). Now a days, after decades of research, genetic algorithm has become a practical and robust optimization method that can solve wide range of problem in real world.

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

There are two types of optimization based on the number of the objectives of the given problem, single-objective optimization and multi-objective optimization. For single-objective optimization, finding a single best solution is the purpose of the optimization. This method has been studied intensively for decades(Srinivas & Patnaik 1994b; Koza 1995). However, many real-world problems can not be described by a single objective, moreover, in many

cases, the optimization requires simultaneous optimization for multiple competing or conflicting objectives. Therefore, for multi-objective optimization, the best solution usually does not exist, instead, a group of "good" solutions are generated to form "pareto front"(Pareto 1906). Multi-objective genetic algorithm is designed specific for such type of problems.

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

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

Horn et al. (1994, 1993) proposed a method that based on the pareto domination tournament selection and equivalence class sharing(NPGA)(Horn et al. 1994). During the selection, a random number of individuals are selected into the comparison set, then two random individuals are compared

against the comparison set for domination, If one is dominated by the comparison set and the other is not, the dominating individual is selected for crossover. If neither or both individuals are dominated by the comparison set, the equivalence class sharing is performed. During this process, the individual that has the least number of neighbours with in the niche radius(Shir & Bäck 2006) is considered "better fits". This method only performs the pareto selection to a part of the population, therefore, high efficiency can be achieved and the pareto front can be well kept throughout large amount of generations. However, because the selection only be performed on a part of the population, the quality of the convergence is highly depends the choice of the niche radius and the number of individuals in the comparison set.

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

Knowles & Corne (1999) introduced the pareto archived evolution strategy(PAES). This method starts with a random solution of the multi-objective optimization problem. Then, this solution is mutated by a normally dis-

tributed probability function that has zero mean and constant mutation strength. The generated offspring is compared with the original solution and the one with better fitness value is used for the mutation of the next generation. A set of solution that contains good solution for each mutation is kept. Every time when mutation is performed, the offspring is compared against with its parent and all the other solutions in this set for their dominations. The PAES has provided a direct method for controlling the diversity of the pareto optimal, therefore, the premature convergence is less likely to occur and a higher probability to achieve real pareto front.

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

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

mization, therefore, outstands itself in the pattern adjusting problem in this thesis.

2.6 Summary

In this chapter, a brief history of the three types of virtual clothing methods in computer graphic has been reviewed.

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

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

In the area where the visual appearance is more important that the physical simulation accuracy such as films or games, hybrid virtual clothing method is applied in order to improve the efficiency of the virtual clothing while maintaining high visual realism. The hybrid virtual clothing methods combine different types of modelling or simulation method together in order to compensate the shortcomings of each method to be applied on its own

and provides a highly efficiency approach for the application. However, the hybrid method also has its own drawbacks, because it uses some techniques such as data-driven deformation method to simulate the wrinkle instead of producing them by solving the physical equation. The variety of the fine detail wrinkle is often limited by the non-physical deformation method.

The methods reviewed in this chapter are mainly focus at reproducing either the static shape or the dynamic behaviour of the cloth for a particular character. Both the results of cloth modelling and dynamic simulation can not be directly applied to another character with different body shapes or proportions. Large amount of manual operation is often required to create cloth for different characters or transfer a cloth from one character to a different one. Normally, the entire virtual clothing process must be performed from scratch in order to generate cloth for a different character with different body shapes or proportions. This duplication of effort can not be eliminated even when the same cloth design is required to fit to different characters. The current methods tie the character with their cloth together, and they can not be separated. Especially in today's virtual character applications, the number of the different characters that required in the virtual environment has been increased rapidly by the fast development of computer hardware and computer graphic techniques. The low efficiency caused by the duplication of effort in current virtual clothing methods has become a significant obstacle. Therefore, dressing different characters efficiently is still considered as a challenging task.

In order to model cloth from cloth patterns, measurements of the character model is crucial for determining the size of each pattern. This thesis utilises geodesic to mimic the tape measuring process in the tailoring in fashion industry. Moreover, using geodesic also results less variation when the posture of the character changes. However, computing geodesic on the high resolution model by using current geodesic algorithms is a very time consuming process. Moreover, followed with the increasing number of 3D scanned

character model, the geometrical representation of the character is no longer limited to polygon, nurbs or subdivision, but also the point cloud. In order to improve the efficiency of the geodesic computation while maintaining high accuracy as well as measuring the point cloud character model, three geodesic algorithms are proposed in this thesis respectively for accurate and approximate geodesic computation on triangulated manifolds as well as approximate geodesic computation on point cloud data set. The most important contribution is that the proposed approximation algorithm can reach linear time complexity with a bounded error on triangulated manifolds. The time consumed for solving the geodesic path between multiple pairs of source and destination has been largely reduced, at the meantime the accuracy of the solution is maintained.

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

For current gradient based optimization methods, preventing algorithm converge into the local minima is always a difficult problem that many researchers are working on. Among these optimization methods, genetic algorithm dwarf others by its simplicity and efficiency. Moreover, since it is

capable of searching through large area of the cost surface, it is less likely to converge into local minima. By using such an approach, the cloth pattern adjusting method presented in this thesis can automatically generates the best combination of the shapes and sizes of each pattern, it will fits the character and the original design of the cloth is preserved. Most importantly, By automating the process of pattern adjustment, the tailoring knowledge that required by current pattern based cloth modelling method is on longer needed. Therefore, animation artists can model varies clothes based on the large amount of existing cloth design patterns in today's fashion industry. Moreover, by automating the process of creating fit cloth for a character, the efficiency of generating cloth for different characters with different body shapes and proportions can be largely improved.

Chapter 3

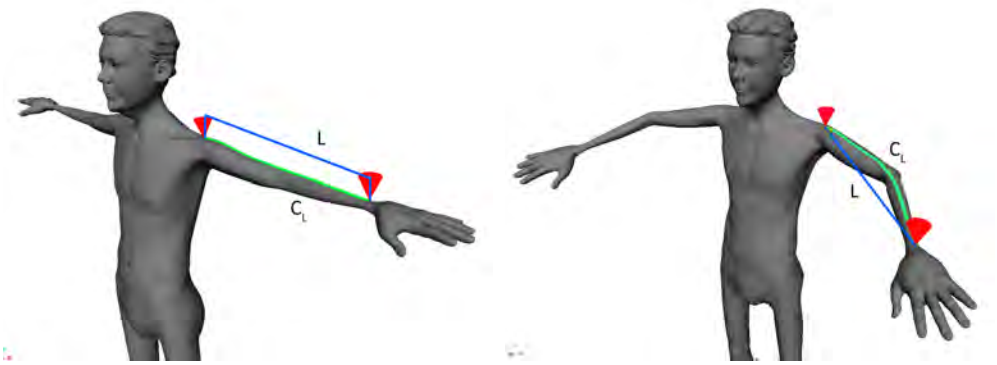
Geodesic Algorithm for Measurements

In the cloth production, the body of the customer need to be measured first to determine the size of the cloth. In anthropometric data acquisition, two types of measurements are associated with cloth making, length and circumference. The length measurements are usually obtained by calculating the euclidean distance between two datum points which are the landmarks on the skin. This procedure requires the character 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 character modelling 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".

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 character is required extend their arm straight, the length is the distance from

shoulder joint to the wrist joint. However, when measuring a character in a bent arm posture, the length of arm is acquired by adding the length of upper arm and lower arm. Another example is the acquisition of the height, for a character in a standing straight posture, the height can be obtained by calculating the Euclidean 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 tailoring, 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. In this thesis, geodesic is used for measuring characters. Comparing with the Euclidean distance calculation, geodesic distance also calculates the distance between two points, moreover, similar as the tape ruler measuring techniques, geodesic path lies on the underlying surface. Therefore, by using geodesics, it can extracts the measurement data effectively, moreover, it can also copes with different character postures. For example, Figure 3.1 demonstrates different measurement caused by different posture. The red cone indicates the datum points on the shoulder and wrist of the character. The green curve indicates the geodesic generated by the method presented in this chapter between two datum points and blue line is the straight line connecting two datum points. L denotes the length of the blue line(the Euclidean distance between two datum points), C_L denote the arc length of geodesic. The character is in two most common "T-pose". When the character in "T-pose" A, $L = 51.708$ and $C_L = 51.075$, when character in "T-pose" B, $L = 45.867$ and $C_L = 51.494$. This shows the geodesic distance between two datum points are more consistence in different posture than Euclidean distance when measuring length of the body part of character. Therefore in this thesis, the length measurement are acquired by



(a) *Character in "T-pose" A* **(b)** *Character in "T-pose" B*

Figure 3.1: *Measurements in different posture*

measuring the geodesic between two datum points on the body of the character in order to cope with different postures of the character.

In order to calculate geodesic, a graph with positive edge weight is required. The discrete polyhedral surface, as the most common shape representation in computer graphic, has this connectivity. Therefore, in the past few decades, many algorithms has been developed to compute the geodesic on discrete polyhedral surface.

Nowadays,with the easy access of 3D shape acquisition device, high-fidelity 3D scanned human body model has been widely accepted as simulation base model for clothing. However, The point cloud data from the 3D shape acquisition device does not have the connectivity between sampling points. This makes geodesic computation much more difficult.

This chapter presents a novel discrete geodesic computation scheme to tackle both input types. It consists of three algorithms, accurate geodesic algorithm on polyhedral surface, approximate geodesic algorithm on polyhedral surface and approximate algorithm on point cloud model. Among these algorithms, the approximate algorithm for polyhedral surface is also able to achieve linear time complexity whilst maintain a bounded small error. Moreover, in order to measure the character represented by point cloud data, a geodesic algorithm for computing geodesics directly on point cloud model is

also presented.

3.1 Introduction

For different shape representation, geodesic has different definitions.

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

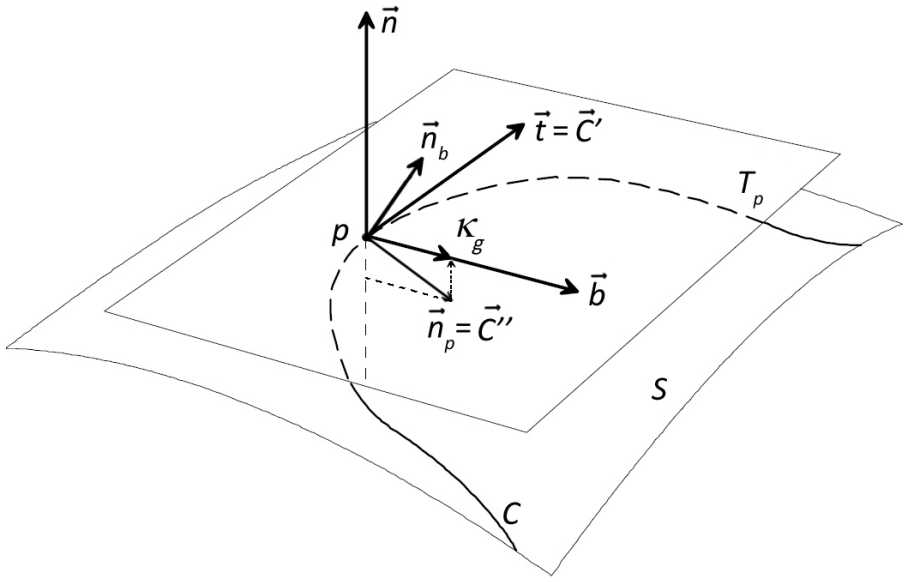


Figure 3.2: *Illustration of geodesic curvature*

Figure 3.2 demonstrates the geodesic curvature at a point on a curve, where S is a two-dimensional smooth surface. C is a curve on S, Tp is the tangent plane at the point $p \in S$, $\vec{n}$ is the surface normal at p, $\vec{C}'_s$ denote first order derivative of C and its the tangent vector at p that has arc length s to C. $\vec{b}$ is a vector perpendicular to $\vec{n}$ and $\vec{C}'_s$. $\vec{C}''_s$ denotes the second order derivative of C at p that has arc length s to C which is the geodesic curvature

vector of C at p. $\vec{n_b}$ is the binormal unit vector that perpendicular to $\vec{C_s'}$ and $\vec{C_s''}$. κ_g is the length of the projection of $\vec{C_s''}$ on $\vec{b}$.

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

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

Secondly, in order to define geodesic on convex polyhedral surface, the vertices of the polyhedral surface need to be categorized first. This can be done based on the total vertex angle. Polthier & Schmies (2006) defines total vertex angle as,

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

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

All vertices of a polyhedral surface can be categorized based on the sign of the "vertex angle excess", which can be calculated by $2\pi - \theta(v)$. Figure 3.3 demonstrates three types of vertices on polyhedral surface.

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

Definition 3 *Let S be a two-dimensional polygon surface, A piecewise curve C is a geodesic if one of the equivalent properties holds(Polthier & Schmies 2006): "A geodesic path P goes through an alternating sequence of hyperbolic vertices and (possibly empty) edge sequences such that the unfolded image of the path along any edge sequence is a straight line segment and the*

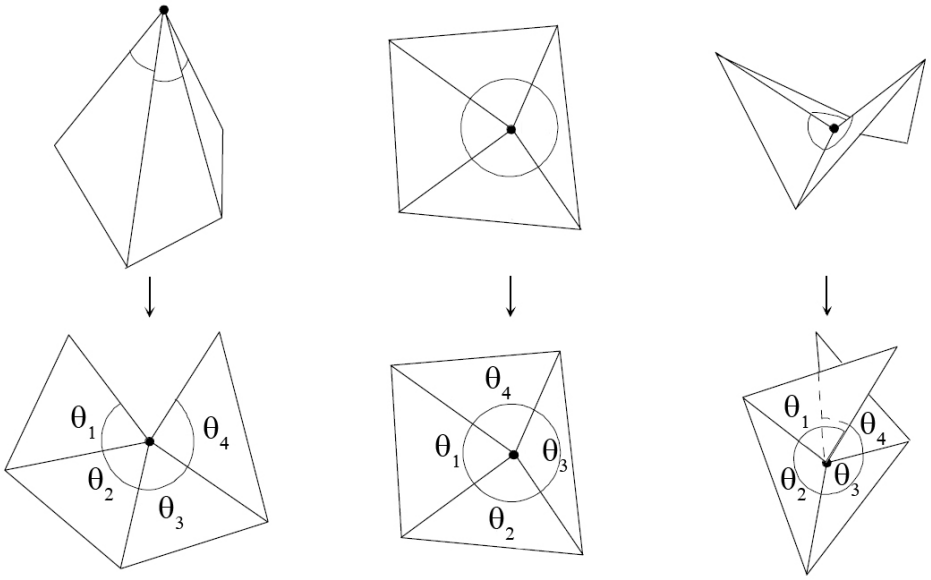


Figure 3.3: *Spherical Vertex(left), where* $2\pi - \Sigma\theta > 0$*; Euclidean Vertex(middle), where* $2\pi - \Sigma\theta = 0$*; Hyperbolic Vertex(right), where* $2\pi - \Sigma\theta < 0$.

angle of the path passing through a vertex is greater that or equal to π*. No edge can appear in more than one time in a edge sequence(Mitchell et al. 1987a)."*

Figure 3.4 illustrates a geodesic starts at point p and goes through a sequence edge $e_1 ... e_k$.

According to Chopp (1993), given an initial piecewise curve that passes a series of vertices on the polyhedral surface, after defining the tangent plane at every point on this piecewise curve, by moving each point on the piecewise curve iteratively for a small step at direction of $\vec{b}$ indicated in Figure 3.2, the geodesic curvature κ_g can be diminished gradually. When this iteration converges, which means the geodesic curvature at every point on this piecewise curve has been diminished, this piecewise curve becomes a geodesic on the polyhedral surface.

In the following sections, a novel geodesic curvature flow based geodesic

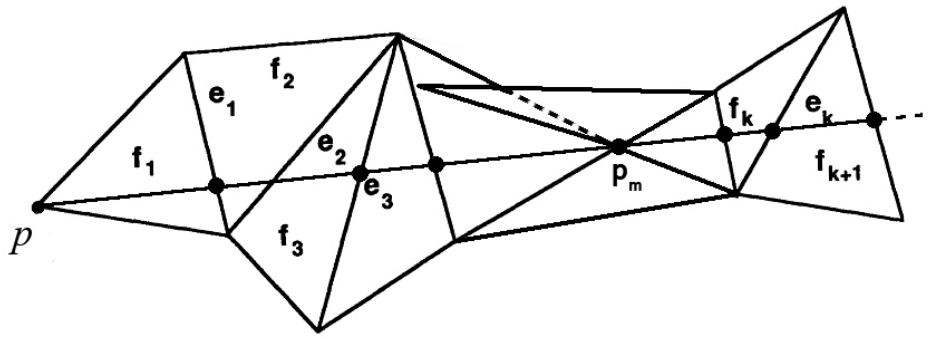


Figure 3.4: *Geodesic on a polyhedral surface, where $f_1 \cdots f_k$ is the planar unfolding of S. $e_1 \cdots e_k$ is the edge sequence E the geodesic goes through, p_m is a hyperbolic vertex on the polygon surface.*

computation scheme is introduced in detail. Three algorithms has been developed to solve different problems,

1. Accurate geodesics on polyhedral surface
2. Linear time complexity approximate geodesics with an bounded error on polyhedral surface
3. Approximate geodesics on point clouds

3.2 Geodesic Curvature Flow

Based on the geodesic curvature defined in Definition 1, a iterative scheme is proposed for computing geodesics over mesh and point clouds. This scheme employs iterative regression process to diminish the geodesic curvature, when the process converges, the curve becomes a geodesic.

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

tor and binormal vector of a point on a continuous and differentiable curve. Frenet-Serret formulas is defined as follow,

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

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

$$\vec{C_s'} = \vec{t} = \frac{\mathrm{d}C(s)}{\mathrm{d}s} \tag{3.1}$$

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

$$\vec{C_s''} = \vec{n_p} = \frac{\frac{\mathrm{d}\vec{t}}{\mathrm{d}s}}{\left\|\frac{\mathrm{d}\vec{t}}{\mathrm{d}s}\right\|} \tag{3.2}$$

The binormal unit vector of C is defined as:

$$\vec{n_b'} = \vec{n_p} \times \vec{t} \tag{3.3}$$

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

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

$$\begin{bmatrix} \frac{d\vec{t}}{ds} \\ \frac{d\vec{n_p}}{ds} \\ \frac{d\vec{n_b}}{ds} \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix} \begin{bmatrix} \vec{t} \\ \vec{n_p} \\ \vec{n_b} \end{bmatrix} \tag{3.4}$$

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

$$\begin{cases} \frac{d\vec{t}}{ds} = \kappa \vec{n_p} \\ \frac{d\vec{n_p}}{ds} = -\kappa \vec{t} + \tau \vec{n_b} \\ \frac{d\vec{n_b}}{ds} = -\tau \vec{n_p} \end{cases} \tag{3.5}$$

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

Equation 3.5 can be written as,

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

where $\kappa_n = \left\langle \vec{C_s''}, \vec{n} \right\rangle$ is the normal curvature at $C(s)$, $\kappa_g = \left\langle \vec{C_s''}, \vec{b} \right\rangle$ 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 C''. C_s'' denote the second order derivative to arc length s.

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*

$\vec{C_s'}$, $\vec{t_s}$ and $\vec{C_s''}$. C_s' is the first order derivative of $C(s)$, Thus, for any point $C(s)$, the speed can be defined by geodesic curvature is,,

$$\kappa_g = \left\langle (\vec{n} \times C_s'), C_s'' \right\rangle \tag{3.7}$$

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

$$k_g \vec{b} = C_s'' - \left\langle \vec{C_s''}, \vec{n} \right\rangle \vec{n} \tag{3.8}$$

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

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

Polthier & Schmies (2006) proved that, on polyhedral surface, if a discrete geodesic does not pass any hyperbolic vertex, this geodesic is also a shortest geodesic. In this thesis, the concept of vanishing geodesic curvature is exploited from smooth surface geodesic 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

$\widetilde{p_{i-1}p_ip_{i+1}}$ as shown in Figure 3.5, where C denotes a curve on smooth surface S and the $C(s)$ is the parametrized form of C by its arc length. p_{i-1},p_i and p_{i+1} are three sample points on C. Thus $\widetilde{p_{i-1}p_ip_{i+1}}$ denotes 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 3.9,

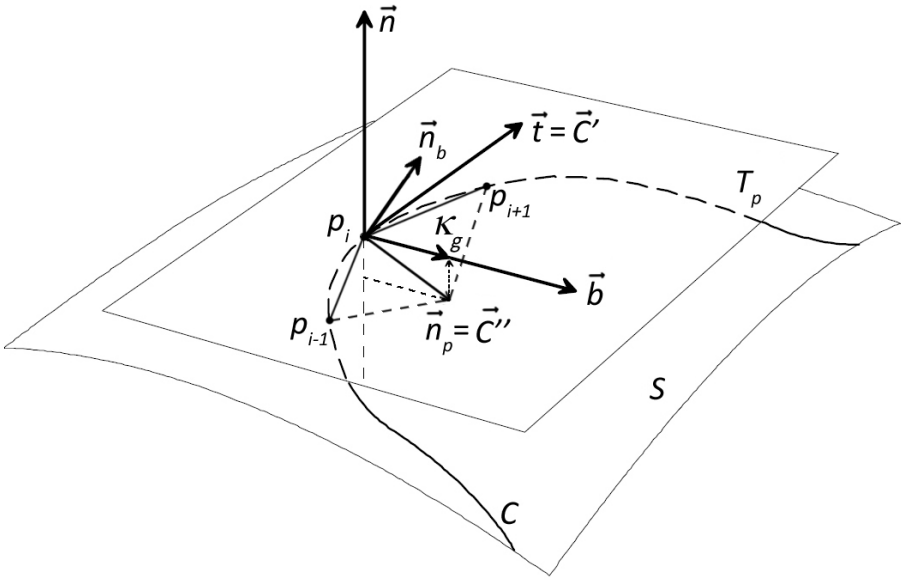


Figure 3.5: *Geodesic curvature on piecewise curve*

$$\vec{C}_s'' \approx \overrightarrow{p_ip_{i+1}} + \overrightarrow{p_ip_{i-1}} \tag{3.9}$$

In order to define the natural coordinate system on a vertex of a polyhedral surface, the tangent plane of S at p_i need to be confirmed. Assume the polyhedral surface is a discrete approximation of smooth surface S, given a point p_i on $C(s)$, the tangent plane of the polyhedron at vertex p_i can be defined by the fitting plane of the one-ring neighbour vertices of p_i. According to Definition 1, it can be said that, if the projection of 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, $C(s,t)$ reaches a stable states, and

becomes a geodesic.

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

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

and

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

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,

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

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

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

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

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

3.3 Geodesic on Mesh

3.3.1 Tangent Space Constraint

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

In our experiment presented in this chapter, during the iteration, the updated sample points tend to deviate from the surfaces. To solve this problem, a constrain in the tangent space needs to be applied to the movement of the

sample points. This can be implemented as follows.

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

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

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

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

where, μ is an iterative step size and $\overline{p_{i,t}}$ denotes the projection of p_i on the tangent plane at the closest vertex to p_i on S. In essential, the tangent space constraint of Equation 3.12 moves a point within the tangent space and therefore it tends to preserve the characterization of the underlying surfaces. After that, a steady solution can be formed by combining Equation 3.12 with Equation 3.11,

$$P_{t+1} = P_t + \mu(\vec{n}\vec{n}^T\overline{P_t} - \vec{n}\vec{n}^T P_t + KP_t - \vec{n}\vec{n}^T K P_t) \tag{3.13}$$

where, $\overline{P_t}$ is a column vector consisting of the coordinates of all the projections of the sampling point of curve C on S.

3.3.2 Implicit Euler scheme

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

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

Applying the backward scheme to Equation 3.13 yields,

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

where I is an identity matrix, μ denotes the step size. The resulting geodesic curvature flow of Equation 3.14 remains stable even at $\mu \rightarrow \infty$(Hundsdorfer & Verwer 2003).

Equation 3.14 is the linear equation form of $AX = B$ where $A = I + \mu(\vec{n}\vec{n}^T - K + \vec{n}\vec{n}^T K)$, $B = P_t + \mu\vec{n}\vec{n}^T\overline{P_t}$. This equation can be solved efficiently by LU factorization with full pivoting presented by Trefethen & Bau (1997).

3.3.3 Least Squares scheme

Although the implicit Euler scheme provides an efficient method for solving Equation 3.14. The experiment shows that the geodesics result from Equation 3.14 still requires a few iterations in order to achieve convergence.

