Title

Abstract

1. Introduction(10 pages)

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

Our species, as the most intelligence creature on the earth, has many distinctive feature than other animals, one of the feature that differs us from others is that, human-being is the only species that able to design, produce and has needs to wear cloth. In terms of functionality, cloth provides to us a shield that protects us against many hazards in the environment, which vastly widens the range that human-being able to reach even in the outer space. Another function that cloth provides to us is to satisfy our ascetic need and after hundreds years of development, cloth has become a major medium for self-expression.

In virtual world of today, research on virtual clothing is a crucial factor that directly affects the fidelity of virtual human. Especially with its high speed development of computer hardware, realistic virtual character has been wildly used not only in research area, but also in film production, TV and games. Moreover, virtual clothing has brought a big leap in garment manufacturing. With the help of virtual clothing techniques, it transferred this traditional labor-intensive industry into a highly automated modern industry.

Virtual clothing requires two major tasks, cloth modeling and cloth simulation, as the area of application differs, the focus point of virtual clothing various. For computer aided cloth design in fashion industry and computer aided cloth cutting in garment manufacturing field, the modeling accuracy for cloth pattern is crucial [reference for automatic cloth grading, cloth pattern design software, computer aided cutting]. In the textile engineering area, the accuracy of physical cloth simulation is essential. In computer animation and computer game area, the efficiency and visual fidelity is the main research object for many researcher who works in this area.

The research proposed in this thesis focus at the virtual clothing problem in computer animation area. Many techniques have been developed in this area, [reference for virtual clothing in animation ] However, virtual clothing, which involves both textile engineering knowledge and artistic expertise, is still being considered as a challenging and time-consuming work. Generally speaking, there are two major goals that researchers are trying to achieve in this area, to reconstruct the shape [reference: clothing modeling, geometrical wrinkle generation] and texture of the cloth[ reference on cloth texture ] as well as it dynamic property[ reference on physical cloth simulation ]. Moreover, since the virtual character is the “star” in computer animation and games. The costume design for the character is equally important as character design and it acting. Thus dressing virtual character has never been considered as an easy task. It evolves many cloth design techniques from real world and many hours of time-consuming cloth simulation process.

Especially in today’s film and game industry, the number of characters capable to be handled in a scene is vastly increased both by the needs of audience and technologic improvement. Dressing multiple characters efficiently and effectively has become an even more challenging task. Physical simulation is well known as a very time-consuming task for all kinds of dynamic application, especially when high physical accuracy is required. Winkles are the main deformation feature in cloth simulation, and fine wrinkles require fine mesh for cloth 3D representation. To create realism wrinkle, physical cloth simulation technique usually takes a significant amount of computation resources [ reference on physical cloth simulation focus on efficiency ]. To improve the efficiency of virtual clothing, many techniques have been developed to minimize physical simulation requirement and mimic fine wrinkle by using other modeling techniques such as geometrical modeling or data-driven wrinkle generation method. However, with those methods, a generic method which can be applied to different character is still considered as a very challenging task. Many method falls in this category only applies on a specific character [ reference on data-driven method and geometrical cloth modeling method ].

In terms of effectiveness, in the real world, a piece of cloth has its certain style as well as many different sizes to fit different customs. In the computer animation and game industry, the reusability of the current resources directly affects the cost of production. In other area such character animation , many techniques have been developed to maximize the reusability of the existing resources such as motion retargetting and motion blending to maximize the reusability of motion capture data. However, in virtual clothing area, how to model a cloth which can be used on many difference characters is still a challenging task. Also for cloth simulation, the challenge is how to use simulation data from a set of cloth to another with different size or style.

This thesis introduced a framework that not only able to model cloth efficiently but also be able to transfer the cloth on to other character with different body size whilst maintain cloth style and fitness of the cloth. Also, for wrinkle generation, this framework can generate fine wrinkle for a cloth with certain types of physical property and the simulation result can be reused for any kind of cloth as long as their textile physical property remains the same.