Experiments show that during the convergence of Equation 3.11 , all the sample points on the geodesic curve are updated within the tangent space except two endpoints p_1 and p_m of curve $C(t)$. Therefore, Equation 3.11 can be written in a least squares form in which two endpoints p_1 and p_m are the constrains(Jiang 1998),

$$\left(K - \vec{n}\vec{n}^T K\right) p = [p_{1_x}, p_{1_y}, p_{1_z}, 0, ..., 0, p_{m_x}, p_{m_y}, p_{m_z}]^T \tag{3.15}$$

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

$$\vec{n} = \begin{bmatrix} 0 & & & & \\ 0 & & & & \\ 0 & & & & \\ & n_{2_x} & & & \\ & n_{2_y} & & & \\ & n_{2_z} & & & \\ & & \ddots & & \\ & & & n_{m-1_x} & \\ & & & n_{m-1_y} & \\ & & & n_{m-1_z} & \\ & & & & 0 \\ & & & & 0 \\ & & & & 0 \end{bmatrix}_{3m \times m}$$

where K is a $3m \times 3m$ size matrix, $\vec{n}$ denotes the normal of the tangent plane. Essentially, the system Equation 3.15 is the 1^{st} order approximation to the geodesic curvature flow. A linear system can be constructed from Equation 3.15,

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,3m} \\ A_{1,0} & A_{1,1} & & \\ \vdots & & \ddots & \vdots \\ A_{3m,0} & & \cdots & A_{3m,3m} \end{bmatrix} p = \begin{bmatrix} p_{1_x} \\ p_{1_y} \\ p_{1_z} \\ 0 \\ \vdots \\ 0 \\ p_{m_x} \\ p_{m_y} \\ p_{m_z} \end{bmatrix} \tag{3.16}$$

where, $A = K - \vec{n}\vec{n}^T K$, m is the number of the sampling point on the piecewise curve C. Combining Equation 3.16 with tangent plane constrain

defined by Equation 3.12, the Equation 3.16 can be written as,

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,3m} \\ A_{1,0} & A_{1,1} & & \\ \vdots & & \ddots & \vdots \\ A_{3m,0} & & \cdots & A_{3m,3m} \\ N^*_{0,0} & N^*_{0,1} & \cdots & N^*_{0,3m} \\ N^*_{1,0} & N^*_{1,1} & & \\ \vdots & & \ddots & \vdots \\ N^*_{3m,0} & & \cdots & N^*_{3m,3m} \end{bmatrix} p = \begin{bmatrix} p_{1_x} \\ p_{1_y} \\ p_{1_z} \\ 0 \\ \vdots \\ 0 \\ p_{m_x} \\ p_{m_y} \\ p_{m_z} \\ B^*_{0,0} \\ \vdots \\ B^*_{3m,0} \end{bmatrix} \tag{3.17}$$

where, $N^* = \vec{n}\vec{n}^T$, $B^* = \vec{n}\vec{n}^T\bar{P}$. $\bar{P}$ is a column vector consist of all the $\bar{p}_i$ which are the projections of the p_i on the tangent plane defined by $\vec{n}$ and the closest vertex on S to p_i. m denotes the number of sample point on curve. The coefficient matrix on the left hand side of the system has size of $(2 * 3m) \times 3m$ and the one on the right hand side of the system has the size of $(2 * 3m) \times 1$.

Note that Equation 3.17 cannot computes a geodesic curve from scratch but requires the initial location of the sample points for computing $\bar{p}$. The vertices near a geodesic path are usually taken as the initial sample points on the path. Such initial sample points are used for computing $\bar{p}$ and $\vec{n}$ in Equation 3.17, which is then solved to obtain the final sample locations. According 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 3.17 cannot guarantees the updated sample points lying on the polyhedral surface since the tangent space and actual polyhedral surface space are two

different surface. the actual polyhedral surface is reckoned as inscribed polygon of the surface that 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.

3.3.4 Geodesic Path Projection

Although the sample point updated by Equation 3.13 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 the mesh. This is showed in Figure 3.6,

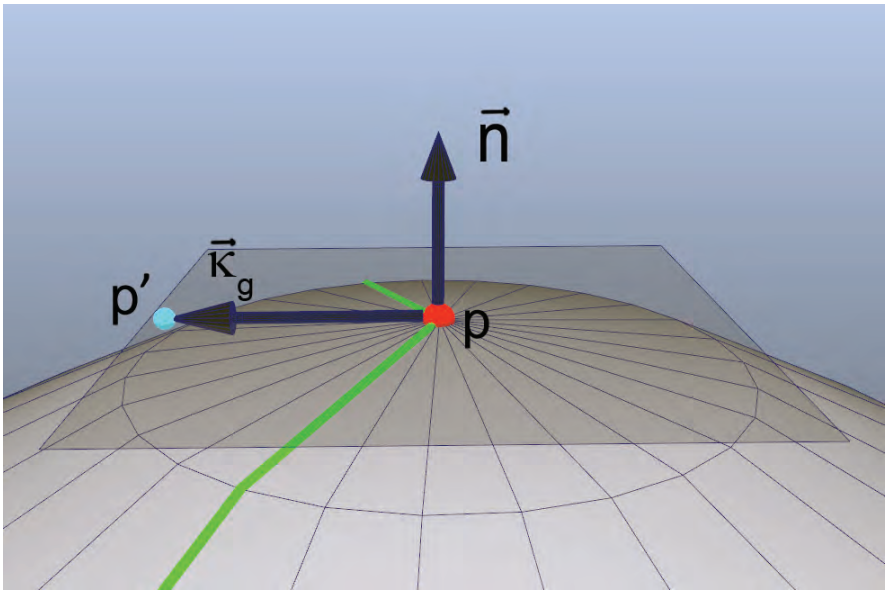


Figure 3.6: *Derived point caused by updating point, green piecewise curve is the curve before updated by Equation 3.11, gray plane is the tangent plane at point p (red point), the point p' (light blue point) is the updated point.)*

In order to calculate geodesics on the polyhedral surface, the offset geodesic curve need to be projected on to the polyhedral surface. Polthier & Schmies (2006) defines geodesic as when at any point on the curve, geodesic curvature vanishes. On polyhedral surface, if a curve fits this 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 3.7.

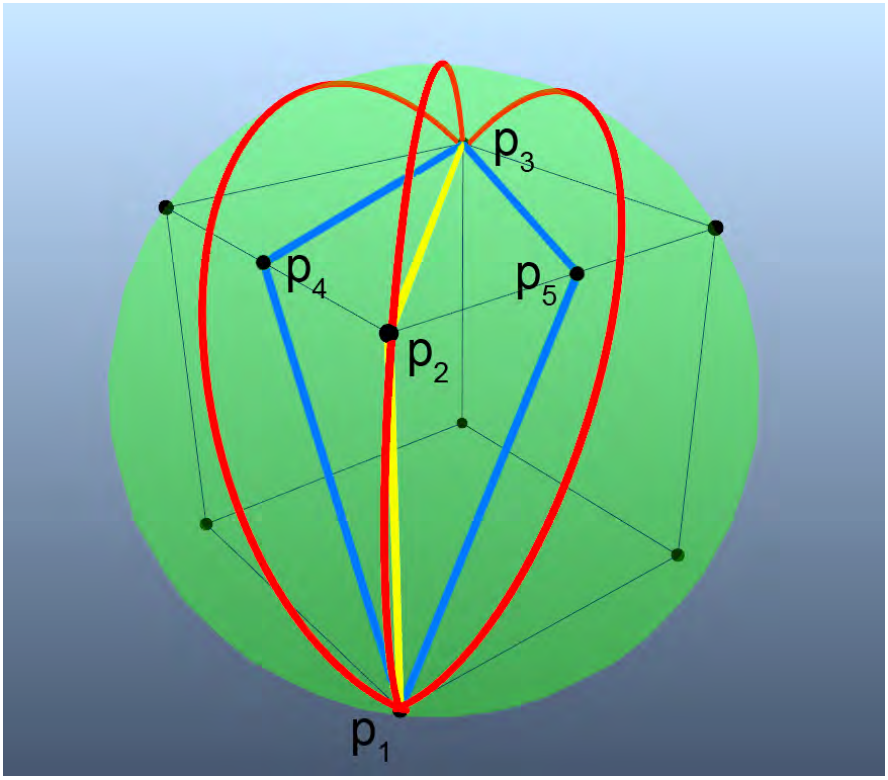


Figure 3.7: *Geodesic on smooth surface and its projection on the its inscribed polygon mesh, the green sphere contains its inscribed cubic, the red curves is the geodesic on sphere (great circle arc),* p_1,p_2 *and* p_3 *are points on sphere,* p_4 *and* p_5 *are the centre point on its edge.*

In Figure 3.7 both the yellow line $\widetilde{p_1p_2p_3}$ and blue lines $\widetilde{p_1p_4p_3}$, $\widetilde{p_1p_5p_3}$ are the projections of their corresponding red geodesic path on sphere. The path $\widetilde{p_1p_2p_3}$ 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 $\widetilde{p_1p_4p_3}$ and $\widetilde{p_1p_5p_3}$ 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 direction of the projection need to be defined. According to the definition of geodesic curvature, geodesic curvature vector $\vec{\kappa_g}$ of a curve C at point p is the projected

vector of the curvature vector $\vec{C''}$ at p onto the tangent plane T_p at point pP on surface S. This is shown in Figure 3.2. Therefore, the curvature κ at point p can be written as:

$$\vec{\kappa} = \vec{\kappa_g} + \vec{\kappa_n}$$

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

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

where θ is the angle between the osculating plane of C at point p and the tangent plane T_p. Therefore, if p moves along the direction of $\vec{C''}$, the osculating plane of C at point p will remains still so as the θ. Hence the geodesic curvature κ_g will remain identical. Therefore, for each sample point on C the projection direction is its C''. However, since there is no guarantee that every successive sample point can 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 neighbour face of the face contains current projected point. Here, by using osculating plane of C at p, we have developed a method connecting all the projected point of p. The procedure is described as follows.

Firstly, select p_0 as the starting point that always lies on a vertex of the mesh. With two successive sample points p_1 and p_2 on the floating geodesic, these three points form a plane cuts the mesh S. This plane, $\Delta p_0p_1p_2$, is viewed as the approximation of the osculating plane at p_1, see Figure 3.8.

In Figure 3.8, the red line indicate the piecewise approximation of a geodesic. p_0,p_1,p_2,p_3 is the sample point on the geodesic path $\widetilde{p_0p_3}$. $\Delta p_0p_1p_2$ is the approximation of the osculating plane at p_1. fp_1 fp_4 are the vertices on

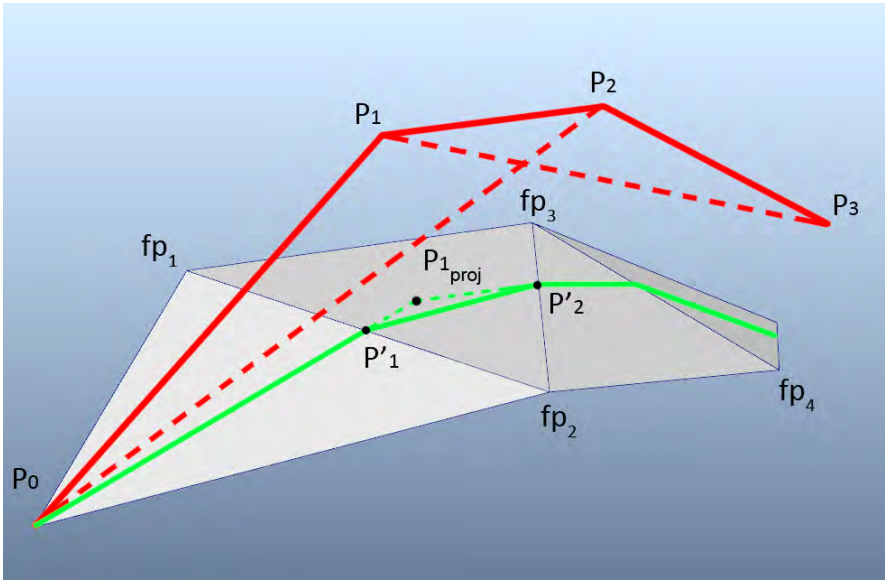


Figure 3.8: *Projection of a geodesic path.*

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 the face $\Delta p_0 f p_1 f p_2$. Firstly, starts from the source point p_0, the rectifying plane of the projected path is approximated by $\Delta p_0 f p_1 f p_2$. As a result, the tangent vector of the projected geodesic at p_0 can be calculated by Equation 3.18.

$$\vec{v} = \vec{N_1} \times \vec{N_2} \tag{3.18}$$

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 $\overline{fp_1 fp_2}$. 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 $\Delta f p_1 f p_2 f p_3$ to the current face $\Delta p_0 f p_1 f p_2$ 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 3.8, $\vec{N_1}$ is updated by the normal vector of approximated osculating plane $\Delta p_1p_2p_3$ at point p_2. Starting from the projection $p_{1_{proj}}$ 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 3.8. The projection method is summarized in Algorithm 1

Algorithm 1 Geodesic Projections on Mesh

procedure PROJECT GEODESIC PATH ON MESH(A mesh S, and a floating geodesic path $\widetilde{p_0p_n}$)
  for $i = 1, i < m, i++$ **do**
    Select the successive vertices of p_i and build osculating plane $\triangle p_{i-1}p_ip_{i+1}$
    $\vec{N_1} \leftarrow$ normal vector of $\triangle p_{i-1}p_ip_{i+1}$
    $F_{end} \leftarrow$ Face contains projection of p_i on S
    select p_{i-1} as the starting point p_s
    while $F' \neq F_{end}$ **do**
      for $F' \in$ adjacent faces of p_s **do**
        $\vec{N_2} \leftarrow$ normal vector of current face
        $\vec{t} \leftarrow p_s + \vec{N_1} \times \vec{N_2}$
        Compute the intersection point p' that an edge of current face F' intersects with $\vec{t}$
        if p' is within the edge **then**
          Add p' into projection path
          Update p_s by p'
          Update F' by the adjacent face of F'
        end if
      end for
    end while
    Update p_s by p_i
  end for
end procedure

3.3.5 "Continuous Dijkstra" Propagation

As described in Equation 3.11, 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 the 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 involves "comparison sorting algorithm" which is a searching algorithm costs $log(n)$ time to n factorial possible orderings of a data set (Cormen et al. 2001). This process requires significant amount of time to complete especially when performing on large model. Because Equation 3.11 only requires an initial path that approximates to the shortest path. 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 1^{st} ring vertices of p_s are pushed into an array denoted as "wavefront" W that is equivalent to the propagation boundary. Each edge that connects p_s and a point in W is the geodesic path between them. This is due to within a triangle, the shortest path between two points is the edge that connects these two points. 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 step. At this moment, r_{max} is updated as the length of the longest edge that connects p_s and W.

After the first W is formed by 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 in order to form its initial path from p_s to pn_i. This process is illustrated in Figure 3.9.

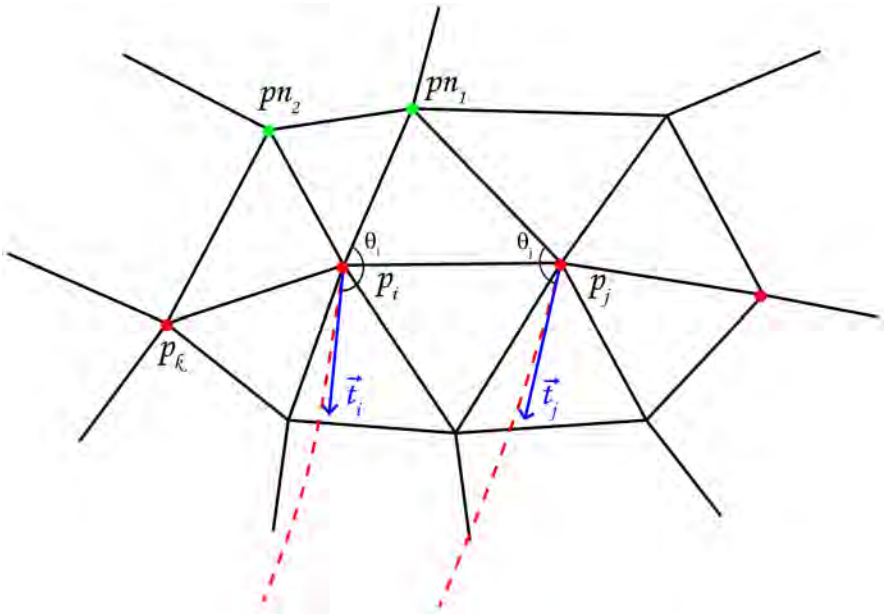


Figure 3.9: *Calculating geodesic path for a unvisited vertex, red point p_i,p_j and p_k are visited vertex in wavefront W, red dash line indicates geodesic path from source to p_i and p_j, vector $\vec{t_i}$ and $\vec{t_j}$ is the tangent vector at p_i and p_j respectively, pn_1 and pn_2 are one-ring neighbour of p_i that has not been visited.*

Where, pn_1 and pn_2 is the one-ring neighbour N' of p_i that have not been visited. For a random vertex $pn_1 \in N'$, the visited vertices in one-ring neighbour of pn_i is selected. In Figure 3.9, p_i and p_j are selected. In the next step, for all the visited vertices that connect to pn_1, the angle θ between $\vec{t_i}$ and the edge that connects pn_1 and the current selected visited vertex is calculated. $\vec{t_i}$ denotes the tangent vector at current visited vertex on its geodesic path. The parent node of pn_1 is selected from the visited vertex with the largest θ.

In Figure 3.9, the included angle θ_i between $\overline{p_i pn1}$ and $\vec{t_i}$ is greater than the included angle θ_j between $\overline{p_j pn1}$ and $\vec{v_j}$. Therefore, it can be observed that, comparing the two curves of edge $\overline{p_i pn_1}$ plus geodesic path at p_i and edge $\overline{p_j pn1}$ plus geodesic path at p_j, the curve of edge $\overline{p_i pn_1}$ plus geodesic

path at p_i are straighter than the curve of edge $\overline{p_i pn_1}$ plus geodesic path at p_i. Thus, vertex pn_1 is pushed into the end of the geodesic path at vertex p_i as the initial path for pn_1. For every new point that 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 in order to retrieve the geodesic path. The aforementioned method propagates the geodesic information from the interior ring to the external one. However, without the help of priority queue, propagates the geodesic information from the vertex with smaller geodesic distances to the vertex with larger distances can not be guaranteed. Therefore, in order to maintain the isotropy of the propagation, a propagation radius limit r_{max} is employed. After pn_1 has selected p_i as its parent node, the estimate geodesic distance d_{est} from source to pn_1 can be approximated by the length of edge $\overline{pn_1 p_i}$ plus geodesic distance from source to p_i. Then Equation 3.11 is performed to calculate the geodesic path from source to pn_1 only if d_{est} is smaller than the current r_{max}. After that, p_i is stored into an array W' as the "new wavefront". After all the vertices in W has been popped out, W is replaced by W'.

However, if d_{est} is larger than the current r_{max}, then pn_1 is put into an array W_{next} as the "next wavefront" for the next propagation. 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 at every step. Because r_{max} is updated every time when W is replaced by W_{next}, as a result, in between the updates of r_{max}, the length of all the geodesic paths are in between the r_{max} and d_{maxEst}. Therefore, the causality is guaranteed by r_{max}. This propagation algorithm is summarized in Algorithm 2.

Algorithm 2 Accurate Geodesics Algorithm

procedure INITIALIZATION(A mesh S, and a source p_s)
  for $p_i \in S$ **do**
    $p_i.geoDist \leftarrow \infty$
    $p_i.geoPath \leftarrow \emptyset$
  end for
end procedure

procedure CALCULATE ONE RING NEIGHBOUR OF p_s(W , r_{max})
  for $p_i \in W$ **do**
    $d_{est} \leftarrow Dist(p_i, p_s)$
    $p_i.geoDist \leftarrow d_{est}$
    $p_i.geoPath \leftarrow p_s + p_i$
    if $currentDist > r_{max}$ **then**
      $r_{max} \leftarrow d_{est}$
    end if
  end for
end procedure

function GETPARENTNODE(W,v)
  for $v' \in oneRingNeighbourOf(v)$ **do**
    $\theta = 0$
    if $v' \in W$ **then**
      $\vec{v_1} \leftarrow v - v'$
      $\vec{v_2} \leftarrow getTangent(v)$ ▷ Return the tangent vector of geodesic path from source to v' at point v'
      if $\theta < angleBetween(\vec{v_1}, \vec{v_2})$ **then**
        $\theta \leftarrow angleBetween(\vec{v_1}, \vec{v_2})$
        $parentNode \leftarrow v'$
      end if
    end if
  end for
  return $parentNode$
end function

Algorithm 2 accurate geodesics algorithm (continued)

procedure CONTINUES DIJKSTRA PROPAGATION(Calculate geodesic for every $p_i \in S$)
  while $W \neq \emptyset$ **do**
    for $p_i \in W$ **do**
      for $pc_j \in oneRingNeighbourOf(p_i)$ **do**
        if $pc_j \in W$ **OR** $pc_j.geoDist \neq \infty$ **then**
          continue
        else
          $parentNode \leftarrow$ GETPARENTNODE(W,pc_j)
          $d_{est} \leftarrow = distanceBetween(parentNode, pc_j) + parentNode.geoDist$
          if $d_{est} < rmax$ **then**
            Perform Equation 3.11 on $initialPath$ results geodesic path from source to pc_j
            Perform Algorithm 1 to project floating path $pc_j.geoPath$ onto S
            Update the neighbourhoods for the samples point on $pc_j.geoPath$ separately
            $W' \leftarrow pc_j$
          else
            Update d_{maxEst} if $d_{est} > d_{maxEst}$
            $W_{next} \leftarrow pc_j$
          end if
        end if
      end for
    end for
    $W \leftarrow W'$
    $W' \leftarrow \emptyset$
    if $W = \emptyset$ **then**
      $W \leftarrow W_{next}$
      $W_{next} \leftarrow \emptyset$
      $r_{max} \leftarrow d_{maxEst}$
    end if
  end while
end procedure

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

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

A challenging issue of using Algorithm 2 to calculate geodesic is to solve the large sparse linear system of Equation 3.16. Moreover, followed by

the propagation of "wavefront" the number of the points in the initial path that need to be updated by Equation 3.11 increases. Therefore, the size of the matrix in Equation 3.16 also increases. This step is the most time consuming procedure in the presented algorithm.

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 3.11 to avoid solving large sparse linear system. When an initial estimation of a geodesic path is obtained, as shown in Figure 3.9, a fixed-size window that covers the last w sample points is employed. Then Equation 3.11 is applied to w to calculate the local geodesic patch. Let $\widetilde{p'_0 p'_{n-1}}$ denotes the geodesic path from source to the parent of p' parent. Thus, only the section $\widetilde{p'_{n-w} p'_n}$ is updated to form the approximated geodesic path. This algorithm is summarized in Algorithm 3.

In Algorithm 3, because the window size w is a constant number, therefore, the matrix size in the linear system of Equation 3.11 does not change throughout the propagation of the "wavefront", and the time used for solving the Equation 3.11 is also a constant. By introducing the concept of window to each geodesic update, Algorithm 3 is able to achieve linear time complexity. Figure 3.10 demonstrates the geodesics results on four different characters performed by Algorithm 3.

Algorithm 3 Approximate Geodesics Algorithm

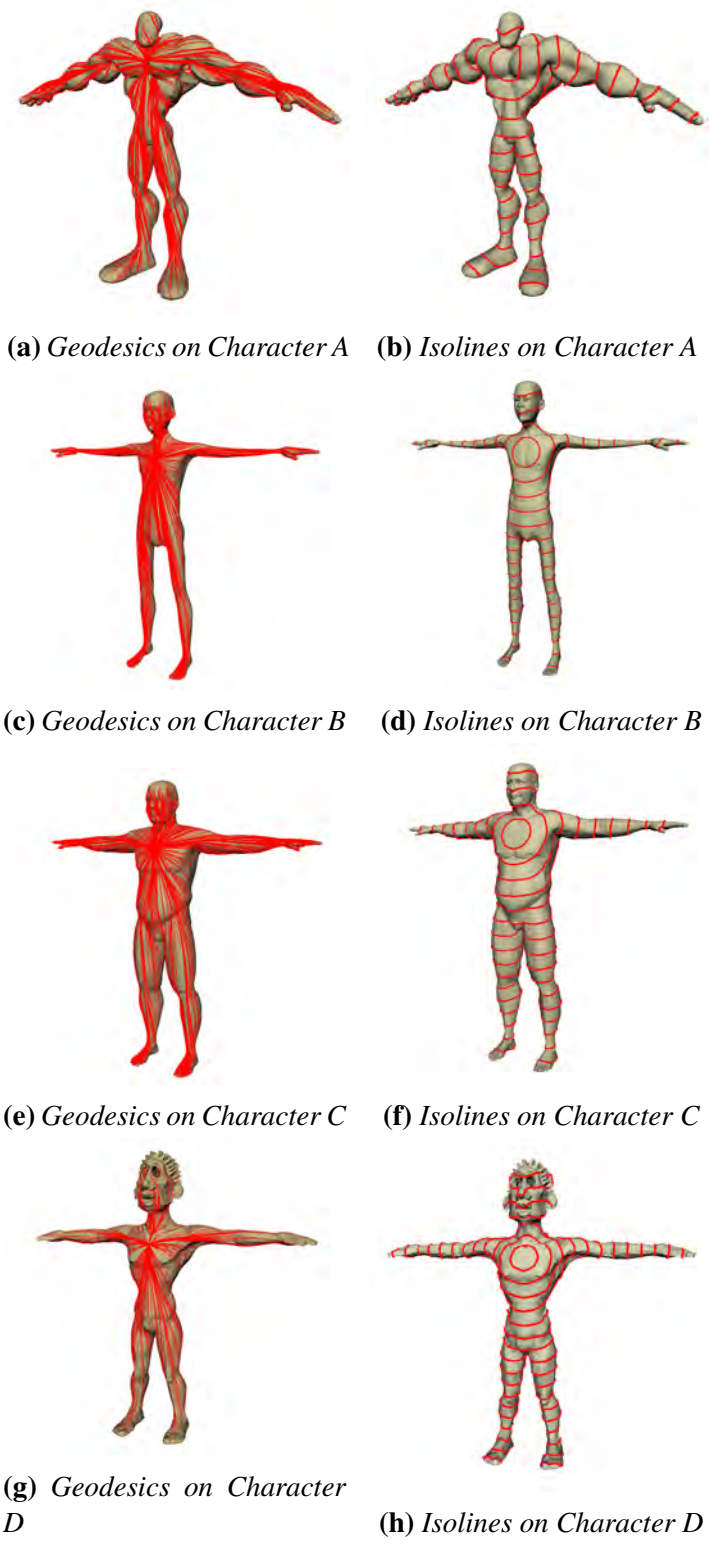
procedure INITIALIZATION(A mesh S, and a source p_s)
  for $p_i \in S$ **do**
    $p_i.geoDist \leftarrow \infty$
    $p_i.geoPath \leftarrow \emptyset$
  end for
end procedure

procedure CALCULATE ONE RING NEIGHBOUR OF p_s(W , r_{max})
  for $p_i \in W$ **do**
    $d_{est} \leftarrow Dist(p_i, p_s)$
    $p_i.geoDist \leftarrow currentDist$
    $p_i.geoPath \leftarrow p_s + p_i$
    if $currentDist > r_{max}$ **then**
      $r_{max} \leftarrow d_{est}$
    end if
  end for
end procedure

function GETPARENTNODE(W,$v_{unvisited}$)
  for $v' \in oneRingNeighbourOf(v_{unvisited})$ **do**
    $\theta = 0$
    if $v' \in W$ **then**
      $\vec{v_1} \leftarrow v_{unvisited} - v'$
      $\vec{v_2} \leftarrow getTangent(v_{unvisited})$ ▷ Return the tangent vector of geodesic path from source to v' at point v'
      if $\theta < angleBetween(\vec{v_1}, \vec{v_2})$ **then**
        $\theta \leftarrow angleBetween(\vec{v_1}, \vec{v_2})$
        $parentNode \leftarrow v'$
      end if
    end if
  end for
  return $parentNode$
end function

Algorithm 3 Approximate Geodesics Algorithm (continued)

procedure CONTINUES DIJKSTRA PROPAGATION(Calculate geodesic for every $p_i \in S$)
  while $W \neq \emptyset$ **do**
    for $p_i \in W$ **do**
      for $pc_i \in oneRingNeighbourOf(p_i)$ **do**
        if $pc_j \in W$ **OR** $pc_j.geoDist \neq \infty$ **then**
          continue
        else
          $parentNode \leftarrow$ GETPARENTNODE(W,pc_j)
          $d_{est} \leftarrow = distanceBetween(parentNode, pc_j) +$ $parentNode.geoDist$
          if $d_{est} < rmax$ **then**
            Perform Equation 3.11 on the $\widetilde{p_{n-w}p_n}$ part of $initialPath$ results geodesic path from p_{n-w} to pc_j
            Combine $\widetilde{p}_0 p_{n-w-1}$ with $\widetilde{p}_{n-w} pc_j$ to form the approximate geodesic from source to pc_j
            Perform Algorithm 1 to project floating path $pc_j.geoPath$ onto S
            Update the neighbourhoods for the samples point on $pc_j.geoPath$ from $n - w^{th}$ to n^{th} point separately
            $W' \leftarrow pc_j$
          else
            Update d_{maxEst} if $d_{est} > d_{maxEst}$
            $W_{next} \leftarrow parentNode$
          end if
        end if
      end for
    end for
    $W \leftarrow W'$
    $W' \leftarrow \emptyset$
    if $W = \emptyset$ **then**
      $W \leftarrow W_{next}$
      $W_{next}\emptyset$
      $r_{max} \leftarrow d_{maxEst}$
    end if
  end while
end procedure



(a) *Geodesics on Character A* **(b)** *Isolines on Character A*

(c) *Geodesics on Character B* **(d)** *Isolines on Character B*

(e) *Geodesics on Character C* **(f)** *Isolines on Character C*

(g) *Geodesics on Character D* **(h)** *Isolines on Character D*

Figure 3.10: *Geodesics on character*

3.4 Geodesic on Point Cloud

Followed by the fast development of 3D shape acquisition device, the geometrical representation of the character model is not limited to polyhedron or implicit surface such as nurbs, but also 3D scanned point cloud data. Therefore, it is important to be able to extract measurements data directly from the point cloth model.

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

In order to facilitate the connectivity among the data 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 considered as the destination for all the points within this cell. This regular grid distributes destinations evenly over the point cloud data. A destination represents all the points contained by a cell, therefore the total number of destinations n is much smaller than the actual number of the scattered points in the 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, x_{seg} denotes the number of cells in the x direction, y_{seg} denotes the number of cells in the y direction, c_x,c_y,c_z are the 3D coordinate of the centre point of a cell, min_y,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, building 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 one-ring neighbour of a given cell. In the implementation of this algorithm, to optimize the memory consumption, the array represents the full regular grid 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 with $O(n)$ space cost. Because the number of non-empty cells are much less than the number of the cell in a full regular grid. The actual memory consumption is close to $O(n)$. This process is summarized below.

Algorithm 4 Build Regular Grid for Point Cloud

procedure BUILD REGULAR GRID FOR POINT CLOUD(A point cloud data set S, a source p_s, a current cell b that contains p_s, the intervals $\triangle x$ of a grid, and number of neighbours, d)
    Build up a searching tree for ANN performing on S
    Select the minimum point p_{min} of the bounding box $BBox$ of S. c is the cell that contains the source point p_s.
    while $b \in BBox$ **do**
        Get the closest point $p \in S$ to the centre of b
        if $p \in b$ **then**
            Store b into regular grid G
            Use ANN fixed radius search to retrieve all the data points in S that are contained by b
            Compute mean of b as the destination of b
        end if
        Move b orthogonally by $\triangle x$
    end while
end procedure

After regular grid G has been built, the "Continue Dijkstra" is employed here to form the initial path for solving Equation 3.11. This process is illustrated below.

Algorithm 5 Approximate Geodesics Algorithm on Point Cloud

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

procedure CALCULATE ONE RING NEIGHBOUR OF c_s(W , r_{max})
    for $c_i \in W$ **do**
        $d_{est} \leftarrow Distance(c_i.centre, c_s.centre)$
        $c_i.geoDist \leftarrow currentDist$
        $c_i.geoPath \leftarrow c_s + c_i$
        if $currentDist > r_{max}$ **then**
            $r_{max} \leftarrow d_{est}$
        end if
    end for
end procedure

function GETPARENTNODE(W,$c_{unvisited}$)
    for $c' \in oneRingNeighbourOf(c_{unvisited})$ **do**
        $\theta = 0$
        if $c' \in W$ **then**
            $\vec{v_1} \leftarrow c_{unvisited} - c'$
            $\vec{v_2} \leftarrow getTangent(c_{unvisited})$ ▷ Return the tangent vector of geodesic path from source to c' at point c'
            if $\theta < angleBetween(\vec{v_1}, \vec{v_2})$ **then**
                $\theta \leftarrow angleBetween(\vec{v_1}, \vec{v_2})$
                $parentNode \leftarrow c'$
            end if
        end if
    end for
    return $parentNode$
end function

Algorithm 5 Approximate Geodesics Algorithm on Point Cloud(continued)

procedure CONTINUES DIJKSTRA PROPAGATION(Calculate geodesic for every $c_i \in G$)
  while $W \neq \emptyset$ **do**
    for $c_i \in W$ **do**
      for $c_i' \in oneRingNeighbourOf(c_i)$ **do**
        if $c_i \in W$ **OR** $c_i'.geoDist \neq \infty$ **then**
          continue
        else
          $parentNode \leftarrow$ GETPARENTNODE(W,c_i')
          $d_{est} \leftarrow = distanceBetween(parentNode, c_i') + parentNode.geoDist$
          if $d_{est} < rmax$ **then**
            Perform Equation 3.11 on $initialPath$ results geodesic path from source to c_i'
            Update the neighbourhoods for the samples point on $c_i'.geoPath$ separately
            $W' \leftarrow c_i'$
          else
            Update d_{maxEst} if $d_{est} > d_{maxEst}$
            $W_{next} \leftarrow parentNode$
          end if
        end if
      end for
    end for
    $W \leftarrow W'$
    $W' \leftarrow \emptyset$
    if $W = \emptyset$ **then**
      $W \leftarrow W_{next}$
      $W_{next}\emptyset$
      $r_{max} \leftarrow d_{maxEst}$
    end if
  end while
end procedure

In Algorithm 5, in order to determine the neighbourhood for each sample point on a geodesic path, ANN search 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, the causality of the propagation can also benefit from the isotropicity of the regular grid. The propagation process is illustrated in Figure 3.11.

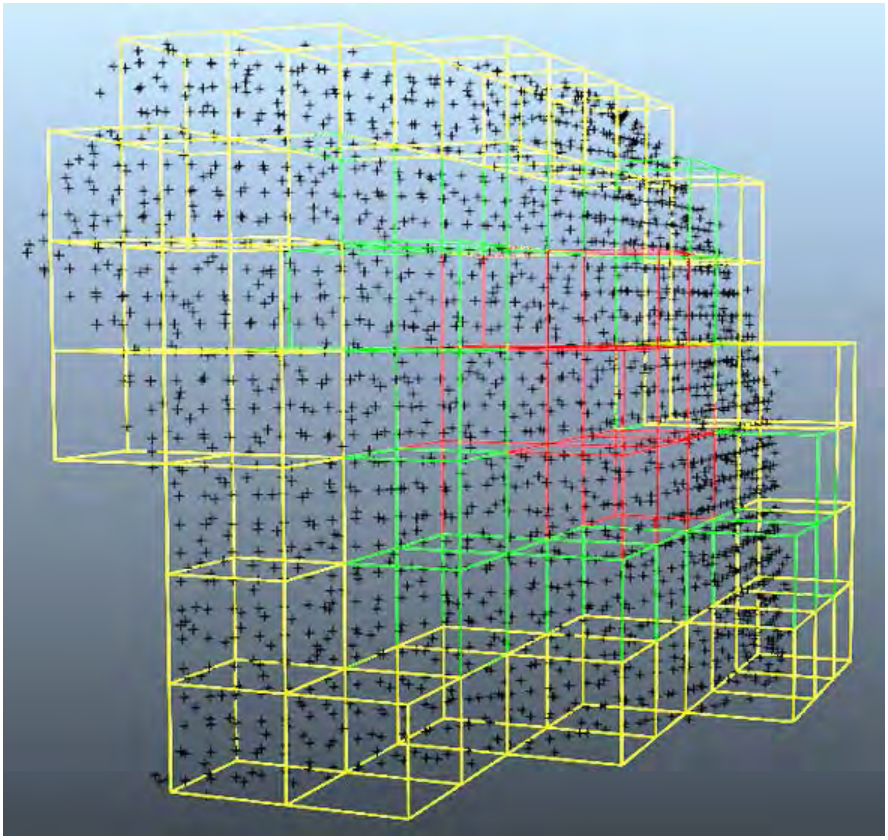
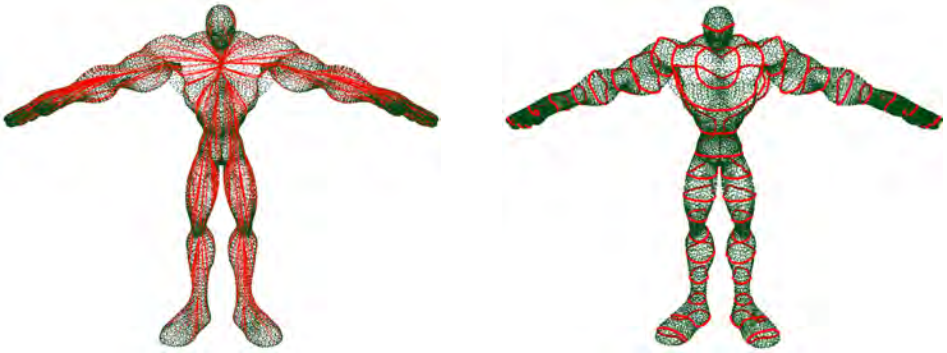


Figure 3.11: *Regular grid covering a point cloud, the propagation is from the red cells to the yellow ones.*



(a) **(b)**

Figure 3.12: *Geodeiscs(a) and isolines(b) on the point cloud model of Character A*

3.5 Performance Analysis

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

3.5.1 Efficiency

The time complexity of Algorithm 2 depends on the number of the sample points on a geodesic. Because the coefficient matrix A on the left hand side of Equation 3.16 is not a square matrix, it is necessary to left-multiply A^T on both side of the Equation 3.16 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 3.16, according to Bunch & Hopcroft (1974); Coppersmith & Winograd (1987), the LU solver has a time complexity of $O(m^{2.379})$. Therefore, for a single source to all destination geodesic computation, the upper bound of the time complexity can be estimated by $O(m^{2.379}n)$, where n

denotes the number of vertices on mesh. Additionally, the computation of the projection of each three successive sample points of a geodesic onto the mesh faces costs a constant time, denotes at h, the projection of the entire geodesic 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 3.16 is constant. Therefore, solving Equation 3.16 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 Algorithm 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 author's project page[1]. 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 3.13. It can be seen that the Algorithm 3 shows great advantage over others when the resolution of the model increases.

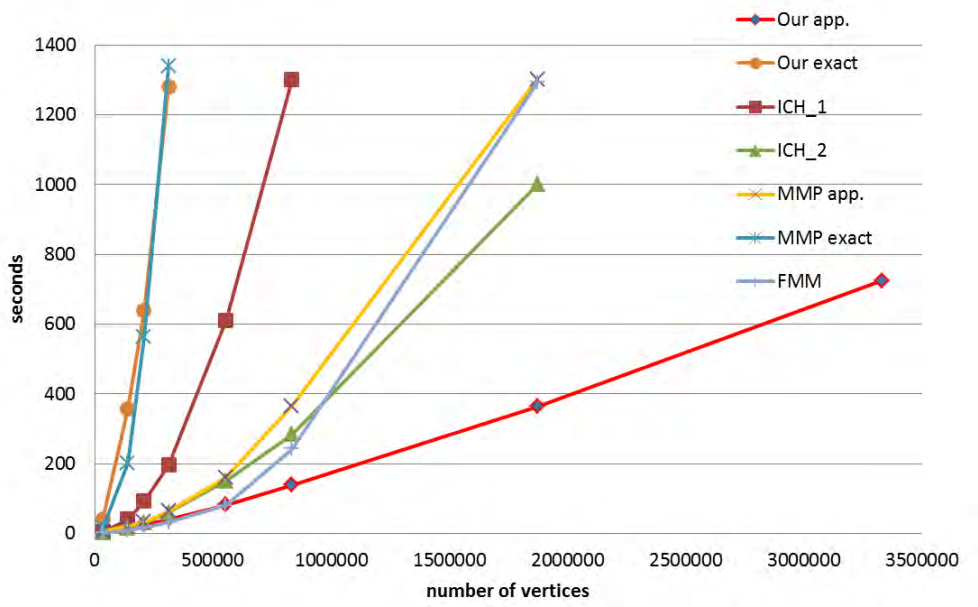


Figure 3.13: *The comparison of running times. The times of the ICH and MMP algorithms include the cost of "backtracing", the window size of our Algorithm 3, w=30.*

#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 3.13*

[1]MMP 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`

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.

3.5.2 Accuracy

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 Algorithm 2 is done by minimizing the geodesic curvature on the smooth surface that the polyhedral surface is approximated from, therefore when Equation 3.11 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 spherical vertex on the polyhedron, the projected curve is a straight and shortest geodesic on the polyhedron. Based on our experiment, none of the geodesic passes a spherical vertex, therefore, in this thesis, Algorithm 2 can be denoted as "accurate geodesic algorithm".

The accuracy of Algorithm 3 depends on the following parameters, the edge length e, the window size w, and the number of the sample points on each geodesic, denoted by 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 3.11 to this patch instead of the whole geodesic.

Let C_g denotes a geodesic on the polyhedral surface that does not pass through any vertex p of the polyhedron unless p is the source point or the destination. It can be observed that the C_g goes across a set of faces. Unfolding this set of faces into a plane, C_g should becomes a straight line linking the source point p_s to the destination point p_d. For a new vertex q, the geodesic $\widetilde{p_sq}$ can be estimated by combining the edge $\overline{p_dq}$ with C_g. Without losing the generality, let a window covers a section of $\widetilde{p_sq}$ from q, w denotes the number of the sampling points covered by this window. The error estimation for any geodesic computed by Algorithm 3 is illustrated in Figure 3.14,

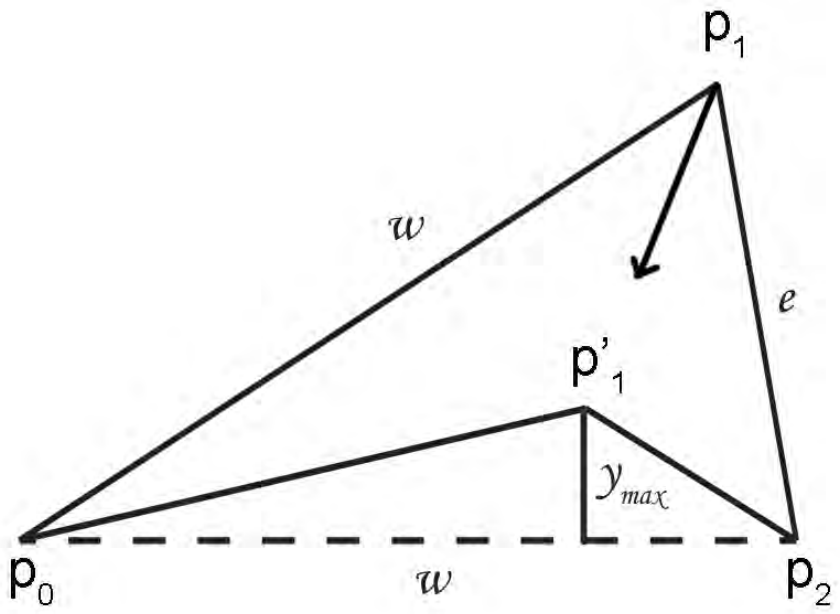


Figure 3.14: *The estimation and desired geodesics, the $\overline{p_0p_2}$ is the desired geodesic while $\overline{p_0p_1}$ plus $\overline{p_1p_2}$ being the estimation.*

Let p_0, p_1,p_2 are three points on a plane as shown in Figure 3.14. The initial path for solving Equation 3.11 is $\widetilde{p_0p_1p_2}$ where p_0 is the source point and p_2 is the destination point. Obliviously, 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 than or equals to the length of $\overline{p_0p_2}$ because the propagation only performs on outward direction. Therefore, in

the worst situation, see Figure 3.14, $\overline{p_0p_2}$ and $\overline{p_0p_1}$ are equal. p_1' is the updated point calculated by Equation 3.11, Let the distance from p_1' to edge $\overline{p_0p_2}$ is y_{max}, $\|p_1p_2\| = e$, $\|p_0p_1\| = \|p_0p_2\| = w$, $\|p_0p_1'\| + \|p_1'p_2\| = L$. Therefore, the based on the Heron's formula, the area of $\triangle p_0p_1'p_2$ can be written as,

$$T_{\triangle p_0p_1'p_2} = \frac{\sqrt{(L^2 - w^2)w^2}}{4} = \frac{wy_{max}}{2} \tag{3.20}$$

therefore,

$$L = \sqrt{4y_{max}^2 + w^2} \tag{3.21}$$

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

$$err = L - \|p_0p_2\| = \sqrt{4y_{max}^2 + w^2} - w \tag{3.22}$$

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

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

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

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

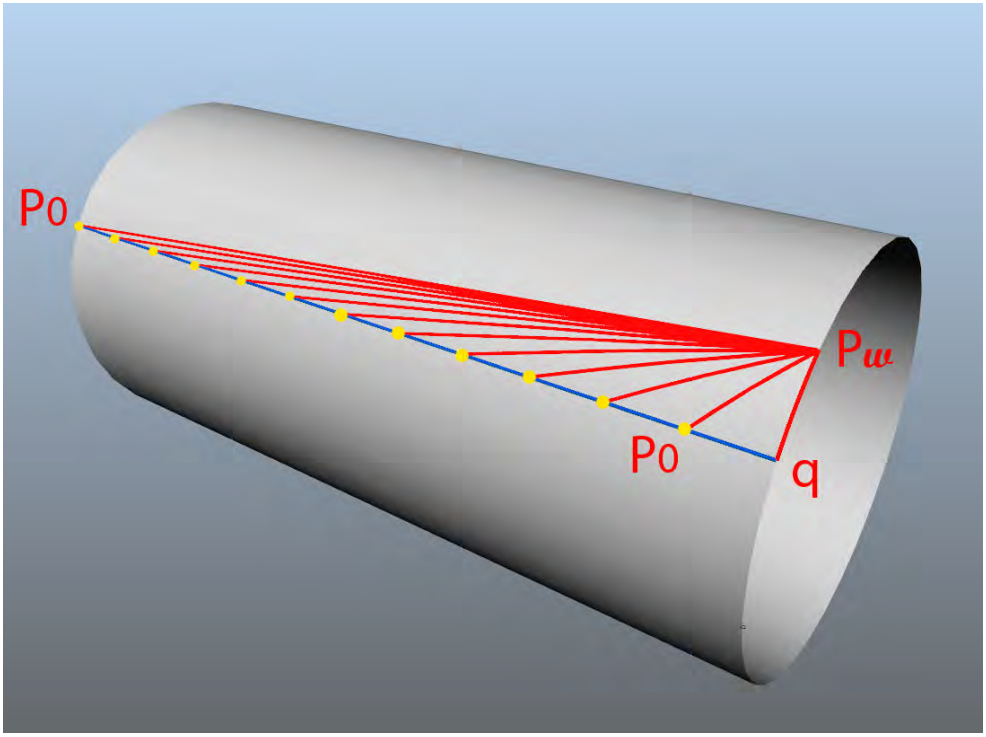
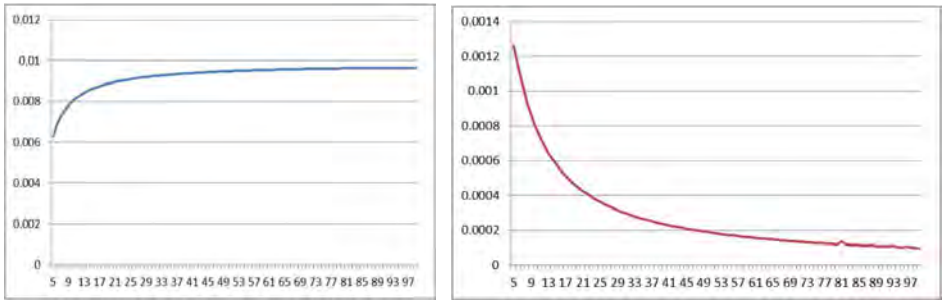


Figure 3.15: *Error estimation of Algorithm 3 with different window size*

where $\varepsilon = \frac{y_{max}}{w} \ll 1$ and y_{max} denotes the maximum offset distance of the sample points to the ground truth. To have an insight into the y_{max}, we performed the Algorithm 3 on a cylinder, see Figure 3.15. $\overline{p_0 q}$ is parallel to the axis of the cylindrical surface, therefore, the true geodesic between p_0 and q is $\overline{p_0 q}$. Let the length of the edge between each yellow point is e = 1 and the window size w varying from 5 to 100, The numerical results are shown in Figure 3.16a. 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 3.14, the curvature of the geodesic estimation $\widetilde{p_0 p_1 p_2}$ is becoming smaller when increasing the window size w. The system of Equation 3.11 is appropriate to deal with high curvature areas. On the other hand, the window size w is usually expected to be as large as possible. In Figure 3.15, 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



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

(b) *The convergence of ϵ with varying window size*

Figure 3.16: *Error of Algorithm 3*

from the source to the p_0. Thus, it is natural to enlarge the window size as much as possible. Figure 3.16a and Figure 3.16b shows that both the y_{max} and the ratio ϵ are converged into 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 3.11 rather than the computational error.