* 1. Virtual clothing

Virtual human is the creation of a human being in image and voice using computer-generated imagery and sound. It has been increasingly used in different domains. Virtual clothing is the area which studying clothing and cloth behavior of cloth of virtual human.

Virtual clothing is the generic terms of simulating clothes and clothing, it reproduces both visual features and physical behaviors of textile objects in computer simulated virtual reality. Cloth modeling has been a research topic in textile engineering and textile mechanics for a very long time. Followed by the increasingly demands of 3D computer generated image and film, researchers in computer graphics became interested in virtual clothing. After years of research, cloth modeling and garment simulation has evolved from basic shape modeling to creation of its complex physical behaviors. Based on the different demands of computer graphics and textile engineering, virtual clothing research has focused into two major objectives. In computer graphics, in order to achieve high level of visual realism, only macroscopic properties of cloth surface are considered [reference to Virtual Clothing: Theory and Practice]. Physical accuracy is compromised due to the limitation of computational resources.

There are two phases involved in virtual clothing, cloth modeling and dynamic simulation. Since for computer graphics, visualization is a means to validate the virtual clothing, the shape of cloth is the most crucial clue in virtual clothing. As dressing is one of the most important activities in people’s daily life, when wearing an item of clothing, its fitness is the key issue to evaluate the comfort of the cloth, and also, fitness is essential to the appealing of cloth. In computer graphics, fitness can heavily influence the physical simulation. An ill-fitted cloth could cause the failure of stitching during the assembling or intersection between body model or cloth mesh. Thus the precise fitness is the main goal of cloth modeling. Moreover, cloth is the combination of soft and flexible textile pieces. Its shapes vary under different conditions, since the shape is revealed as a consequence of interaction with its environment. Cloth simulation is focused at the recreation of shape changes while the external environment changes. In this thesis, our research mainly focuses at precise cloth modeling area.

* 1. Motivation

When making cloth in real world, usually, it involves following steps: choose the style, measure custom’s body, grading the cloth pattern, assemble patterns and try-on. For bespoke clothing, before final assembling, patterns will be loosely stitched together and put onto customer for further trimming the excise to achieve better fitness. [reference: The complete book of tailoring, [Adele P. Margolis](http://www.google.co.uk/search?tbo=p&tbm=bks&q=inauthor:%22Adele+P.+Margolis%22), 1964, Patternmaking, Helen Joseph Armstrong,2000]. In 1863, Ebenezer Butterick, who was born in Sterling, Massachusetts, United States 1826, invented the first graded sewing pattern for cloth making[US patent No.632361, US patent No.1313496], soon after his invention, it became massively popular, as they made modern fashions much more easy to access to the rapidly expanding lower middle class. This innovation also provided a standard for fashion designs to be preserved and spread with much lower cost.

Today, given the high speed development of computer hardware, character animation has played a major role in film and game industry.

Although many methods have been developed to achieve good results in this field, new needs also exist being produced by high speed development in many related area such as garment design, computer game, and animation.

However, many applications which involve virtual clothing, modeling and dressing are usually tackled in isolation. The virtual clothing which has the ability to visualize the designs has become the major tool for garment industry. Most of the current methods are focused on the detail of the physical behavior it produced. But in normal circumstances the design is based on the general human body. However, the same cloth will provide different dressing experiences when worn by different people with different body sizes. In reality, over hundreds years of cloth making history has given us a well developed method for producing different size of cloth from a single design [ reference for cloth grading and pattern design ] . However, in the virtual world, fit the same cloth design onto different types of body while convincingly maintaining its style efficiently has become the major issue of current cloth modeling methods. And this problem enlarges when more character are used in a scene.