Characters		A #V:21250 #F:42946	B #V:15368 #F:30644	C #V:15266 #F:30440	D #V:14876 #F:29704
Algorithm 3 (floating)	ave abs	0.00309	0.00098	0.00786	0.005218
	ave rel	0.5054%	0.1911%	1.3454%	0.4852%
Algorithm 3 (projected)	ave abs	0.00401	0.00759	0.00716	0.01095
	ave rel	1.2298%	1.0439%	1.3053%	1.4867%
MMP app.	ave abs	0.00416	0.00991	0.00752	0.00917
	ave rel	1.4183%	1.3665%	1.3283%	1.0886%
ICH_2	ave abs	0.00344	0.003751	0.00795	0.01198
	ave rel	0.7867%	0.5886%	1.3311%	1.5128%

Table 3.2: *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 four characters.

MMP exact algorithm is performed on these four characters and its solutions are considered as the ground truth for the this experiment. The absolute errors is the differences between the exact geodesic distances and approximate ones and the relative errors is the ratios of the absolute errors over the exact distances.

Note that, for Algorithm 3, two results are kept as one is the offset geodesics and the other is the projections of the offset geodesics onto the mesh. Table 3.2 gives the average absolute and average relative error of these three algorithms. It can be noted that the offset solution of Algorithm 3 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 comparing to MMP approximation and ICH_2 algorithms.

Moreover, in order to further validate the error estimation for Algorithm 3, MMP exact algorithm is performed on the lowest resolution bunny model in Figure 3.1 and length of the geodesics result from MMP exact algorithm is used as the ground truth.

The ratio ε in Equation 3.24 is usually a very small number. Figure 3.17 shows the histogram of the ratio ε over the bunny model, where the horizontal axis indicates the value of ε and vertical axis indicates the number of sample points in a geodesic. Figure 3.17 indicates that on the bunny model, for all the geodesics, ε remains a small value.

Figure 3.18 further shows the distribution of the real absolute errors and the estimated ones in the bunny model test. It can be noted that the error estimation of Equation 3.24 can accurately reflects the upper error bound of Algorithm 3

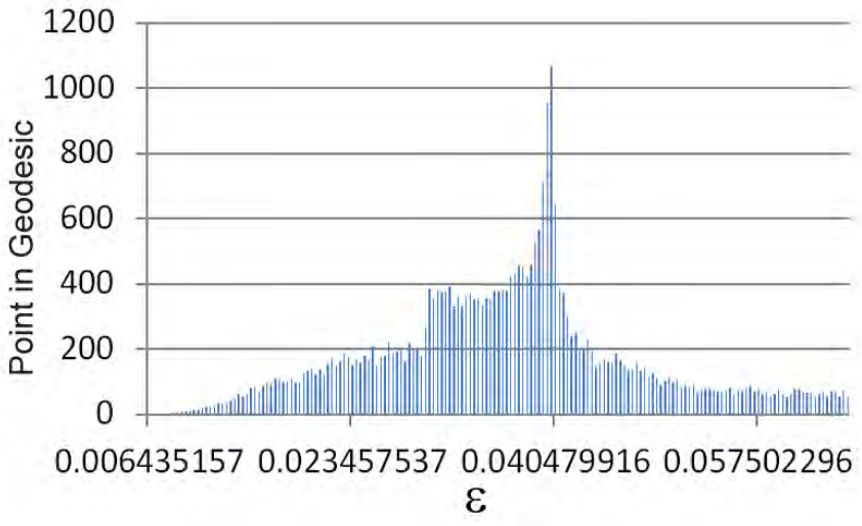


Figure 3.17: *Histogram of the ratio ε on bunny model*

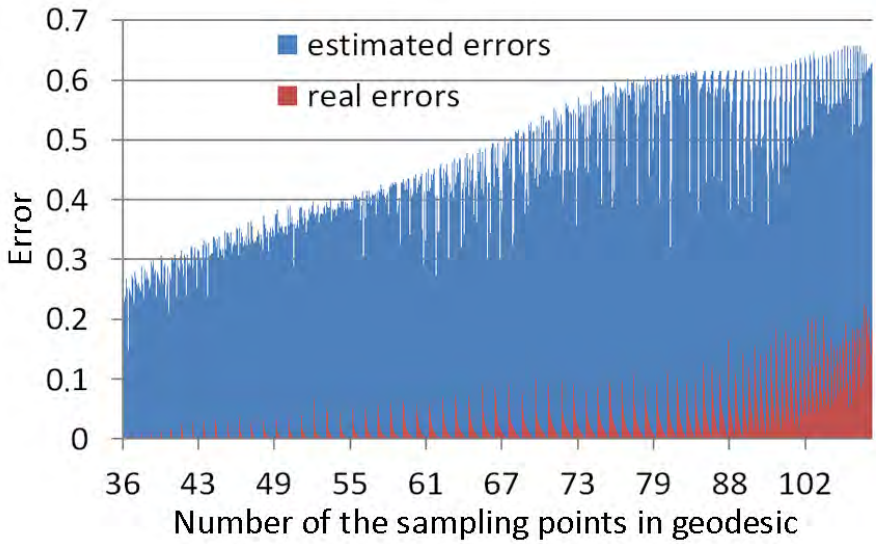


Figure 3.18: *The distribution of the estimated errors and real errors, the window size of our Algorithm 3, w = 30, the real error is bounded by teh estimated error for all the geodesics on the bunny model.*

Algorithm 5 is further preformed on three point cloud models, which are, Stanford bunny, Buddha and sculpture, respectively for geodesic computation. The experimental results are shown in Table 3.3. Although the ANN search 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 com-

plexities in practice. Figure 3.19 shows the shortest paths and isolines on the point clouds in Table 3.3.

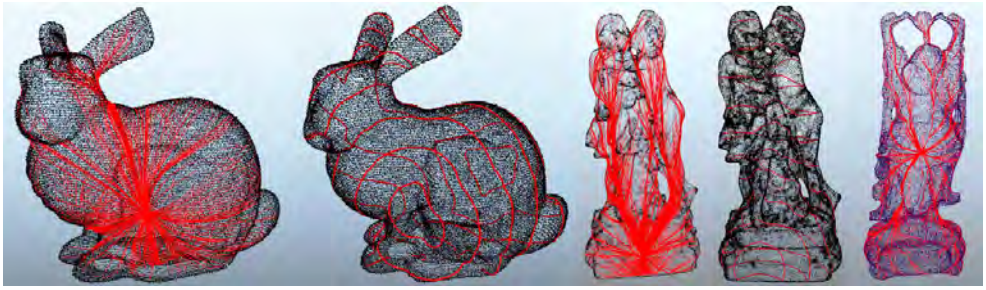
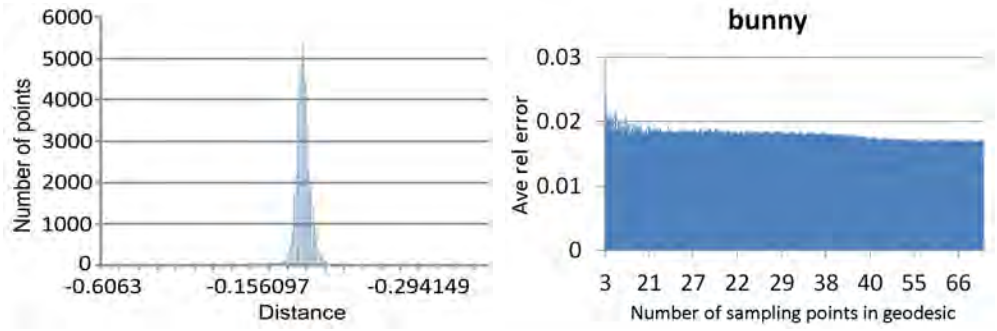


Figure 3.19: *Models used for the test in Table 3.3*

	Character A	Bunny	Sculpture	Buddha
#Points	134175	69668	406224	841151
#Destinations	7904	3748	10878	30177
Time(sec)	53.677	32.924	73.019	226.772
Memory(Mb)*	17.381	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) **(b)**

Figure 3.20: *Accuracy of Algorithm 5. (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 the Algorithm 5 on bunny.*

The accuracy of the resulting geodesic paths can be evaluated by the

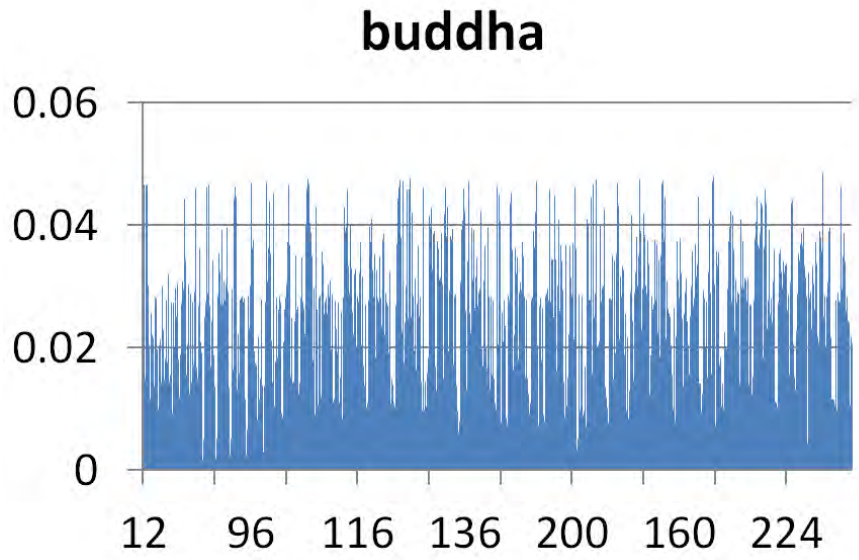


Figure 3.21: *The distribution of the average relative errors of Algorithm 5 on buddha. The x-axis indicates the number of the sample point in a geodesic path*

offset distances of the sample points on geodesics to the implicit surface of the point cloud model. Figure 3.20b shows the histogram of the offset distances of the sample points to the reconstructed implicit surface of the bunny model.

According to the Figure 3.20b, Algorithm 5 can guarantees the resulting geodesic paths close to the implicit surfaces of the point cloud model. 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 3.20b and Figure 3.21 show the distribution of the average relative errors of the Algorithm 5 on the two point cloud models. It can be noted that the error on Buddha is obviously higher than that on Bunny. This is due to the fact that the Buddha model has more details than the Bunny model. The regular grid tends to ignores the fine detail on the model. Further increasing the number of cells in the grid can decrease this error.

3.6 Geodesic in body measurement

During the process of measuring, two types of measurements are used for describing the body dimension of the character, 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 lists out the major measurements need to be extracted from the character model. Figure 3.22 illustrates the datum points for general measuring on a female figure.

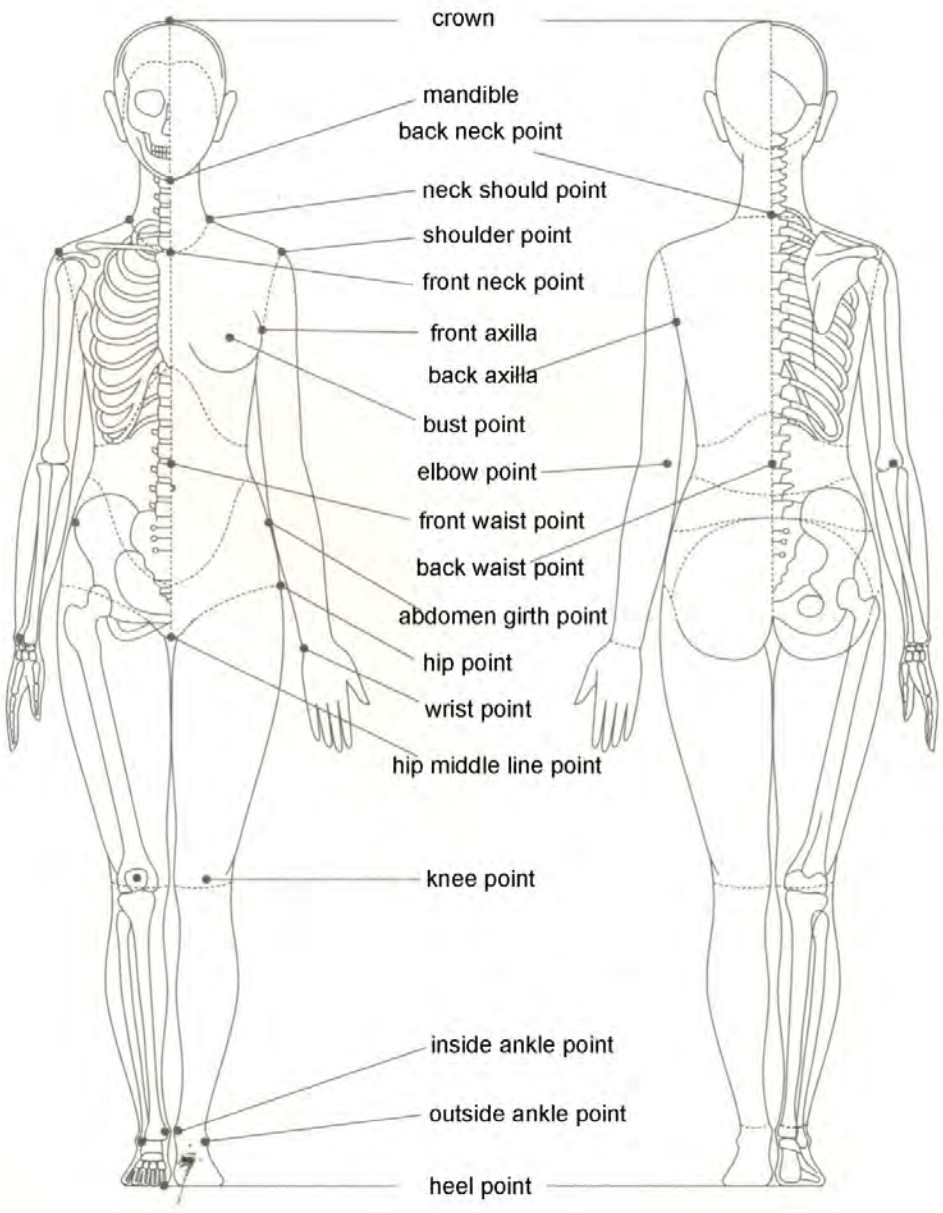


Figure 3.22: *Datum point on a female body(Xiong 2008)*

In this thesis, the circumference is measured by applying convex hull algorithm(Graham 1972; De Berg et al. 2008) to the cross-section of the body part, the arc length of the convex hull is the circumference of the body part. Figure 3.23 demonstrates two circumference measurements on the character.

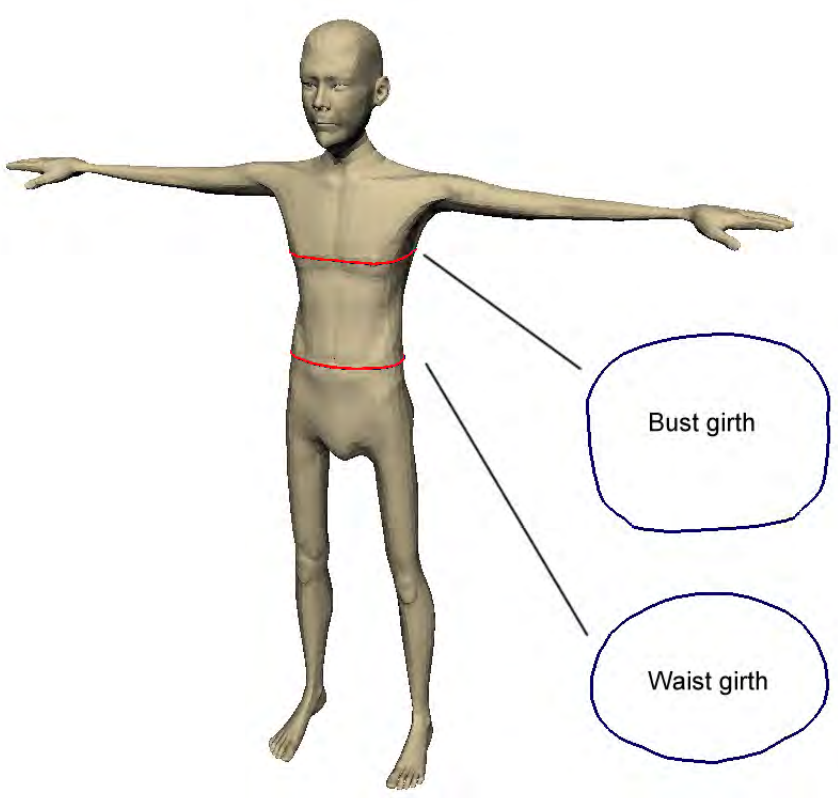


Figure 3.23: *Two circumference measurements on character, two blue close curves indicate the convex hull of its corresponding measurements*

For length measurement, the geodesic algorithm presented in this chapter is applied to the character body. With each measurement, two datum points that are associated with the measurement are used as the source point and the destination point for the geodesic algorithm respectively. Figure 3.24 demonstrates two length measurements on the character. Table 3.5 lists out all the measurements used for cloth modelling in the experiment presented in the next chapter.

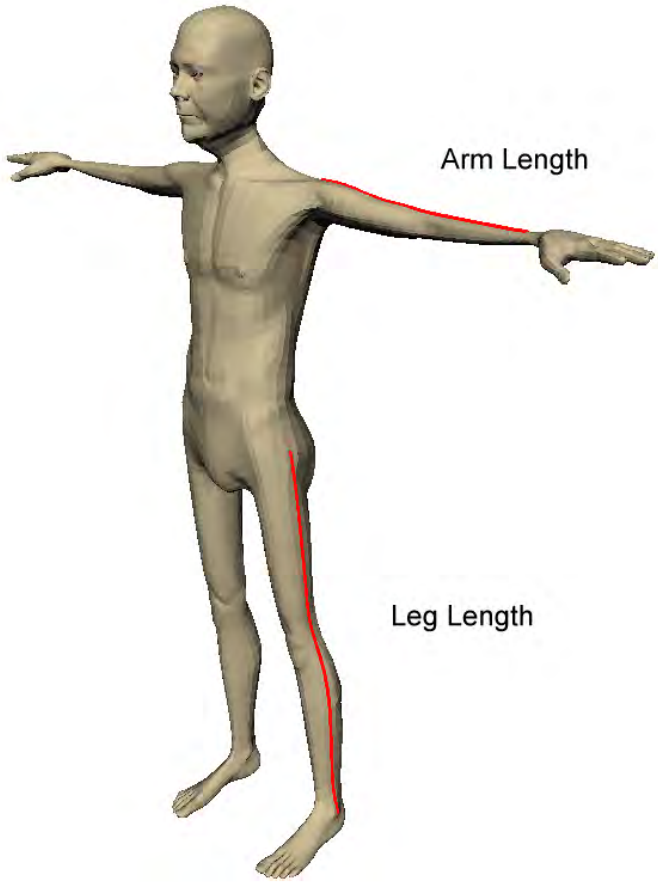


Figure 3.24: *Two length measurements on character, the "Arm Length" is the length of the geodesic path from "Shoulder Point" to "Wrist Point", the "Leg Length" is the length of the geodesic path from "Hip Point" to "Outside Ankle Point", for the datum points, see Figure 3.22.*

	Character A	Character B	Character C	Character D
Bust Girth	211.9015	79.0833	111.9165	95.8160
Waist Girth	73.8146	70.2424	96.6514	70.7769
Hip Girth	83.2698	87.2813	107.5671	73.1573
Neck Girth	67.3815	43.6637	68.2416	59.6871
Cuff Girth	59.3813	8.3771	15.3368	14.5720
Shoulder width	89.9452	40.0348	51.3362	34.2857
Back width	74.8823	38.6953	50.3038	32.2461
Front width	88.7650	45.7562	59.4946	38.1302
Sleeve length	122.8928	57.3615	58.3478	43.0211
Arm hole	157.3058	29.2281	42.9175	37.9510
Back length	77.6125	44.3503	45.1839	36.4137
Outside Leg length	129.1429	89.1428	85.0919	51.4826
Inside Leg length	109.7143	74.2857	71.4285	41.3055
Thigh Girth	65.5714	33.7142	57.1428	38.5871
Ankle Girth	47.4285	16.5714	20.0174	21.7142

Table 3.5: *Measurements for all the characters.*

3.7 Conclusions

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

the real world, it also results less variation when the posture of the character changes.

This chapter proposed three algorithms respectively for accurate and approximate geodesic computation on triangulated manifolds as well as approximate geodesic computation on point cloud data set. The most important contribution is that the proposed approximation algorithm can reach linear time complexity with a bounded error on triangulated manifolds. Numerical comparisons with existing algorithms (i.e. MMP, ICH_1 and ICH_2) have further demonstrated the advantages of our algorithms in terms of both speed and accuracy. By integrating Algorithm 3 into the measuring system, the time consumed for solving the geodesic path between multiple pairs of source and destination has been largely reduced, at the meantime the accuracy of the solution is maintained.

Chapter 4

Virtual Cloth modelling and Re-targeting

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

4.1 Introduction

The construction of a virtual cloth is a time consuming process which usually takes a well trained animation artist several hours to model a complete set of outfit for a character. Normally, modelling a cloth from cloth patterns involves three steps, firstly, the 2D cloth patterns have to be created, then patterns are placed onto the 3D character model, finally, physical simulation is performed to stitch all the patterns together. If the cloth needs to be adjusted, the modifications are carried out on 2D cloth patterns and this entire process

needs to be performed iteratively till the cloth fits to the character. Moreover, if the same cloth design needs to be dressed onto a different character who has different body shapes and proportions, this tedious process would needs to be repeated again from scratch. To make matter worse, adjusting cloth pattern to fit to a character requires lots of tailoring expertises, few animation artist possess such knowledge. Therefore, current pattern based cloth modelling methods are rarely used in the production of the films or games.

When using geometrical modelling method to create cloth, that is, to model cloth patch by patch(polygon, nurbs or subdivision). This approach seems easier than pattern based modelling technique because the cloth can be modelled directly onto the character and the appearance of the cloth can be viewed throughout the modelling process, therefore the fit of the cloth is guaranteed. However, this advantage only exists when modelling cloth for one character, the geometrical modelling method also suffers from the same drawback as the pattern based cloth modelling method when handling multiple characters. That is the modelling process needs to be repeated from scratch to fit a cloth onto different characters with different body shapes and proportions.

Followed by the fast development of computer hardware and computer graphic techniques, more and more characters can be handled simultaneously. However, limited by current cloth modelling techniques, dressing different characters with different body shapes and proportions is still considered as a very tedious and time consuming process.

In the fashion industry, after generations of development, cloth pattern has become an important medium that transfers cloth design into wearable object. Cloth patterns are the basic items that consist a functional cloth. In order to implement a cloth design, firstly, the design needs to be broken down into a set of textile pieces, then through the assembling techniques such as stitch and adhesive, all the pieces are assembled together in certain order to

form a complete cloth. The process that breaks down a cloth design into patterns is called “patternmaking” (Armstrong 2000). Through this process, cloth design is transferred from design concept into a set of patterns that are can be cut from the raw textile material. Therefore, the same cloth design can be reproduced easily(Hannah 1919).

Modern cloth pattern is designed to follow the measurements of the ideal figure to maximize the generality. By providing the measurements of a specific customer, the shape of each pattern is adjusted respectively to fit the customer. In the massive cloth production, a complex pattern resizing procedure named as pattern grading is developed to adapt the needs for dressing general public. Cloth pattern grading aims at resizing the cloth into several size categories based on the human anthropometric statistic data to fit the most individuals in the general public(Moore et al. 2001).

In film or game industry, characters are usually designed into an exaggerated manner. Characters usually have a very different body proportions. Therefore, the human anthropometric statistic data no longer applies to the virtual character, the pattern grading technique can not be used for resizing the cloth pattern to fit to different characters. The cloth for each character needs to be modelled individually. The tediousness and unintuitiveness of the current pattern based cloth modelling methods are getting worse when dressing multiple different characters with different body shapes and proportions. However, many methods such as Brouet et al. (2012); Ebert et al. (2002) was developed for dressing different characters automatically, their methods fit a cloth to a new character by adjusting 3D cloth mesh directly. The resulting cloth patterns are extracted after the cloth adjustment by using 3D surface flattening techniques. Despite the extra computation required by this method, the pattern shape distortion introduced by 3D surface flattening process can not be avoided. Therefore, the preservation of the cloth pattern can be hardly maintained.

4.1.1 Cloth Patternmaking

Armstrong (2000) defines the cloth patterns as the templates from which the parts of a garment are traced onto the fabric before cutting out and assembled. When creating patterns, blocks(also as known as slopers) are created first. Blocks are two-dimensional templates that consist the basic design of the a cloth type(Howland 2008), see Figure 4.2. Blocks are constructed based on the measurements taken from the wearer(Armstrong 2000). 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 cloth. Therefore, this process is usually performed by an experienced tailor. In the massive cloth production, the blocks of 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 shares the similar body proportion. For example, US standard clothing size chart(ISO/TR-10652 1991) contains the body proportions of the general public of the Americans and European standard clothing size chart(EN:13402 2001) contains the size data of the people who were born in Europe. Based on the blocks, cloth patterns are created by introducing pockets, style line, drapes and other adjustments. Figure 4.1 demonstrate the blocks for a woman's shirt.

Normally, only the standard measurements are used for block construction. Different cloth sizes are created by pattern grading process. Pattern grading is the process of systematically increasing and decreasing the dimensions of the patterns in to a range of sizes for production. In fashion industry, only one size patterns are developed for a cloth design, then other sizes of pattern are graded from these patterns. During pattern grading, patterns that derived from blocks are scaled proportionally into different sizes with predefined intervals introduced by a size chart. This process not only retains the original design of the cloth during the distribution but also very

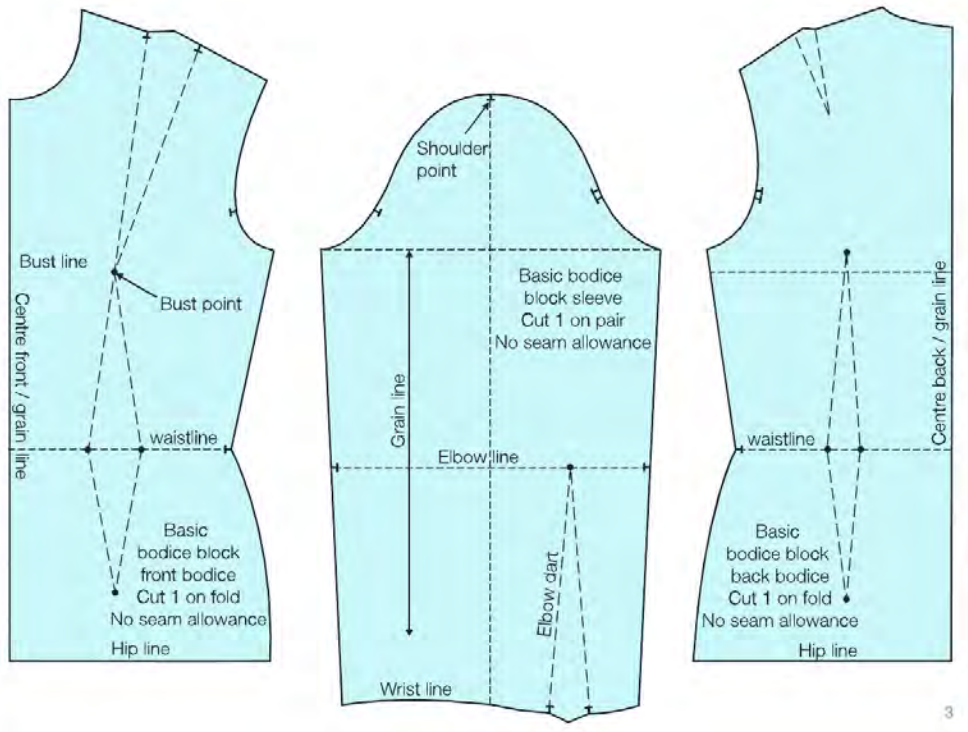


Figure 4.1: *Garment block for woman shirt(Rosen 2004).*

cost effective during the process of manufacturing due to the people who fit a predefined size are normal distributed into an interval introduced by a size chart(Schofield & LaBat 2005; Moore et al. 2001).

However, in the computer animation the situation is much complex from what it is in reality. In reality, for each ethic or certain group of people, the similarity of the body proportion exists among individuals. In virtual world, the body proportion is only limited by the imagination of the artist, therefore using the size chart to perform proportional scaling onto patterns to create fit cloth no longer applies to the virtual character. Nonetheless, as shown in Figure 4.1, on each block, there are several important landmarks that associate with the datum points on the body of character. By matching the distance between the landmarks on the block with their associated measurements of the character, the cloth pattern can be fit to a character with any body proportion.

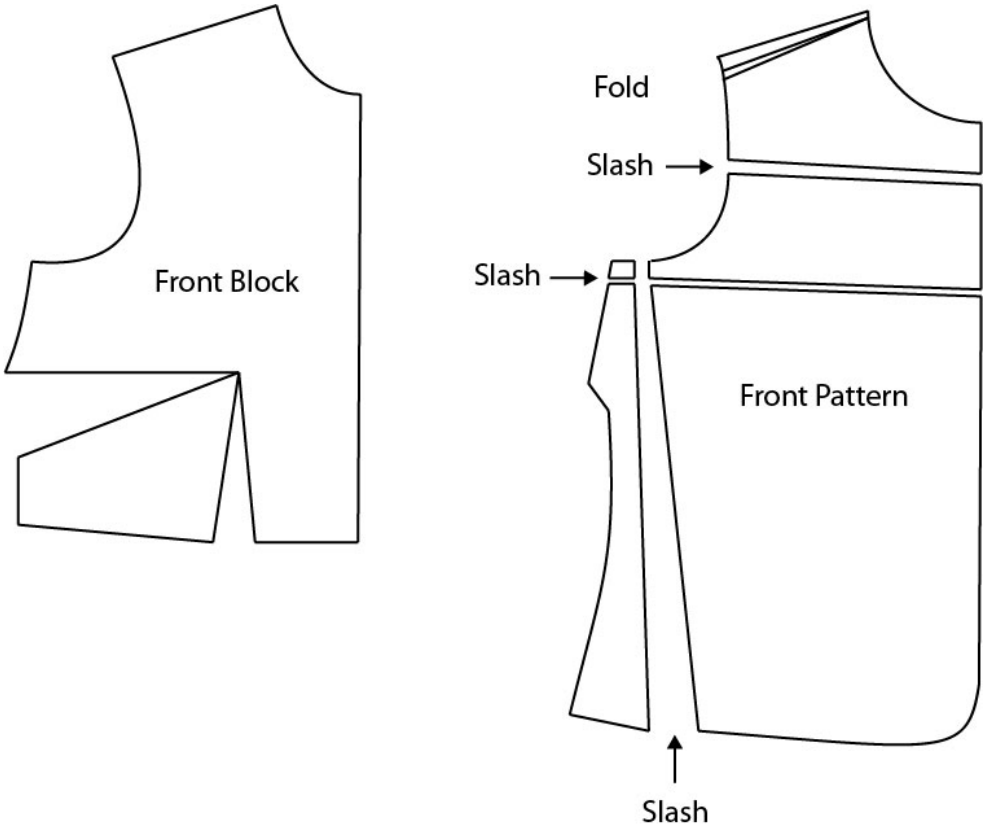


Figure 4.2: *Block(left) and a pattern(right) made from it by adding details(Howland 2008).*

4.1.2 Pattern Resizing Criteria

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

Figure 4.3 indicates that the same shirt design to be dressed onto two characters with same height and limb length by using proportional scaling method presented by Moore et al. (2001). Because the length of their corresponding body parts are similar, both of the character falls into the same

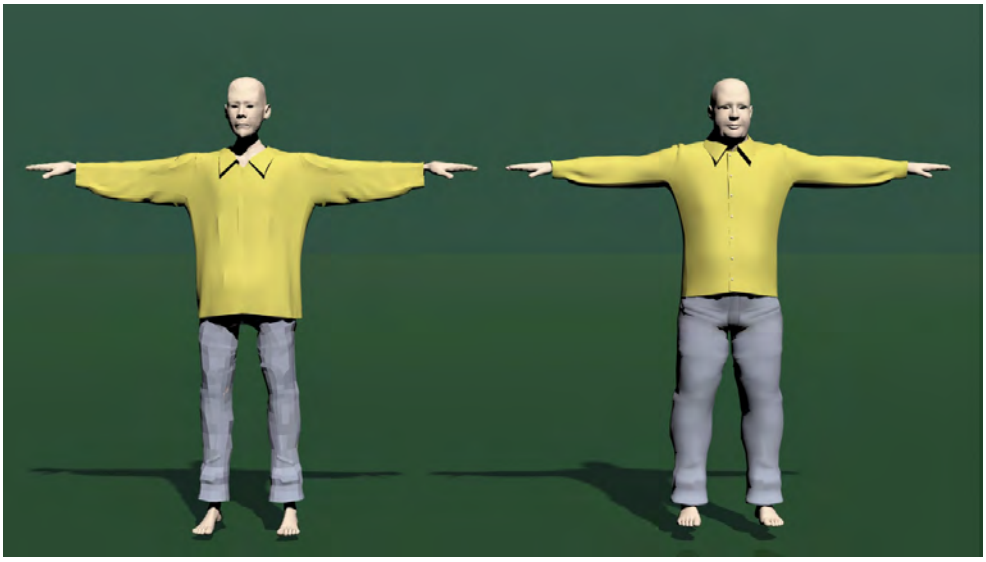


Figure 4.3: *Same shirt design proportional scaled onto two characters with similar body proportion*

grading intervals. However, the character on the right side is much stronger than the one on the left, his body supports the cloth much better than the other. On the contrary, the skinny character has much thinner body which results a baggy shirt. During made-to-measure cloth tailoring, extra measurements such as "Arm Hole Length" or "Cuff Girth" are taken from the customer in order to fine tune the cloth for better fit. With those extra measurements, different parts of a pattern might to be scaled in different manners. Performing the non-proportional scaling to the pattern is the biggest different between bespoke cloth making and massive cloth production.

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

of the cloth.

Character Measurements Cloth pattern represents different parts of a cloth which correspond to different parts of the character body, therefore, all the patterns need to be deformed to a manner that each pattern matches to the measurement data of its corresponding body part. Because all the patterns are derived from blocks, during the pattern adjustment, for each block that a pattern is developed from, the distance between all the landmarks that are associated with the datum points on the character body need to satisfy the requirement set by the measurements taken from the character. Especially when the character has an unusual body proportion, different parts of a block might scaled in different fashions. For instance, in Figure 4.3, the arm of both two characters have the same length, however, the circumference of the arm are mush different, Moreover, despite the difference between the circumference of the arms, the wrist girth are the same for both characters. Therefore, in order to satisfy the all the measurement requirements, different part of sleeve pattern need to be adjusted in different manners.

Pattern shape Cloth pattern is the most basic component that consists a complete garment and the shape of each pattern defines the final shape of the cloth. The pattern grading process is considered as the most difficult process during the tailoring in which only the professionals is able to master. The overall shape of the cloth patterns can be altered significantly to accommodate the difference of body proportions between different customers. 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 a different character without altering the design of the cloth. However, each cloth pattern needs to be deformed in order to match the measurements of the character. Therefore a shape

evaluation process need to be performed after adjusting of the cloth pattern in order to maintain the design of the cloth.

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

4.2 Cloth Resizing Algorithm

In this section, an automatic cloth pattern adjusting method is explained in detail. This method operates directly on 2D patterns and optimizes each pattern to ensure all the patterns satisfy the criteria introduced in previous section.

Two inputs are required for this method. The first input is a group of patterns representing a cloth design. Within each pattern, the landmarks that associates with the datum point on the character body are defined. Moreover, the seam-lines are presented along with the pattern. Note that the size of the patterns is irrelevant to the size of the final modelled cloth. Figure 4.4 demonstrates an example of the patterns.

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

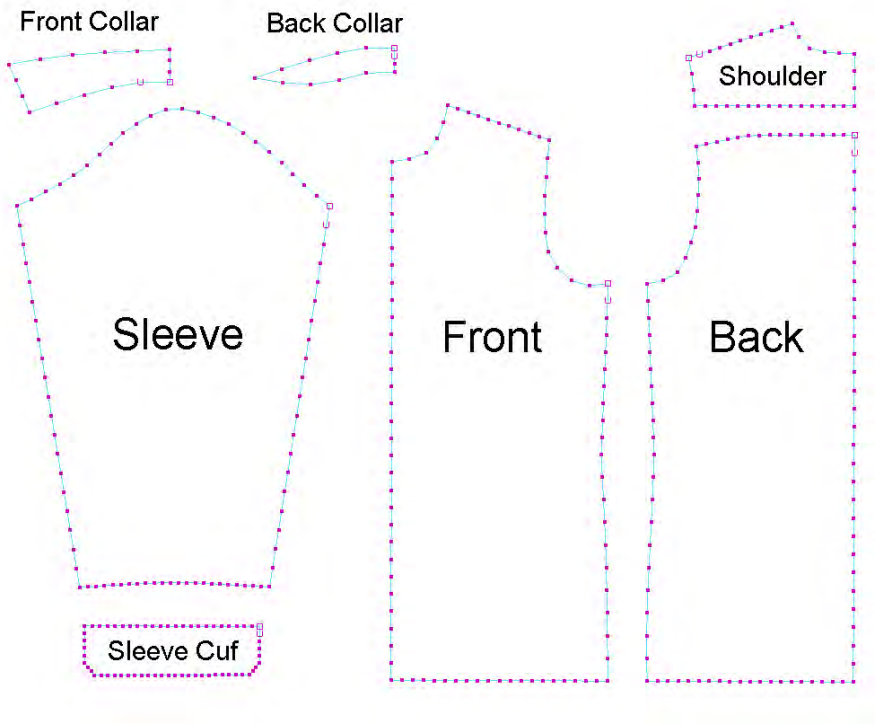


Figure 4.4: *An example of the cloth pattern*

character are also provided by using the method from previous chapter.

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

When constructing a block from a pattern, firstly, the "critical points" of the pattern are selected. The "critical points" are defined by the points which are most capable of representing the geometrical features of a pattern.

For example, the sharp turning points on the contour of the pattern are considered as “critical points”. Figure 4.5 illustrates the “critical points” on the sleeve pattern. Finally, a bounding box is created for each pattern, within the bounding box, for every “critical point”, two orthogonal line are created to form a grid. This is demonstrated in Figure 4.5.

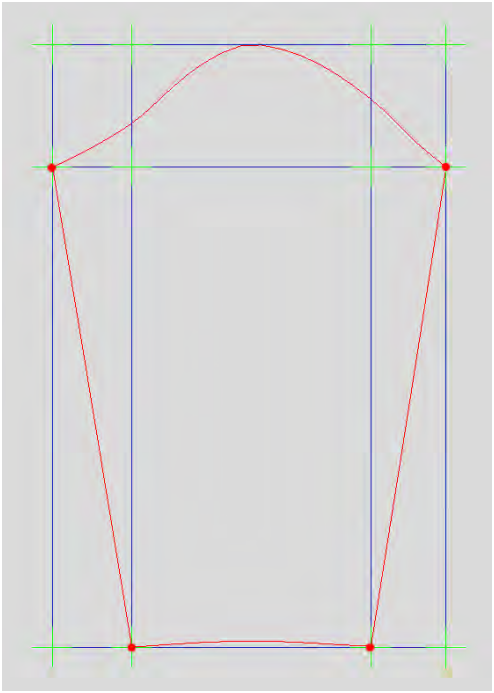


Figure 4.5: *Block generated from sleeve pattern(red closed curve). The red points are “critical points” of sleeve pattern.*

In the next step, based on the gird, a nurbs plane is constructed in which the control points of the nurbs plane are the intersection points on the grid(this is demonstrated by the green cross in Figure 4.5). All the points on each pattern can be represented by a parametric coordinate on the nurbs plane. For every pattern, a unique nurbs plane is created and it is considered as the block of the given pattern.

Then, the proportional scaling used in the traditional cloth grading technique is performed onto each pattern. In general, the measurements that are associated with the length of the limbs are used as the reference of the scaling such as “Height”, “Back Length” and “Arm Length”. Figure 4.6 demonstrates

the result of performing proportional scaling to the sleeve block based on the "Arm Length" measurement.

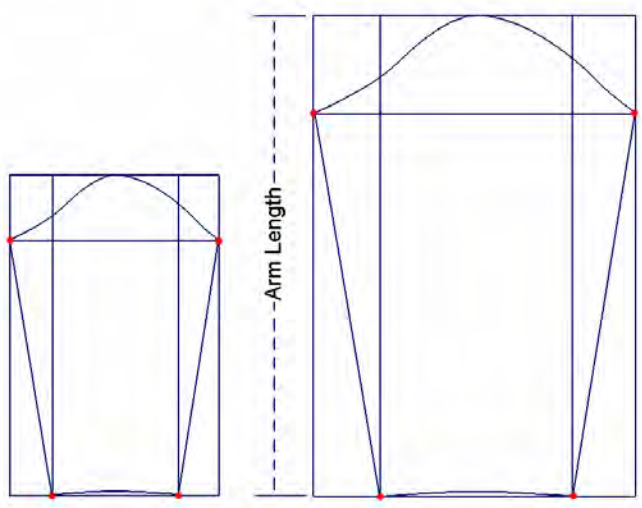


Figure 4.6: *Block before(left) and after(right) the proportional scaling. The height of the sleeve pattern after scaling has reached the measurement "Arm Length"*

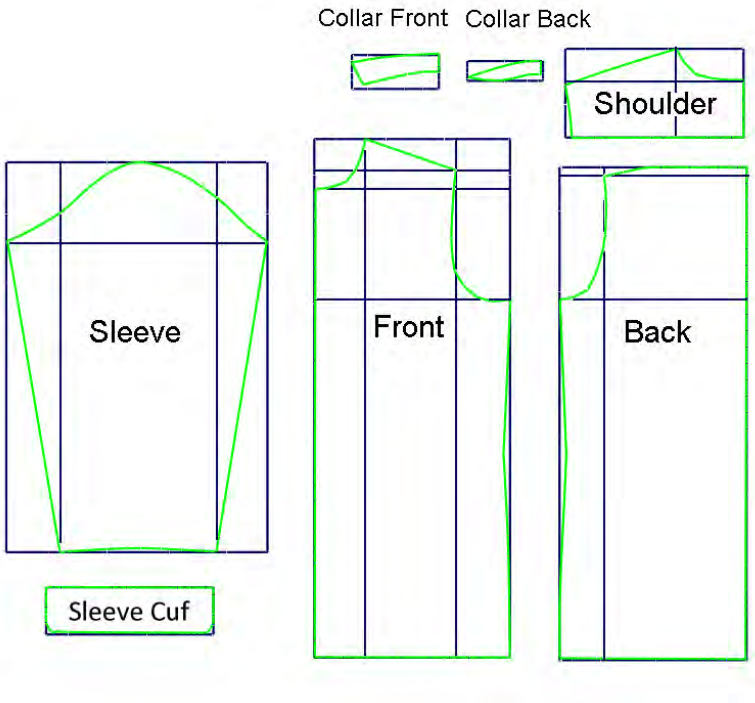


Figure 4.7: *Blocks after proportional scaling is performed.*

The proportional scaling of the patterns 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 a measurement that is directly associated with. Therefore the proportional scaling can not be performed onto such type of cloth pattern. As depicted in Figure 4.7, "Collar Front" and "Collar Back" do not associate with any measurement taken from the character, Therefore, they cannot be resized by the proportional scaling method. Moreover, for such type of patterns, 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 types of proportional scaling are performed to them. This might leads to the 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". Normally, almost every pair of patterns that need to be stitched together have 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 determines 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 connects P_b with P_a. However, the length of the seam-line $S_{a,b}$ can only be determined once the size of P_a is confirmed by M_a. Therefore, if there is another pattern P_c that stitches to both P_a and P_b by $S_{a,c}$ and $S_{b,c}$ and the size of P_c is determined by measurement M_c, the size of P_c is determined by M_c, $S_{a,c}$ and $S_{b,c}$ simultaneously. Because M_c, $S_{a,c}$ and $S_{b,c}$ are defined individually, in many cases these conditions may not concord with each other

and in some cases even conflict to each other. Therefore, it is very difficult to adjust cloth patterns based on one criteria.

4.2.1 Genetic Algorithm

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

Optimization problem is a very hot research area that has been studied intensively for decades and many techniques have been developed to tackle such a problem. In the case of cloth pattern adjustment, since a complete cloth is consisted by a set of patterns, each pattern is formed by a set of points(sampling points on the contour of a pattern). The parameters of this optimization problem are the locations of the sampling points.

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

able to avoid local optimal solution effectively. In general, the advantages of the genetic algorithm over the traditional optimization method are,

1. Genetic algorithm searches through a wide range of the cost surface simultaneously. Therefore it is able to deal with very complex cost surfaces and avoid local minimum.

2. Genetic algorithm works on the chromosome instead of real parameters, therefore it is able to handle large number of parameters.

3. The initial proposals do not effect the final solutions as bad solutions are discarded by selection at every evolution. Therefore, the genetic algorithm is not sensitive to the initial seeds.

Therefore, by introducing genetic algorithm into the cloth adjusting process, this method can generates the best combination of fitted cloth patterns and maintains the original design of the cloth. Because cloth patterns are adjusted based on the measurements of the character, adjusting cloth patterns using genetic algorithm has three extra benefits. Firstly, given the measurements of any character, this method can fits the cloth onto the character automatically. By automating the cloth fitting process, the duplication of effort required by traditional cloth modelling method can be eliminated and the efficiency for modelling cloth for different characters with different body shapes and proportion can be largely improved. Secondly, this method empowers the creativity of animation artists and amplifies their productivity by allowing them to use the large amount of existing cloth patterns in the fashion industry to create various clothes that fit different characters. Thirdly, because the 3D cloth is generated based on the adjusted patterns, the work flow of modelling cloth for the character is one direction. There is no turning back for extracting patterns from 3D cloth which is required by current cloth modelling method. Therefore, the shape distortion that is introduced by 3D surface flattening process can be avoid.

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

A genetic algorithm starts from a group of randomly generated solutions called "initial population". Within the initial population, each individual is a set of variables that represents a solution of the problem. By evaluating every individual using the cost function, a ranking is assigned to the individual in terms of the performance of the cost function. Then, a selection method is applied to select a group of individuals to perform the crossover and mutation. Finally, the convergence of the current generation is evaluated. If the current generation does not reach the minimum of the cost surface, the process goes back to step one, the algorithm is executed iteratively till the minimum cost is reached, this procedure is demonstrated in Figure 4.8

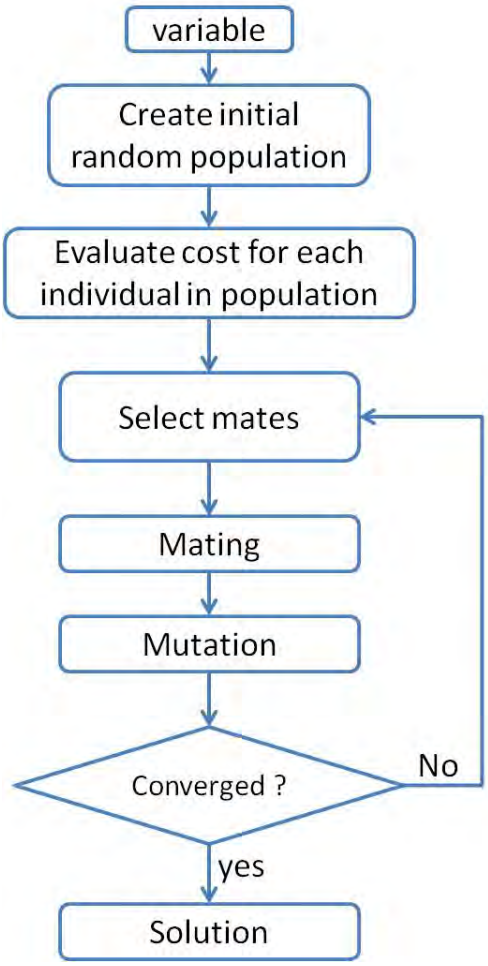


Figure 4.8: *The general flowchart for single objective genetic algorithm*

In genetic algorithm, "fitness" refers to the cost of the individual in terms of its performance to the cost function, higher the fitness, lower the cost. In the single objective optimization, only one individual who has the highest fitness is selected as the final solution of the problem. However, in may applications, more than one objective are needed for describing the problem. Moreover, among the objectives, conflicts often occurs. For example, 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 solution set results from a multi-objective optimization is called Pareto-optimal solutions(Pareto 1906). 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 following section, a multi-objective genetic algorithm for pattern adjustment is presented in detail. This algorithm uses multi-objective optimization to find the best combination of the patterns that best fits the measurements, the seam-line and maintaining the shape of pattern.