Generally, most methods of virtual clothing focus on the shape modeling, texturing or the physical simulation of micro-textile-properties. However in those applications, geometric modeling and physical simulation are usually tackled separately. Since computer science is a relative new area that only started at 1970’s, the visual experience are heavily constrained by the computational ability of the hardware. In the past decade, technology of computer hardware has had a big leap forward and this trend is still going strong nowadays.

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

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

The detail of the apparelling for the simulation done by physical method is directly determined by the quantity of basic element it uses such as polygon or particle. For this reason, the only way to increase the detail level of the simulation is to increase the resolution of the mesh or particles. This directly leads to the incensement of the computation. Especially with the high speed of development of film production, TV, games and online trading, more and more applications require real-time ability of cloth simulation or seeking for a new method that in order to speed up the current simulation procedure. Thus, the hybrid method is bringing into the cloth simulation area. Based on the geometrical method or physical method, combine with other computer techniques such as, data driven method, image based modeling, etc. the most time consuming task which is fine detail wrinkle generation is usually handled to other high efficiency methods rather than physical calculation. Thus, the hybrid method has the ability to overcome many drawback of each method on its own and provides a high efficiency approach to the application.

In the past only few characters can be handled in a CG film or computer game due to the reason aforementioned. The cloth for a CG character usually considered as a second layer of skin of character. It created with character and in many cases, they cannot be separated. However, given the high speed development of computer hardware, paralliaze computing technique and physical simulation techniques such as CUDA, PhyX, OpenMP. More and more characters are puts in the scene and this number multiplies when crowd techniques applies. With hundreds and thousands of different characters, model and simulate cloth for every each of them individually no longer applicable considering the time it consumes and labor it costs.

Weil (1986) presented a geometrical cloth simulation technique. It is an efficient geometrical method to generate the wrinkle of the hanging cloth. But his method can only simulate the hanging cloth thus it cannot be used to simulate whole pieces of cloth or be used for cloth animation. Hinds et al. (1990) presented a method which aimed at the automation of garment production. In his system, the fold is created by harmonic function superimposed on the panels. Because this method was mainly aimed at the stationary visualization of cloth manufacturing, it is difficult to use in animation. Etzmuß et al. (2001) presented a simulation engine that was based on a particle system built up on continuum mechanics model. This system can work with 3D animation software, Curious Lab’s Poser , by exporting the animated character from Poser to their system. However, because of the limitations of the modeling module, it can only be used for a single piece of cloth. Fuhrmann et al. (2003) was the first team to address the problem of how to dress up a virtual human fast and automatically. But this method can only handle certain types of cloth. For instance, some more complex cloth design with initial folded aspects such as collars cannot be modeled. Volino and Magnenat-Thalmann. (2005) described a system aimed at fulfilling the requirements of garment designers. It described a general mechanical model for textile that combines the particle system with the accuracy of surface-based models to fulfil the requirement of accurate reproduction of the behavior of cloth on animated character. But their system is too complicated and very computational expensive for animation.

The major problem of the current method used for virtual clothing is that most of them are not designed to fit multiple characters from modeling. One style of cloth cannot be putting on another character without redo the entire modeling process. On the other hand, in the animation phrase, although the physical cloth simulation is able to produce realistic result, it consumes large amount of computational resource. Moreover, as the crowd animation technique become more popular, current methods of doing virtual clothing for massive amount of characters limits the creativity of artist.

However, In garment manufacturing, through generations of improvement, it has grown to a well developed industry, since cloth style from loosen to fit in [ reference for cloth history and when pattern based cloth making was invented ] to product different size of cloth from single design has become a fundamental skill to tailors. Nowadays, modem cloth manufacturing techniques has already been standardized just like many manufacture industry such as automobile or electronics. Cloth pattern is an important medium that transfer cloth design sketch into real wearable object. Cloth pattern is the partitioned textile piece that based on the human body structure and cloth design. It is the fundamental parts of the cloth[ reference on the cloth pattern, and cloth pattern definition]. Based on cloth pattern, and its associated measurements from human body, a cloth design can be graded into many different size, this is called tailoring. In vast manufacturing, based on the human body statistic data, standard body size for different racees are developed to standardize the cloth size, thanks for this techniques, today, we can just go into the shop and pick the cloth with the size that fits us and no matter where we go, as long as the grade of cloth correct, the cloth we brought fits.