4.2.2 Definition of Population

Within a population, individual is the basic element represents one solution to the problem (Haupt & Haupt 2004). In our case, each pattern is described by a set of points located on the contour of the pattern $Pattern_i = [p_0, p_1, ..., p_n]$. A block is a nurbs plane that the pattern is inscribed to and it is defined by a set of control points, noted as $b_i = [cp_{i1}, ...cp_{im}]$. After a block is created based on the a pattern, all the points of the pattern can be interpolated by the parametric coordinate on the block plane. Moreover, each pattern has its unique block, therefore, for each pattern, it can be represented uniquely by the control points of the corresponding block plane. Note that $m << n$, 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 effectively.

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

Table 4.1: *The structure of gene*

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

The initial population is the first sample over the cost surface. Because in the case of pattern adjustment, the measurements of the character can be largely different form character to character, the boundary of the cost surface can be difficult to determine. Therefore, the distribution of the initial population is very important to the efficiency of the algorithm 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, the searching range can be significantly reduced which 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 $M_i = [m_{i_1}, ..., m_{i_n}]$. 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 4.9

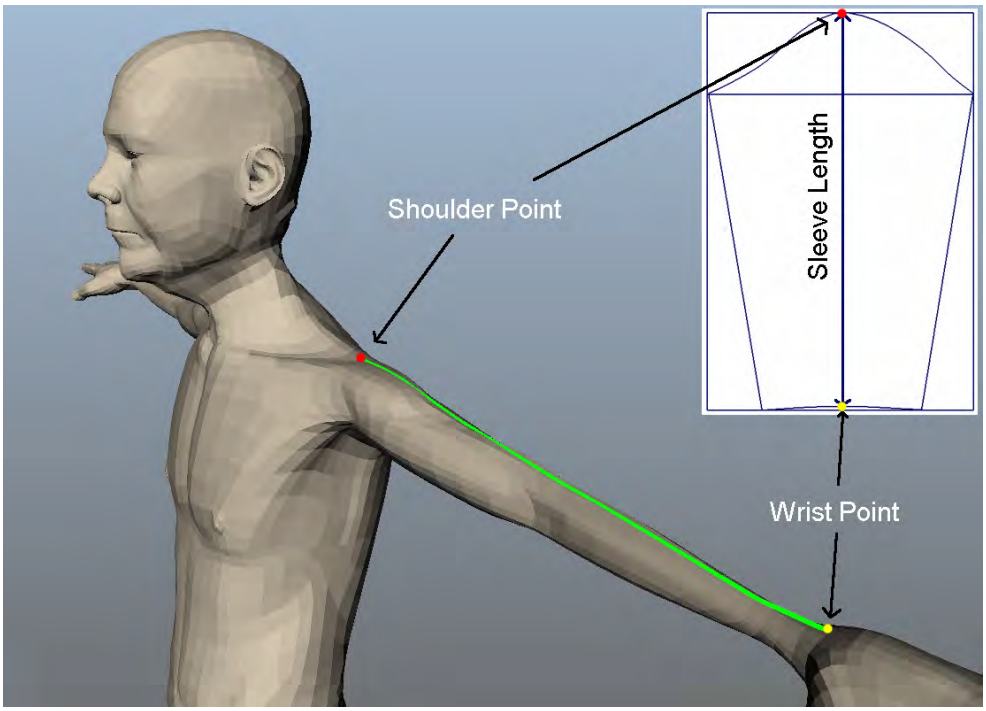


Figure 4.9: *Association between body datum point and patter landmarks. The green line is the geodesic from shoulder point(red point) to wrist point(yellow point). The red and yellow point on the sleeve pattern is the pattern landmarks that associated with the datum points on the body of character. When 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(Pattern P, Measurements M)
2: **for** measurement $m_i \in M$ **do**
3: p_s and p_e is the two datum points of m_i
4: l_s and l_e is two landmarks on the pattern that $l_s \sim p_s$ and $l_e \sim p_e$
5: $d \leftarrow$ distance between l_s and l_e
6: $Sr \leftarrow m_i/d$
7: **end for**
8: Select largest Sc from S as the scaling factor for P
9: **end procedure**

By using Sc as the scaling factor, the input patterns are proportional scaled. Although the proportional scaling method cannot produce a well fit cloth for the character, it is able to resize the patterns to a certain degree so that the cloth 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 normal distributed random number between -1 to 1 is added to every control point of the block.

In order to improve the efficiency of the evolution, the individual can age during the evolution. In nature, the size of the population changes over generations because the individual who carries a fit gene has better crossover opportunity than those who carries 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.

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

Equation 4.2 is a 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 the current individual that carries this gene. $cost_e$ denotes the expected value of the individual and $cost_i$ is the actual cost value for current individual. When a gene reached its life span, it dies before it is evaluated by the cost function. During the evaluation, if lots of individuals died in 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.

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

4.2.3 Crossover and Mutation

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

Crossover is an exploitations to a certain area of the cost surface. It requires two individuals to be involved to the process and normally generates two offspring. In order to perform crossover, two parents need to be selected from the population first, the difference between two parents needs be large enough to create an effective offspring, because two similar parent will re-

sult a very similar offspring and further leads to an over dense sampling in a small region near the parents, which will cause premature convergence. In each generation, the individuals are sorted based on domination. Therefore, when selecting parents from population for crossover, an index interval is used for maintaining the difference between two selected parents. After both parents are selected, then recombination of the chromosomes takes place, this is demonstrated below.

$$Father = [cf_0, cf_1, cf_2, cf_3, ...cf_n]$$

$$Mother = [cm_0, cm_1, cm_2, cm_3, ...cm_n]$$

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

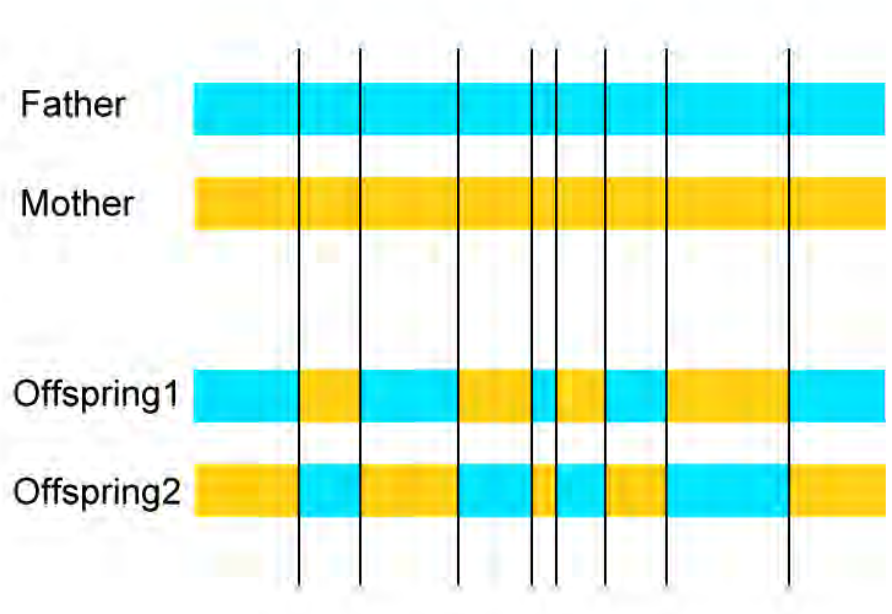


Figure 4.10: *Crossover of parents*

Figure 4.10 demonstrates the uniform crossover method(Spears & Anand

1991; Gwiazda 2006; Ghosh & Tsutsui 2003). This method uses a fixed ratio exchange rate between two parents to generate offspring. The black line indicate the exchange point in the chromosome that are randomly chosen by a random number generator.

For pattern adjustment, each 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.

Algorithm 7 Crossover

procedure CROSSOVER(Two individuals ind_1, ind_2, crossover rate $cxpb$)
  if $cxpb > random()$ **then**
    for same block(b_1,b_2) in ind_1 and ind_2 **do**
      $N \leftarrow$ number of control point in current block
      for $i = 0; i < N; i++$ **do**
        if $rand() < 0.5$ **then**
          $offspring_{1_i} \leftarrow b_{1_i}$
          $offspring_{2_i} \leftarrow b_{2_i}$
        else
          $offspring_{1_i} \leftarrow b_{2_i}$
          $offspring_{2_i} \leftarrow b_{1_i}$
        end if
      end for
    end for
  end if
end procedure

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

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

When performing genetic algorithm, a high mutation rate enlarges the searching range on the cost surface, whilst prevents the population to converge to a specific point. In the meantime, a very low rate of mutation will leads to a premature convergence very easily. According to the work of Yaman & Yilmaz (2010), in different stages of the 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, exploitations is much preferred than exploration so that the convergence can be ensured. At this stage, a high mutation rate will disturb the convergence of the algorithm. In the implementation of this algorithm, the mutation rate starts from 0.6 and decreased linearly by 0.002 for every generation that has evolved. Assume that Gen_{max} is the predefined maximum number of generations of the genetic algorithm. Then, for each individual, the randomly chosen chromosome cp_i is replaces

by one of the two values demonstrated in Equation 4.3,

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

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

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

where, λ is an uniformly distributed random value from 0 to 1, Gen_{max} is the maximum number of generations of the evolution. The exponential factor r controls the influence of gen on the distribution of $\Delta(gen, t)$ within its range. In which this operation becomes an uniform mutation if $r = 0$. The mutation process is introduced in more detail by Algorithm 8

For each mutation, only one orthogonal direction is modified in either row wise or column wise. For example, this mutation operator performs on U direction only for a selected row of the control points or on V direction only for a selected column of the control points. The choice between two directions are randomly chosen for each block. For each row of the block, The mutated U value of each control point is stored into an array denotes as U_{row}. The original U value of every control point 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. Therefore the U value of each row or column of a block is in the same incremental or decremental order. this is demonstrated in Figure 4.11.

Algorithm 8 Mutation

procedure MUTATION(Individuals ind, Generation index gen, Mutation rate $mutpb$)
  if $mutpb > random()$ **then**
    for $block_i \in ind$ **do**
      $dpb \leftarrow random()$
      $U \leftarrow$ the number of control point on U direction
      $V \leftarrow$ the number of control point on V direction
      if $dpb < 0.5$ **then** ▷ Variate cp_i in U direction
        for Each row of cp_j on $block_i$ **do**
          Append u value of cp_j into array $Variables$
        end for
        Apply Equation 4.3 to each element in $Variables$ and evaluate the newly formed $Variables'$
        if $Variables'$ is invalid **then**
          Reapply Equation 4.3 to each element in $Variables$ and evaluate the newly formed $Variables'$ till $Variables'$ is valid
        end if
        for Each row of cp_j on $block_i$ **do**
          Assign back u value of cp_j from corresponding element in $Variables$
        end for
      else ▷ Variate cp_j in V direction
        for Each column of cp_j on $block_i$ **do**
          Append v value of cp_j into array $Variables$
        end for
        Apply Equation 4.3 to each element in $Variables$ and evaluate the newly formed $Variables'$
        if $Variables'$ is invalid **then**
          Reapply Equation 4.3 to each element in $Variables$ and evaluate the newly formed $Variables'$ till $Variables'$ is valid
        end if
        for Each column of cp on $block_i$ **do**
          Assign v value of cp_j with corresponding element in $Variables$
        end for
      end if
    end for
  end if
end procedure

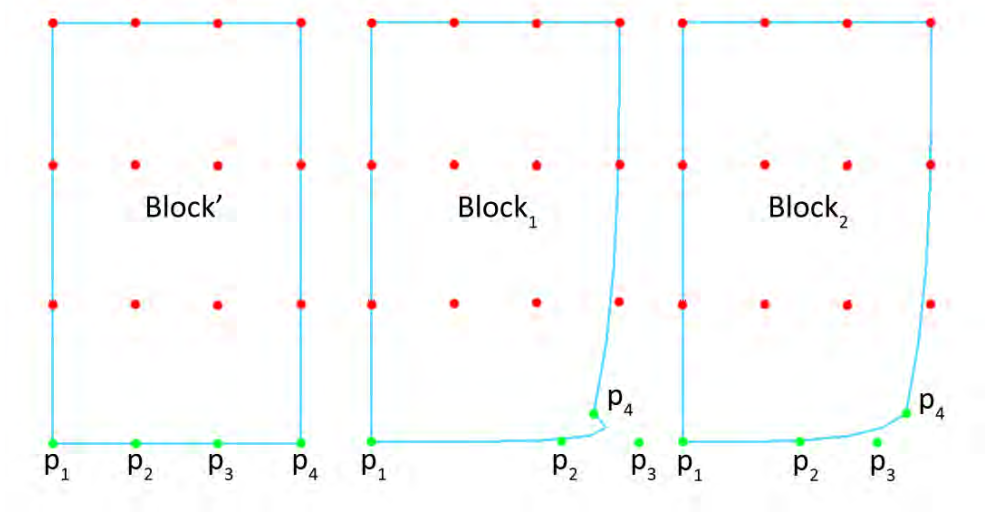


Figure 4.11: *$Block'$ is at its initial status, $Block_1$ and $Block_2$ are the results of mutation operation applied on the bottom row of control points(green points).*

When a block is at its initial status, such as $Block'$ indicated in Figure 4.11, the order of the control points in the selected row is denoted by $R' = [p_1, p_2, p_3, p_4]$. $Block_1$ and $Block_2$ are two possible results generated by mutation operation on $Block'$. Sorted by the U coordinate, the order of the control points in the corresponding row can be written as $R_1 = [p_1, p_2, p_4, p_3]$ and $R_2 = [p_1, p_2, p_3, p_4]$. In order to avoid the overlapping of the face 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 triangulated first to from the polygon from the outline of the pattern. In the case of $Block_1$, because the order in R_1 is changed, therefore, it will cause the face overlapping during the triangulation of the pattern. therefore, the mutation need to be reapplied till $R' = R_1$. In this case, $Block_2$ is a valid gene of the current individual.

4.2.4 Evaluation and Selection

In genetic algorithm, fitness function(cost function) describes the objective of the algorithm. Given a solution, the fitness function is used to measure how far the current solution is to the desired goal of the convergence. When a population evolves into a new generation, the selection operator deletes the n worst individuals and breeds n new individuals from the best individuals. In order to perform such a process, each individual need to be awarded a rank to indicate how close it comes to the 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 are joint together remains. Most importantly the consistency of cloth design must be retained after resizing. Therefore, three fitness functions are developed for evaluating the fitness for each individual respectively.

Measurements evaluation

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

$$Error_m = \frac{\sum_i \|M_n - Dist(l_{i_s}, l_{i_e})\|}{n} \tag{4.5}$$

Where M_i denotes the ith measurements associated with the current block. l_{i_s} and l_{i_e} are two landmarks on the block that are associated with M_i. n denotes the number of the measurements that determines the size of the

current block b_m.

Seam-lines evaluation

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

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

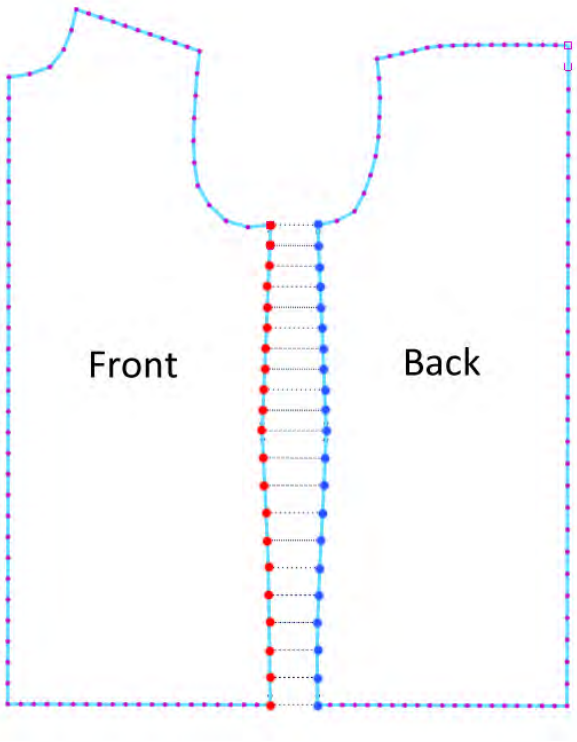


Figure 4.12: *The seam-lines between "Front" pattern and "Back" pattern. The red points indicate the seam-line on the front pattern and blue points are the seam-line on the back patter.*

Let $sline_f$ denotes the seam-line on the "Front" pattern that depicted by the red points and $sline_b$ denotes the seam-line on the "Back" pattern that depicted by the blue points in Figure 4.12. Both points in a point pair that connected by a constrain have the same index in its corresponding seam-line. Given a point pf_i on the $sline_f$, the greatest included angle θ_f between $\overrightarrow{pf_{i-1}pf_i}$ and $\overrightarrow{pf_{i+1}pf_i}$ is recorded. Next, the point at the same location on $sline_b$ is also recorded as θ_b. Therefore, the angle criteria can be written as Equation 4.6.

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

Where n denotes the number of the point pairs in a seam-line. The other criteria is the edge length in seam-line. For a pair of seam-lines with the

same total length, different points distribution will results an undesired tension along the seam-line in which causes wrinkle to occur around the seam-line. Within every seam-line, all the point are stored in an predefined order, thus in order to ensure the consistency of the distribution of the points in the seam-line pairs, the edge that connects 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.

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

Where l_{i_f} denotes the length of the edge $\overline{p_i p_{i+1}}$ in a seam-line and l_{i_b} 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.

$$Error_s = \frac{Error_{angle} + Error_{edge}}{n} \tag{4.8}$$

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. Therefore, the consistency of the shape for each pattern need to be kept through out the evolution to ensure the cloth that fits to a character remains the same. The shape of a geometry can be described via many methods. In this thesis, the definition 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, if the included angle on each corresponding crucial point is identical between two patterns, the shape of these two pattern is considered same.

In this algorithm, during the evolution, the changes of all the inner angle on every point of the contour of a block is measured as the cost of shape evaluation. see Figure 4.13.

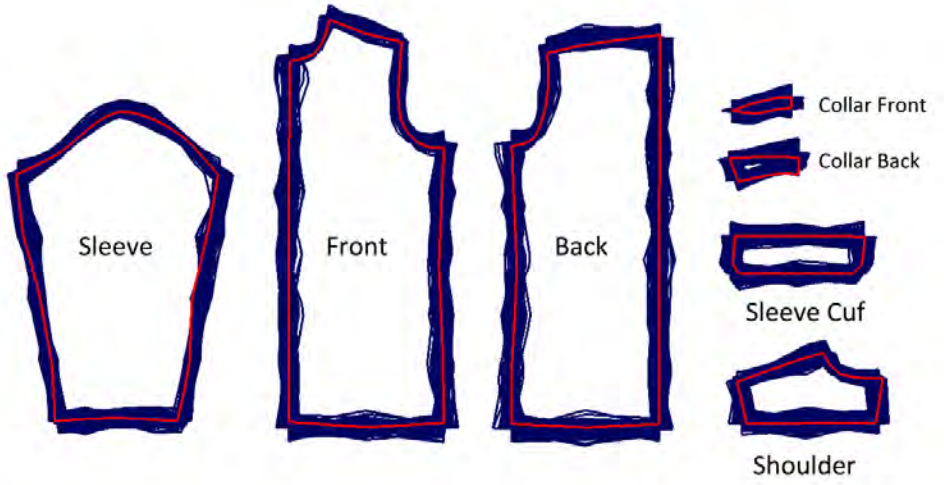


Figure 4.13: *Shape evaluation for the first generation patterns, Blue lines indicate the variations of the patterns, and the patterns that are indicated by the red line has the best shape preservation.*

Because this algorithm solves a multi-objective 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(explained in following section) by the rest of the solutions within this set (Michalewicz 1996). However, lots of real problems in the real world are multi-objective problems that require a single solution. The common method for finding a single solution for a multi-objective optimization problem is that, multiple fitness function are used for the construction of the "Pareto front", then a higher-level objective is used for find the "best" solution among the "Pareto front".

When adjusting a cloth, given a character, only one size of the cloth is needed for dressing the character. In order to select one solution from the "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". When the evolution limit is reached, all the solutions in the "Pareto front" are considered as "good" solutions to the character dressing problem. In other words, all the patterns in the "Pareto front" has achieved good fit and all the seam-lines are consistent. Therefore, at this stage, the solution with the least shape difference than the original cloth patterns is selected as the final solution of this optimization.

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 used as the criteria for selecting one solution from "Pareto front".

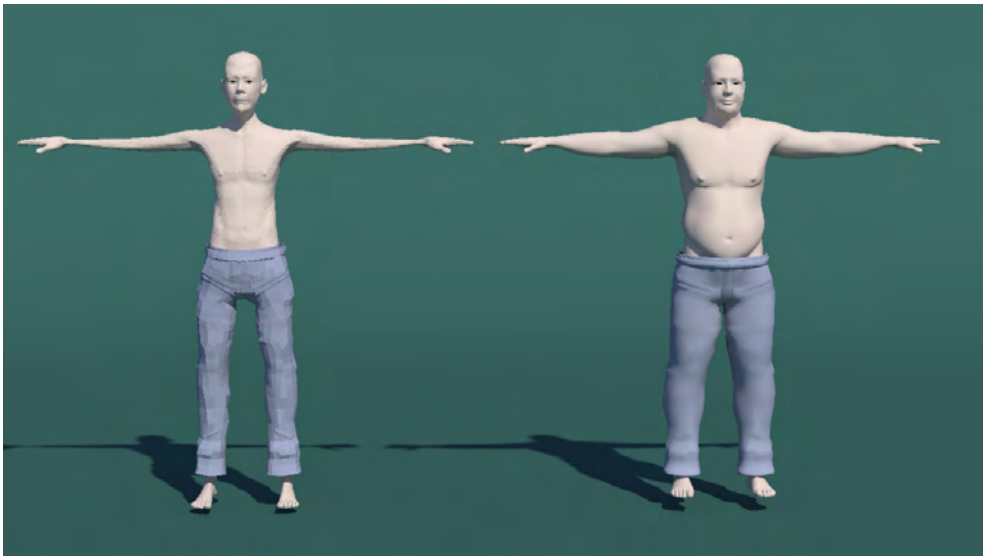


Figure 4.14: *Character on the left has much thinner arms than character on the right.*

For the character on the left side, because the arm is much thinner than the one on the right, the arc length of the "Arm hole" is much shorter than the one on the right. However, two character have identical height, therefore,

their shirt have the same back length. Figure 4.15 demonstrates the "Front" pattern resized for the characters in Figure 4.14. The pattern is selected based on the minimum changes of inner angle of the "critical point" of the solutions. As demonstrated in Figure 4.16, the design of the shirt is preserved.

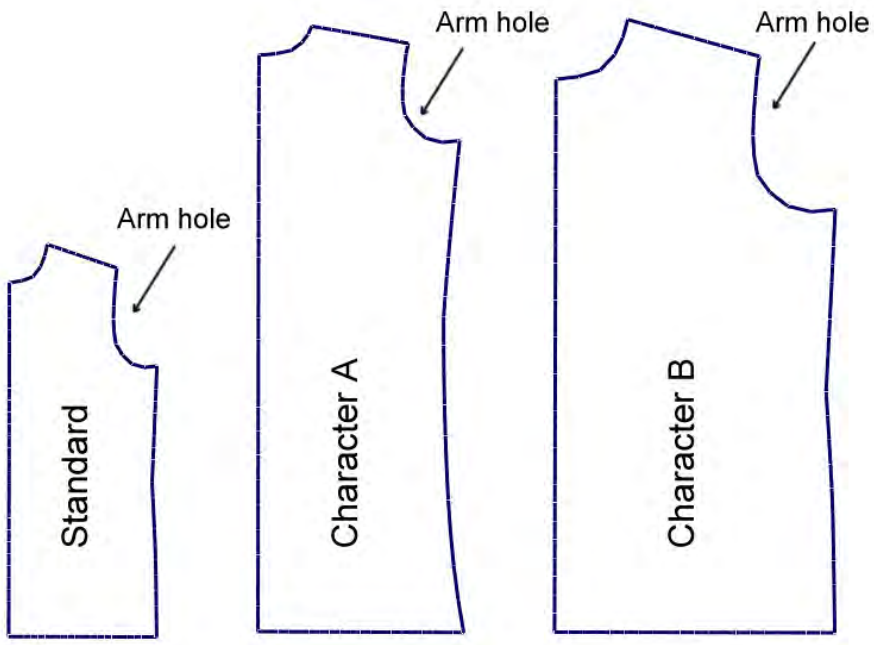


Figure 4.15: *The standard pattern(left), the "Front" pattern for the character on the left of Figure 4.14(middle), the "Front" pattern for the character on the right of Figure 4.14(right)*

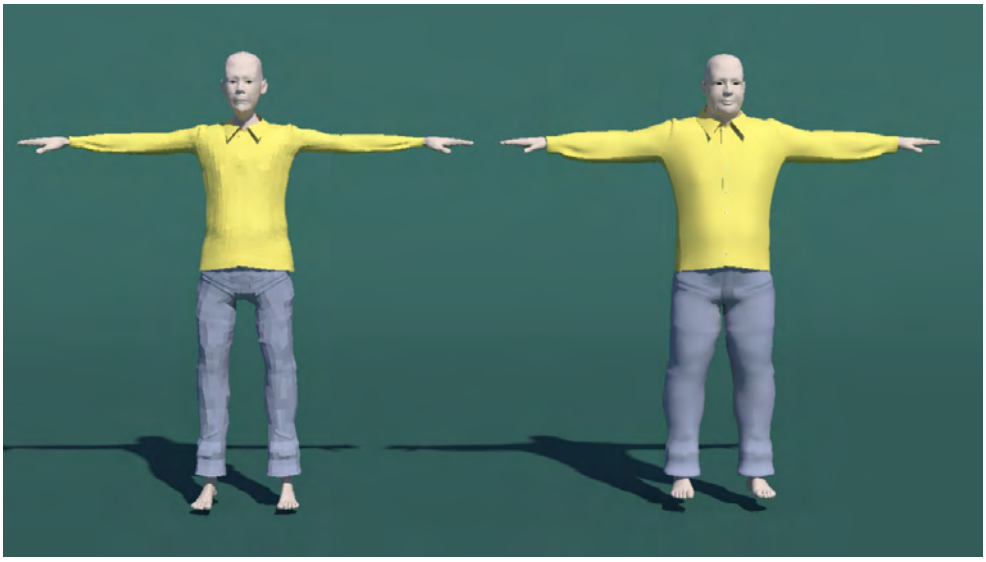


Figure 4.16: *Fit shirt modelled for characters*

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 according to the 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 objectives at the same time. Therefore, the concept of domination is introduced into this area and two solution are compared by whether one is dominated by another. Deb (2001) defines the concept of domination, that is, a solution A is said to dominate 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 then solution B in at least one objective.

To demonstrate the concept of domination, Figure 4.17 illustrates an optimization problem with five solutions $s_1 ... s_5$ and two objectives f_1 and f_2.

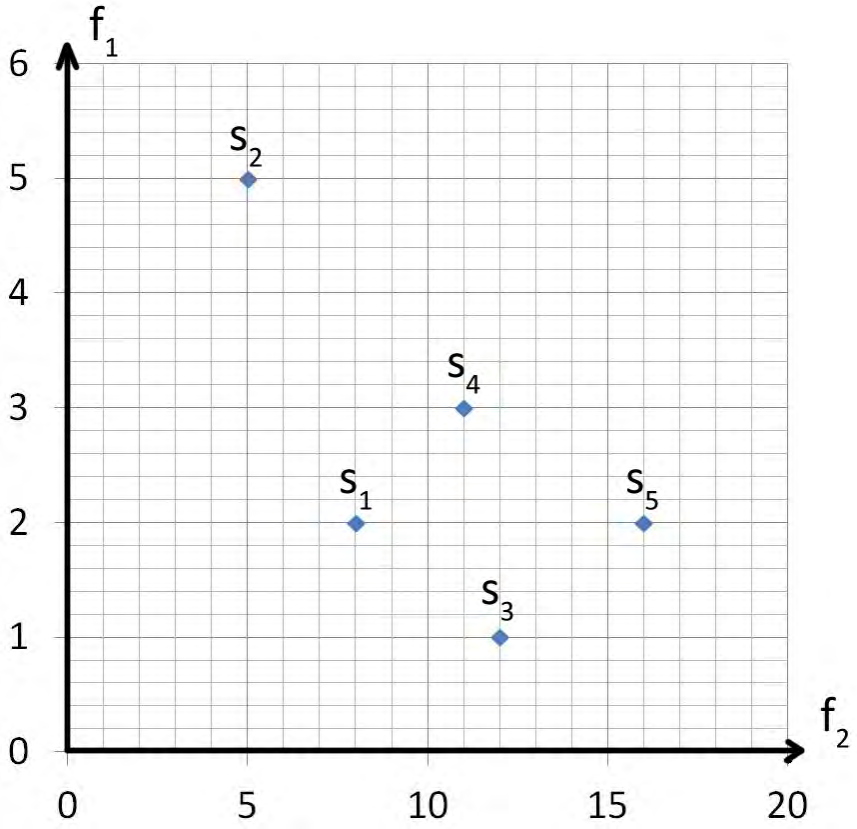


Figure 4.17: *Domination between five solutions*

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 equality 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 4.17, 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. When comparing s_4 to s_1, s_4 performs better on f_2 than s_1 but worse on f_1, therefore, s_1 and s_4 does not dominates each other.

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

In this case, after comparing all pairs of the solutions, it can be concluded that s_5 dominates s_1, s_2, s_4 as well as s_3 dominates s_1, s_2, s_4. Among s_3 and s_5, s_3 exceed s_5 in f_1 but dwarfed by s_5 in f_2. Therefore, s_3 and s_5 are non-dominated by each other but dominate the rest of the population. Consequently s_3 and s_5 consist the "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 from the "Pareto front". In the final step, the second shape objective is used for selecting one solution with less shape distortion as the final solution for the character dressing problem.

4.3 Pattern assembling

After every patten is adjusted into the correct shape and size based on the measurements from the character, they are ready to be placed on the character 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 takes place. Three steps are involved for the construction of the 3D cloth. Firstly, the bounding surface of the body part of the character 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,

constrains are created to every point pair in the seam-line and physical simulation is performed to to stitch patterns together to give the final shape of the cloth.

The concept of the bounding surface is based on the idea of the 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 piece of textile materials. Therefore, construct a developable surface around the body of the character first and then transfer the patterns that are also developable onto it can minimize the distortion during the pre-positioning.

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

After 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 4.18

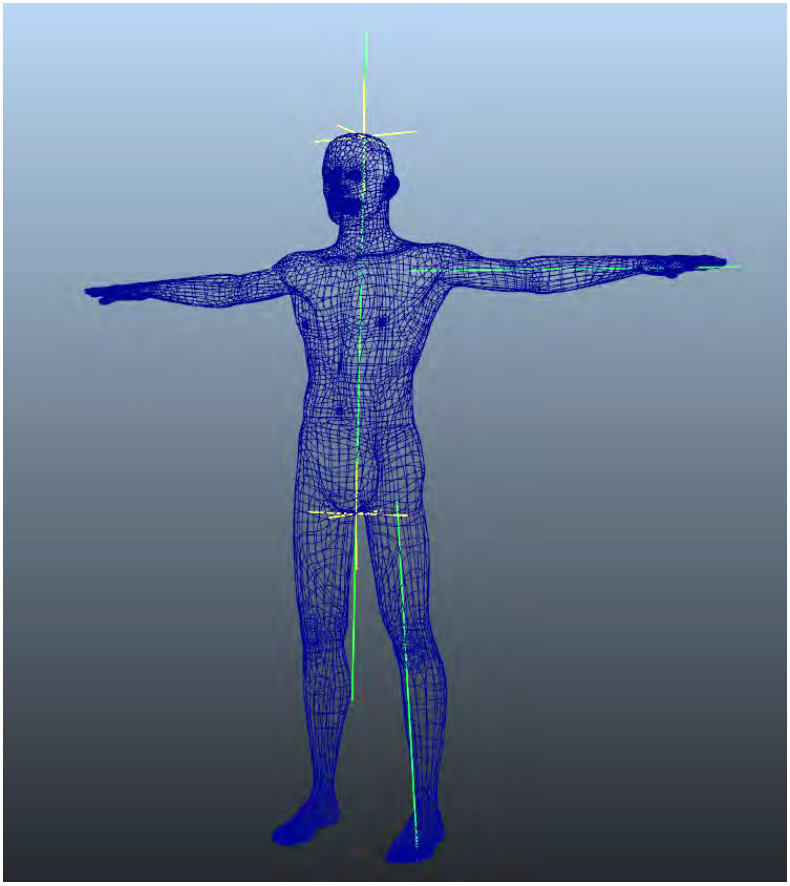


Figure 4.18: *Centre axis for the body part on the left side of the subject.*

The radius of the cylindrical bounding surface is determined by the number of pattern associated with this body part. At this step, pattern are arranged on the flattened bounding surface side by side according to position of the sewing relationship as showed in Figure 4.19. The width on the U direction of all the patterns is the circumference of the bounding surface. Figure 4.20 demonstrates all the bounding surfaces for a character.

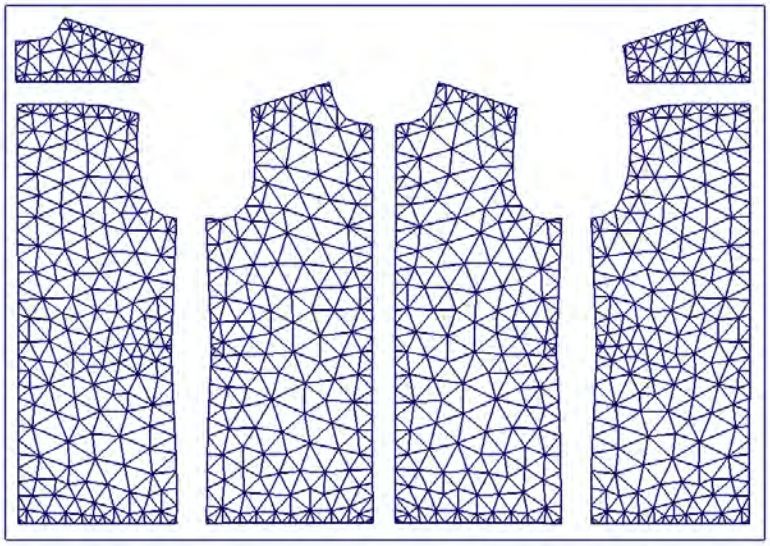


Figure 4.19: *Patterns arranged side by side within the range of a box, this box is the flattened bounding surface.*

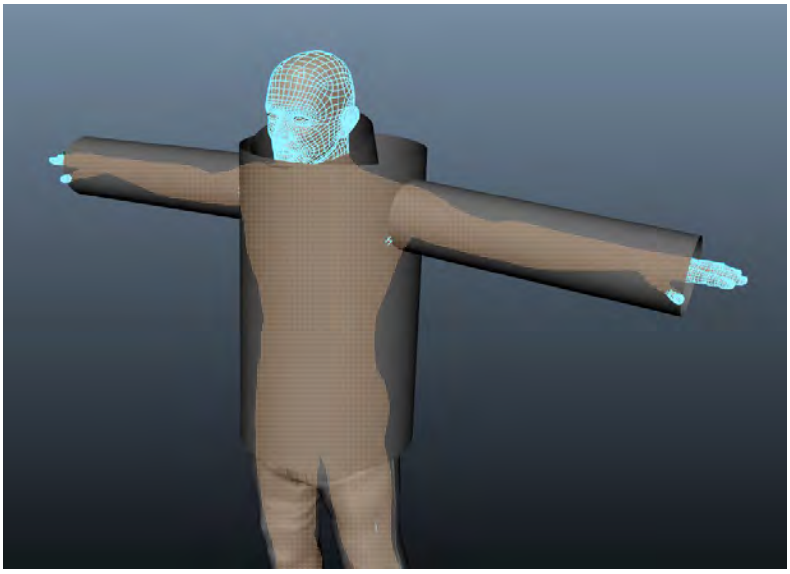


Figure 4.20: *Bounding surfaces on character.*

In the next step, the vertices of the 2D pattern are transferred onto the bounding surface. Based on the parametric coordinate of each point of the pattern on the flattened bounding surface, each point can be located by the same parametric coordinate on the cylindrical bounding surface. Finally, by applying the same topology structure to the points on the bounding surface,

the 3D cloth patterns can be constructed. Figure 4.21 demonstrate the fully positioned shirt patterns of a character.

Figure 4.21: *3D patterns*

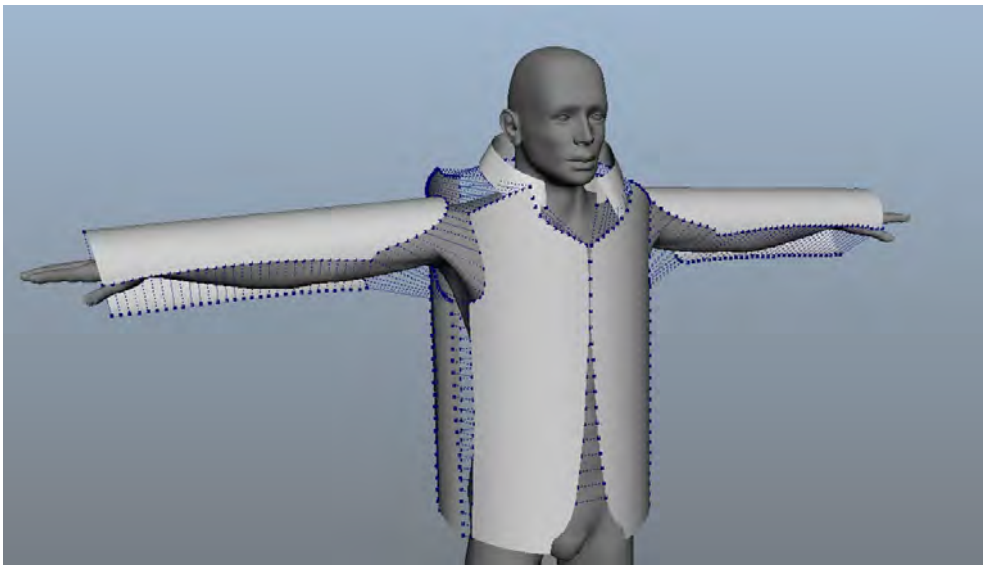


Figure 4.22: *3D pattern sewing*

In the final step, based on the seam-lines between each pattern, constrains are created between two point in a point pair on the seam-line. Then physical simulation is performed to pull all the patterns together. In this the-

sis, the nCloth module in Maya2012 is used to perform the physical simulation. Figure 4.22 illustrates all the sewing constrains.

In order to validate the algorithm presented in this chapter, a shirt design and a trousers design presented in Xiong (2008) are used for dressing four different characters with largely different body shapes and proportions. This experiment is performed on an Intel Xeon 3.33GHz PC with 8GB RAM running Windows 7 (64-bit) operating system. For cloth pattern adjusting genetic algorithm, 400 initial solutions are generated at the beginning of the algorithm. For each character, 200 evolutions are performed. Figure 4.25, 4.26, 4.27 and 4.28 demonstrate the final solution for each character respectively. After pattern adjusting algorithm is complete, all the patterns are transferred onto the bounding surface of the character, this is shown in Figure 4.29. Figure 4.30 and 4.31 demonstrate the final cloth fit onto each character. Finally, Table 4.2 lists out the fitness value of the final solution of all characters. Notices that for character A and D, because the body proportion is largely differ from the standard human body proportion that the shirt pattern and trousers pattern are based on, cloth pattern requires larger deformation to fit the character A and D than character B and C. Therefore, the result of the shape evaluation for the patterns for character A and D is worse than the patterns for character B and C.

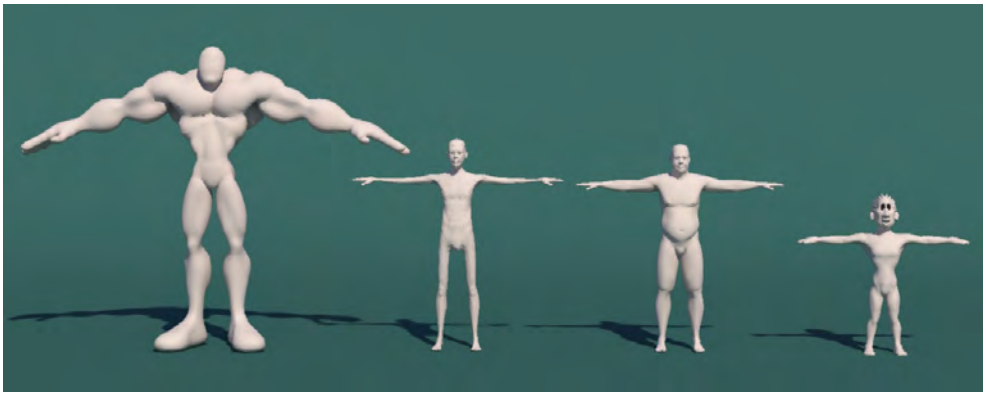
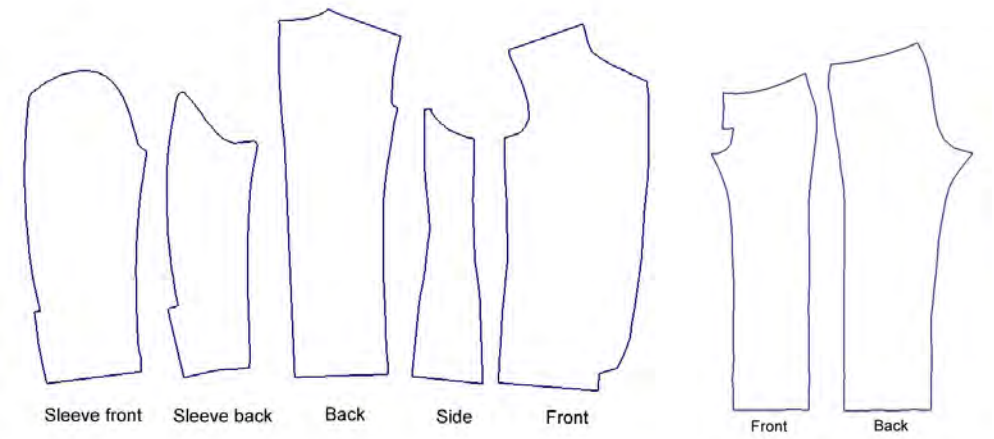


Figure 4.23: *Four different characters used for the cloth modelling experiments, from left, Character A, B, C and D.*



(a) **(b)**

Figure 4.24: *Cloth patterns(Xiong 2008), standard shirt patterns(left); standard trousers patterns(right)*

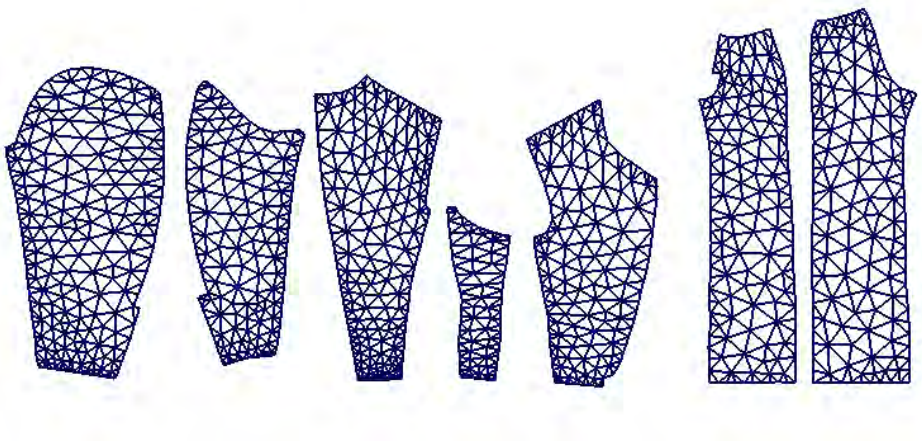


Figure 4.25: *Patterns for Characters A*

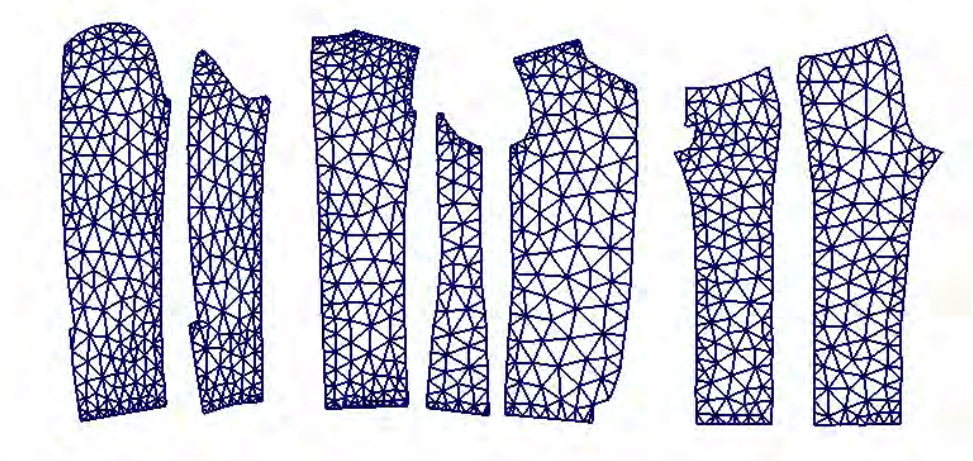


Figure 4.26: *Patterns for Characters B*

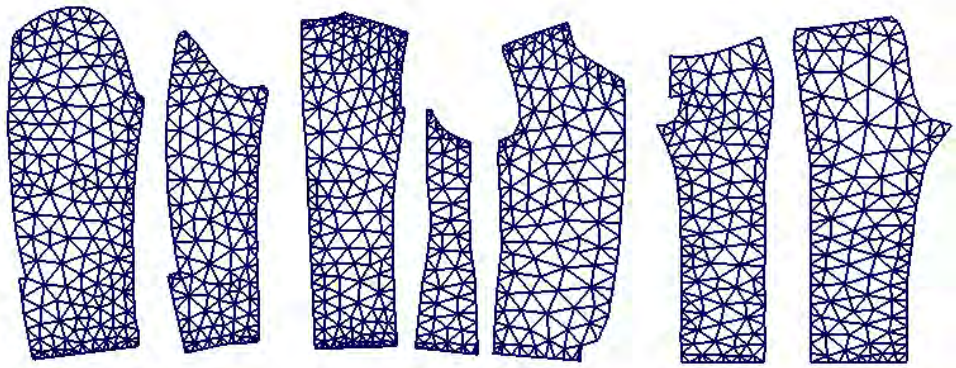
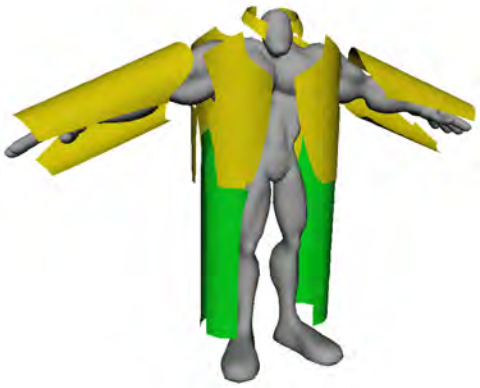


Figure 4.27: *Patterns for Characters C*

Figure 4.28: *Patterns for Characters D*



(a) *Character A*

(b) *Character B*

(c) *Character A*

(d) *Character B*

Figure 4.29: *3D Patterns on the bounding surface of characters*

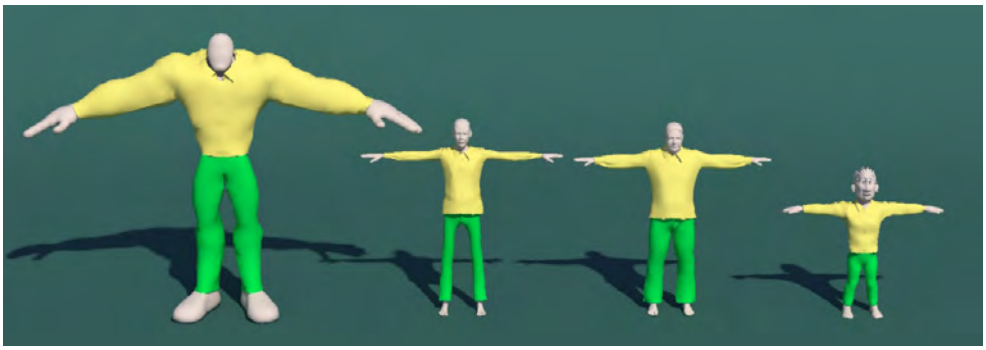


Figure 4.30: *Front view of four different characters in the same cloth design.*

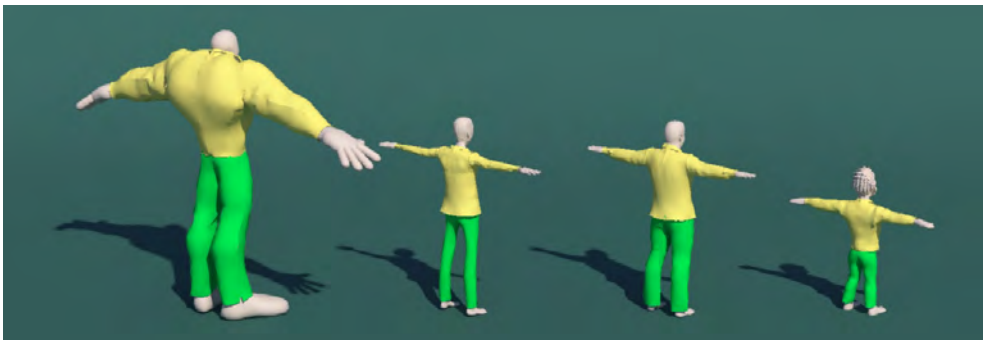


Figure 4.31: *Rear view of four different characters in the same cloth design.*

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

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

4.4 Conclusion

This chapter presents an automatic pattern based cloth modelling method for dressing cloth onto characters with different body shapes and proportions.