Moreover in the apparel industry there is a huge number of cloth designs that has been stores as pattern form ready to be pass to everyone, However, in the film industry, an animator could not use those cloth patterns and resizing rules directly to assist them to model the cloth for characters directly. In fact, as mentioned above, the major methods of cloth modelling in animation still require the animator to model every part of the cloth manually. The topology structure and modelling accuracy for cloth are seriously constrained by the workload of the animator.

* 1. Challenges

To fulfill the needs of dressing multiple characters efficiently and effectively, two major problems need to be tackled.

1. Measurement acquirement.

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

Same as the finger prints, body size differs largely among people, this process is performed by experienced tailor in the real world since the accuracy of the measurement directly determines the fitness of the cloth that is made. In computer graphics, the body shape of character varies even more as the deformation and exaggerations are the sprite of animation. The body of a character is a complex surface. And cloth is designed to follow the complex shape of the body to achieve the fitness. When measuring in real world, tailor puts tension on the tape-ruler and holds the ruler as close to skin as possible to follow the body curvature to maximize the measuring accuracy. The result of the measuring is no longer a cadecean distance, it is the arc length of the bended tape ruler. In this thesis, a novel geodesic path algorithm is developed to extract the geodesic distance between measuring points.

1. Cloth pattern grading

Once all the measurements are acquired, each cloth pattern needs to be resized into the correct size that can be assembled into a completed garment. When cloth pattern was invented, the patterns are designed based on the ideal human figure of certain race. The proportions of each part of body are constructed based on the statistic data of the body size of that certain group. With the help of the body parts proportions, customs that have different height and weight are categorized into some grades and the cloth pattern are also resized to fit the body size of those grades to achieve the maximum fitness to majority.

In computer graphics, the lack of character size standard makes categorizing character by its body size impossible. Moreover, the exaggeration of character design exceeded the body parts proportion to human-beings’. Thus the cloth pattern grading principle in tailoring no longer applies.

Despite few method were developed to transferring cloth from one character to another using morphing techniques, the result were not adequate. The cloth pattern grading technique was invented to fit a cloth design onto different customs with different body size. The design itself should remains after grading. However, when using morphing, the cloth design hardly remains after transferred onto another character.

* 1. Research objective

1. Reviewing different techniques for modelling and simulation cloth and analyzing their advantages and disadvantages

2. Design an algorithm for gaining the feature measurements from a character automatically

3. Design an algorithm to create and adjust the size of the 2D cloth pattern from the feature measurement data.

4. Develop a method to create 3D virtual cloth from garment manufacture standard 2D cloth pattern.

5. Design a method using data-driven technique and based on the low-resolution coarse physical simulation to generate wrinkles.

6. Develop a prototype of virtual clothing system.

I have studied various cloth modeling and cloth simulation techniques and designed a method of virtual clothing which fits the standard of both animation and apparel industry and only involves minimal manual operation. Meanwhile it does not require the user to be professionally trained to handle the complex computer graphic techniques. Most importantly, it models the cloth based on the size of a virtual character’s body so that the clothes are able to fit onto those characters. More specifically, our method not only has the ability to generate 3D cloth from standard 2D cloth pattern with few user interactions but also can adjust the existing 3D cloth model to fit the characters. In addition, our method does not require heavy computational high-resolution mesh physical simulation to generate the fine wrinkle. We have designed a data-driven scheme based on the pressure distribution on the cloth to generate while using course physical simulation to calculate the global deformation.