This method takes 2D cloth patterns as inputs. By extracting the measurements from a character model, the shape and size of the cloth patterns are adjusted automatically to fit the cloth onto the character. During the pattern adjustment, by considering the measurements, seam-line among the patterns as well as the shape of each pattern, the original design of the cloth is preserved throughout the fitting process. Finally, all the patterns are transformed into mesh and positioned around the body, physical simulation is used to stitch all the patterns together based on the seam-line of each pattern to construct the final 3D cloth.

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

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

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

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

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

Chapter 5

Conclusion and Future Works

5.1 Conclusion

Virtual character has been widely used for the construction of the virtual environment. It has become increasingly important in the production of films, TV series and computer games. The outfit of the virtual character is one of the most important elements consist of the personality of the virtual characters. Virtual clothing has been widely used in the fashion industry and computer animation.

In the fashion industry, a real cloth consists of a number of cloth patterns, therefore, pattern based virtual clothing method is widely used in the cloth designing process. In film and game industry, the clothe of the character is normally modelled by using general-purpose 3D modelling method, this type of techniques requires large amount of manual operation. Because cloth is modelled manually, the appearance of the cloth is largely determined by the artistic skill of the modeller. With the increasing computer power, more and more characters can be simulated simultaneously in a virtual environment. Modelling cloth for multiple character with different body shapes and proportions efficiently has become an imperative task for the production

of the films and games. Using general-purpose modelling technique to model cloth for different characters not only requires large amount of manual work, but also consumes a large proportion of the production budget.

This thesis proposed an automatic pattern based cloth modelling method for computer animation. Current pattern based cloth modelling technique requires deep knowledge in textile engineering and tailoring expertise, few animation artists possess this skill. Therefore, it is rarely used in the production of films and games. However, by automating the process of character measuring, pattern adjustment and pattern assembling, the pattern based cloth modelling method no longer requires tailoring knowledge to use. It eliminates the tediousness of the traditional cloth modelling method and opens a door to animation artists to directly use the large amount of existing cloth pattern designs in the fashion industry to create cloth for their characters.

The virtual clothing method presented in this thesis mainly consists two parts, character measuring and pattern adjustment. In order to use cloth patterns to generate 3D cloth for a character, the measurements of the character need to be extracted first. Because the standard posture used for modelling character differs from person to person. The traditional anthropomorphic data acquisition method no longer suits the requirement of animation character cloth modelling. In order to solve this problem, geodesic distance is used for measuring character body instead of euclidean distance. Now, the character can be measured in any posture.

This thesis presented a novel geodesic computation scheme that utilises geodesic curvature flow to calculate geodesics on character model. In order to cope with the increasing number of 3D scanned models, this scheme is designed to consider two types of inputs, polygon model and point cloud model. In order to improve the efficiency of the character measuring process, a linear time complexity geodesic algorithm is also proposed.

For the pattern adjustment, an automatic cloth patter adjustment method

is presented. This method combines the traditional cloth making techniques with modern evolutionary algorithm to generate fit cloth for characters. This method takes 2D cloth patterns and measurements of the character as inputs. Considering the measurement, seam-line among the patterns as well as the shape of each pattern, the original design of the cloth is preserved through out the fitting process. Then all the patterns are transformed into polygon mesh and positioned around the character body for assembling. Last but not least, physical simulation is performed to the positioned cloth pattern meshes for pattern assembling. One of the most significant advantages of this method is the ability of dressing character with any body shapes and proportions automatically. By automating the process of cloth pattern adjustment, it is able to transfer a cloth design from one character to another character with different body shapes and proportions.

The main contributions of the virtual clothing method presented in this thesis are,

1. An automatic virtual clothing method is presented to bridge the gap between traditional tailoring techniques and virtual cloth modelling method in computer animation. This method enables the animation artists to use the existing cloth design patterns in the fashion industry to create cloth that fits to their character automatically. Also the same cloth design can be fit onto different characters with different body shapes and proportions for the application such as crowd system.

2. This thesis described a geodesic curvature flow based geodesic computation scheme for the measurement extraction. The most important contributions is that, one of the proposed algorithms 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 this algorithm in terms of both speed and accuracy. By integrating Algorithm.3 into the

measuring system, the time consumed on solve the geodesics between multiple pair of source and destination has largely reduced while the accuracy of the solution is still maintained.

3. In order to cope with the increasing number of the 3D scanned character models. Another geodesic algorithm is introduced to handle the point cloud data. This algorithm allows geodesics to be calculated directly on the scattered point data so that the measuring method can be performed directly on the 3D scanned character model.

4. An automatic cloth pattern adjustment method is proposed. This method utilizes the evolutionary algorithm to generate fit cloth for the virtual character. It preserves the original cloth design by evaluating the measurements, seam-lines as well as the shape of each pattern through out the fitting process. By automating the process of pattern adjustment, using pattern based cloth modelling techniques no longer requires expertises in tailoring. Animation artists can directly use the existing cloth design pattern to create cloth for their characters. The unintuitiveness and tediousness of the current cloth modelling method can be largely reduced. Moreover, this is the first attempt to utilize the genetic algorithm on 2D cloth patterns for solving the virtual cloth fitting problem.

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 task of cloth modelling, it cannot generates the motion of the cloth. The full cycle of the virtual clothing involves simulating the dynamic property of the

cloth, therefore, the proposed method could be improved by adding the simulation module. Data driven wrinkle generation is suggested. By creating wrinkle database for each type of fabrics, cloth patterns can be associated with a wrinkle database for a particular type of fabric that the cloth is made from. When dressing different characters with the same cloth design, as long as the dynamic property of the cloth consists, only the coarse physical simulation is needed for each character, the same wrinkle database can be applied to multiple character to generate fine deformation details. This allows the efficiency of creating virtual cloth for multiple character to be further improved.

2. **Automatic datum point detection**

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

3. **Interactive cloth design**

   Given accurate measurements, the cloth modelling method presented in this thesis can generates fit cloth for any character. However, sometimes, during the fashion design, for a particular kind of body proportion, the design of the cloth need to be changed a bit for the best appearance. Therefore, the research in this thesis can be improved by adding a pattern editing interface to allow cloth pattern to be edited by users.

Bibliography

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.

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.

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.

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

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

K. Deb (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley Interscience Series in Systems and Optimization. Wiley.

K. Deb, et al. (2002). 'A fast and elitist multiobjective genetic algorithm: NSGA-II'. *Evolutionary Computation, IEEE Transactions on* **6**(2):182–197.

P. Decaudin, et al. (2006). 'Virtual Garments: A Fully Geometric Approach for Clothing Design'. *Computer Graphics Forum (Eurographics'06 proc.)* **25**(3).

R. Digest (2010). *New Complete Guide to Sewing: Step-By-Step Techniquest 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.

A. Ebert, et al. (2002). 'Efficient Assistance of Virtual Dress Fitting using Intelligent Morphing'. In J. J. Villanueva (ed.), *Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing*. ACTA Press.

EN:13402 (2001). *Size designation of clothes*.

R. Enns & G. McGuire (2000). *Nonlinear Physics With Maple for Scientists and Engineers*. Springer.

M. Frings (2002). 'The Golden Section in Architectural Theory'. *Nexus Network Journal* **4**(1):9–32.

A. Ghosh & S. Tsutsui (2003). *Advances in Evolutionary Computing: Theory and Applications*. Natural Computing Series. Springer.

D. E. Goldberg & K. Deb (1991). 'A comparative analysis of selection schemes used in genetic algorithms'. In *Foundations of Genetic Algorithms*, pp. 69–93. Morgan Kaufmann.

R. L. Graham (1972). 'An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set'. *Inf. Process. Lett.* **1**(4):132–133.

J. Grefenstette (1986). 'Optimization of Control Parameters for Genetic Algorithms'. *Systems, Man and Cybernetics, IEEE Transactions on* **16**(1):122–128.

D. Gupta & N. Zakaria (2014). *Anthropometry, Apparel Sizing and Design*. Woodhead Publishing Series in Textiles. Woodhead Publishing Limited.

T. Gwiazda (2006). *Crossover for single-objective numerical optimization problems*. Genetic algorithms reference. TOMASZGWIAZDA E-BOOKS.

S. Hadap, et al. (1999). 'Animating wrinkles on clothes'. In *Proceedings of the conference on Visualization '99: celebrating ten years*, VIS '99, pp. 175–182, Los Alamitos, CA, USA. IEEE Computer Society Press.

G. Hairer (2010). *Solving Ordinary Differential Equations II*. Springer Berlin Heidelberg.

G. M. Hannah (1919). 'Dressmaker's Pattern Outfit'. US Patent No.1313496.

E. Harms (1938). 'The Psychology of Clothes'. *American Journal of Sociology* **44**(2):pp. 239–250.

R. Haupt & S. Haupt (2004). *Practical Genetic Algorithms*. Wiley.

B. Hinds & J. McCartney (1990). 'Interactive garment design'. *The Visual Computer* **6**(2):53–61.

B. Hinds, et al. (1991). 'Pattern development for 3D surfaces'. *Computer-Aided Design* **23**(8):583 – 592.

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.

J. Horn, et al. (1993). 'Multiobjective Optimization Using the Niched Pareto Genetic Algorithm'. Tech. rep.

J. Horn, et al. (1994). 'A niched Pareto genetic algorithm for multiobjective optimization'. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pp. 82–87 vol.1.

K. Howland (2008). 'The Merits of a Basic Fitting Pattern'. *Threads* **79**:48 – 52.

W. Hundsdorfer & J. Verwer (2003). *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer Series in Computational Mathematics. Springer.

H. Ishibuchi & T. Murata (1996). 'Multi-objective genetic local search algorithm'. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pp. 119–124.

H. Ishibuchi, et al. (2008). 'Evolutionary many-objective optimization: A short review'. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pp. 2419–2426.

ISO/TR-10652 (1991). *Standard sizing systems for clothes*.

B. Jiang (1998). *The Least-Squares Finite Element Method: Theory and Applications in Computational Fluid Dynamics and Electromagnetics*. Lecture Notes in Mathematics. Springer.

I. Jolliffe (2002). *Principal Component Analysis*. Springer Series in Statistics. Springer.

S. Katz & A. Tal (2003). 'Hierarchical mesh decomposition using fuzzy clustering and cuts'. *ACM Trans. Graph.* **22**(3):954–961.

D. G. Kendall (1977). 'The Diffusion of Shape' .

R. Kimmel & J. A. Sethian (1998). 'Computing geodesic paths on manifolds'. *Proceedings of the National Academy of Sciences* **95**(15):8431–8435.

R. Kittler, et al. (2003). 'Molecular Evolution of Pediculus humanus and the Origin of Clothing'. *Current Biology* **13**(16):1414 – 1417.

J. Knowles & D. Corne (1999). 'The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation'. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 1, pp. –105 Vol. 1.

J. R. Koza (1995). 'Survey of genetic algorithms and genetic programming'. In *WESCON/'95. Conference record. 'Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies'*, pp. 589–.

E. Kreyszig (1991). *Differential Geometry*. Differential Geometry. Dover Publications.

T. L. Kunii & H. Gotoda (1990). 'Singularity theoretical modeling and animation of garment wrinkle formation processes'. *Vis. Comput.* **6**(6):326–336.

Q. Liu, et al. (2011). 'Automatic body segmentation with graph cut and self-

adaptive initialization level set (SAILS)'. *Journal of Visual Communication and Image Representation* **22**(5):367 – 377.

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 form 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.

V. Pareto (1906). *Manual of political economy (manuale di economia politica)*. Kelley, New York. Translated by Ann S. Schwier and Alfred N. Page.

G. Peyré & L. D. Cohen (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.

J. D. Schaffer (1985). 'Multiple Objective Optimization with Vector Evaluated Genetic Algorithms'. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 93–100, Hillsdale, NJ, USA. L. Erlbaum Associates Inc.

N. A. Schofield & K. L. LaBat (2005). 'Exploring the Relationships of Grading, Sizing, and Anthropometric Data'. *Clothing and Textiles Research Journal* **23**(1):13–27.

G. Schott (1992). 'The extent of man from Vitruvius to Marfan'. *The Lancet* **340**:1518 – 1520.

Y. Schreiber & M. Sharir (2006). 'An optimal-time algorithm for shortest paths on a convex polytope in three dimensions'. In *Proceedings of the twenty-second annual symposium on Computational geometry*, SCG '06, pp. 30–39, New York, NY, USA. ACM.

H. Selin (2008). *Encyclopaedia of the History of Science, Technology, and Medicine in Non-Western Cultures*. Springer Reference Series. Springer.

J.-A. Serret (1851). 'Sur quelques formules relatives la thorie des courbes double courbure.'. *Journal de Mathmatiques Pures et Appliques* pp. 193–207.

M. Sharir(& A. Schorr (1984). 'On shortest paths in polyhedral spaces'. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, STOC '84, pp. 144–153, New York, NY, USA. ACM.

A. Sheffer, et al. (2005). 'ABF++: fast and robust angle based flattening'. *ACM TRANSACTIONS ON GRAPHICS* **24**:311–330.

O. M. Shir & T. Bäck (2006). 'Niche radius adaptation in the CMA-ES niching algorithm'. In *Proceedings of the 9th international conference on Parallel Problem Solving from Nature*, PPSN'06, pp. 142–151, Berlin, Heidelberg. Springer-Verlag.

M. Shoben & J. Ward (1987). *Pattern Cutting and Making Up*. Butterworth.

SmithMicro (2012). 'Poser2012'.

W. Spears & V. Anand (1991). 'A study of crossover operators in genetic programming'. In Z. Ras & M. Zemankova (eds.), *Methodologies for Intelligent Systems*, vol. 542 of *Lecture Notes in Computer Science*, pp. 409–418. Springer Berlin Heidelberg.

A. Spira & R. Kimmel (2002). 'Geodesic Curvature Flow on Parametric

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

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

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

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

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

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

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.

M. B. M. T. Thomas Stumpp, Jonas Spillmann (2008). 'A Geometric Deformation Model for Stable Cloth Simulation'. In *Workshop on Virtual Reality Interaction and Physical Simulation, 2008*.

L. Trefethen & D. Bau (1997). *Numerical Linear Algebra*. Miscellaneous Bks. Society for Industrial and Applied Mathematics.

N. Tsopelas (1991). 'Animating the Crumpling Behavior of Garments'. In *Proc. 2nd Eurographics Workshop on Animation and Simulation*, pp. 11–24.

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.

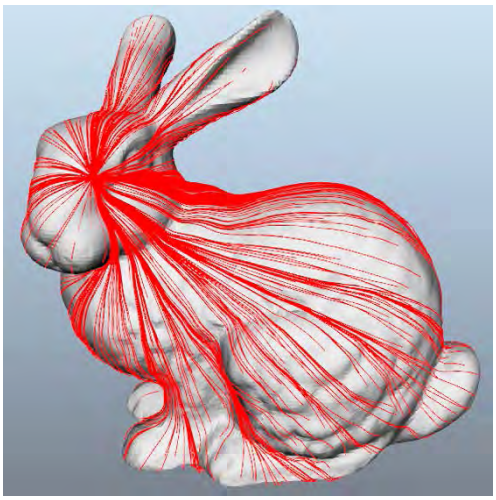
E. Zitzler (1999). *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Ph.D. thesis, ETH Zurich, Switzerland.

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

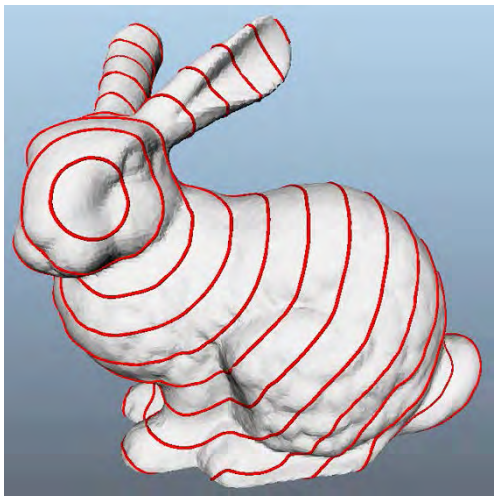
Appendix A

Appendix Title

Experiment for comparing time complexity that performed on large models:

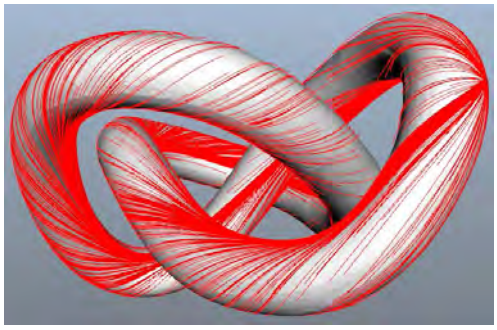


(a) *geodesics on bunny*

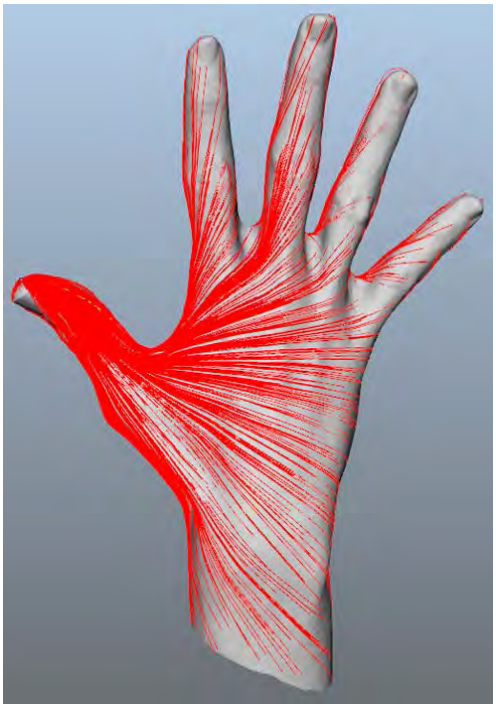


(b) *isoline on bunny*

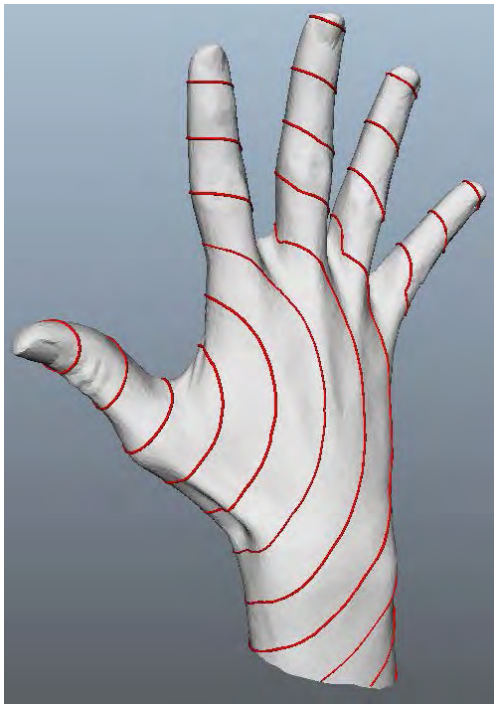
(c) *geodesics on asian dragon*



(d) *geodesics on knot*



(e) *geodesics on hand*



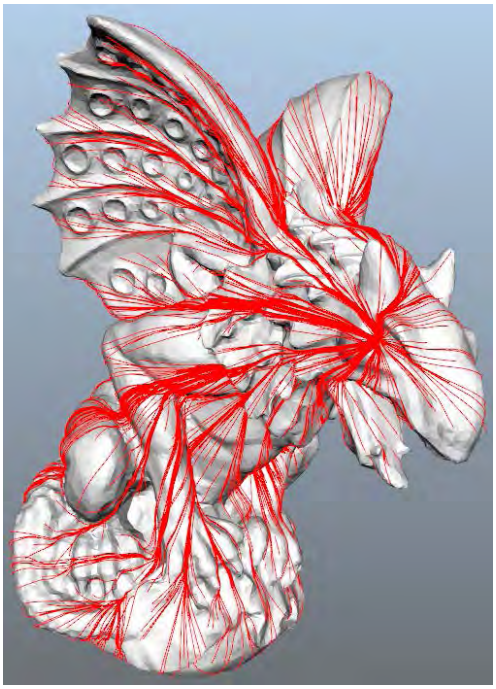
(f) *isoline on hand*

(g) *geodesics on Neptune*

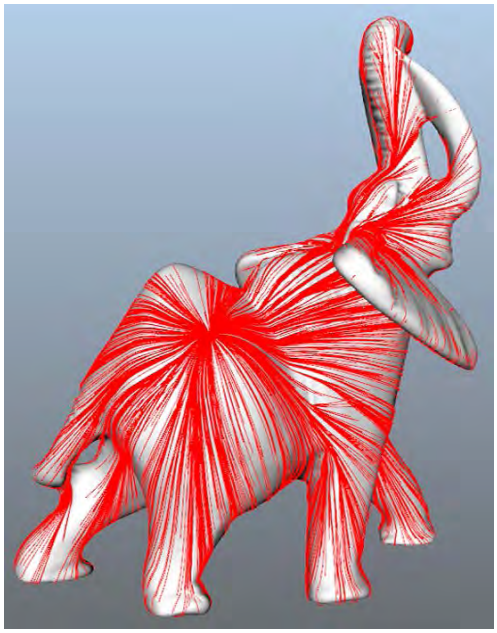
(h) *isoline on Neptune*

(i) *geodesics on Lucy*

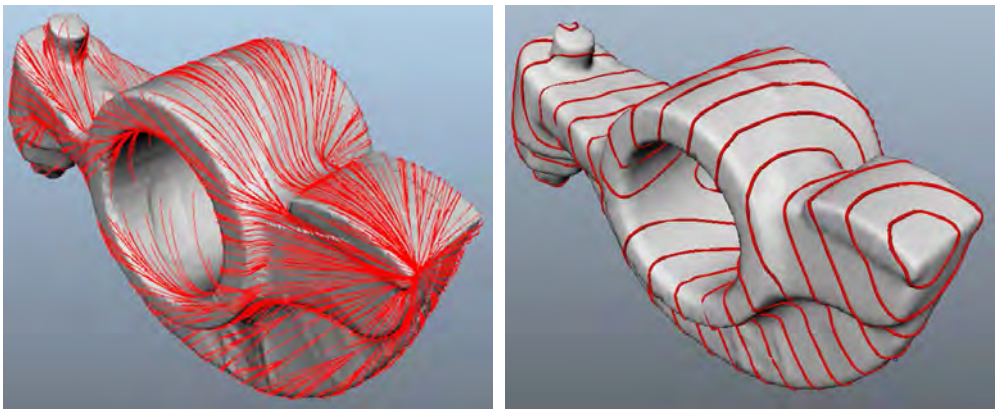
(j) *isoline on Lucy*



(k) *geodesics on Gargoyle*



(l) *isoline on elephant*



(m) *geodesics rockarm* **(n)** *isoline on rockarm*

Figure A.1: *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 memory 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" time 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 demonstrates the shortest paths or isolines on these models that are used in Table.A.1. It can be noted 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.1c, Lucy in Figure.A.1i and Gargoyle in Figure.A.1k the approximate algorithm presented in this thesis is many times faster than the MMP approximation and ICH_2.

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.