Future works will mainly concentrate on how to design the algorithm to create wrinkles on the cloth using data-driven method and evaluate its efficiency and meantime, to create a fast intuitive and efficient integrated software package developed to speed up the whole procedure of modelling character’s costume in the animation. This method primarily focuses on, but is not limited to, animation. It also can be extended into game and garment design, and help the designer to preview the design and modify it in virtual reality before the real cloth has been made. In summary, it can be used in both film and apparel industry.

1. Research Background (30 pages)

Section 1. introduction

Section 2. Garment manufacturing research

1. Cloth design
2. Textile engineering and material science

Section 3. Human measurement

1. Landmark location
2. Measuring method
3. Measuring posture

Section 4. Computer Cloth Simulation

1. Physical cloth simulation
2. Geometric cloth simulation
3. Data-driven cloth simulation

Section 5. Conclusion

2.4.1 limitation

2.4.2 future work

1. Virtual Cloth modeling and Retargeting (60 pages)

Section 1. introduction

In this chapter, a pattern-based cloth modeling and retargeting technique is introduces. This method consist with modern made-to-measure techniques [ reference for make-to-measure ] for making bespoke cloth. It involves three steps, character body measuring, choosing the cloth pattern, resizing the pattern and assembles patterns to form a complete cloth.

In the section of human body measurement, a novel algorithm for computing geodesic is introduces which fits the special need in character measurements. It achieved linear time-complexity which never be achieved before.

In section 3, the data structure of cloth pattern and pattern database are introduces in detail.

In section 4, several methods are proposed for resizing pattern based on the measurements.

During resizing, the pattern shape is preserved and seemliness remains consist by using global optimization technique. When transfer 2D cloth pattern into 3D space, geodesic distance algorithm is used to construct string between each stitch point pairs to assemble cloth.

Section 2. Human body measurement

­­­­­­­­­­­­­­­­­­­Measurements is the 前提 of model a fit cloth. This section introduced a method that measures all the necessary data for defining a cloth accurately and efficiently.

Section 3. Cloth pattern data base

1. Cloth pattern Data structure
2. Cloth pattern Data compress method
3. Data retrieval method

Section 4. Cloth modeling

1. Definition of “style” and “accuracy”
2. Pattern mesh generation
3. Pattern resizing

Multiple criteria

Multiple objectives global optimization

Definition of error function

Evolution algorithm :

Definition of individual

Crossover method

Mutation method

Select method

Evaluation

1. 2D to 3D transfer
2. Stitching

Section 5. Experiments

Section 6. conclusion

1. Linear Time-Complexity Geodesic Algorithm (60 pages)

Section 1. Introduction

Geodesic is defined as the locally shortest path between two points on a curved space. Termed as the generalization of Euclidean straight lines, geodesic plays a fundamental role in problems relating to differential geometry. The applications of geodesics have found a widespread use in many fields, ranging from computer aided geometric design to computer animation, from robotic navigation to geography information system, and from brain flattening and warping in computational neuroscience to machine learning on manifolds. There is a rapid demand for geodesics on polyhedral surfaces that can be computed during an acceptable running time, such as surface-based brain flattening [Bartesaghi and Sapiro, 2001; and Wandell et al., 2000]. re-meshing triangulated meshes in [Peyré and Cohen, 2006], segmenting a mesh into subparts in [Katz and Tal, 2003] and mesh editing in [Lee, 1999] which employed geodesics to multi-resolution mesh modelling.

In terms of the number of source and destination points, problems on finding geodesic paths and distances can be classified into three sub-problems, (1) computing a geodesic from one source to one destination; (2) computing the geodesics from one source to all destinations; and (3) computing the geodesics from all sources to all destinations. This paper aims at the 2nd sub-problem. The pioneer work of “single source, all destinations” was presented in [Mitchell, Mount and Papadimitriou, 1987] (also called as MMP algorithm), whose time complexity is . Later, Chen and Han gave an  algorithm for exact geodesics [Chen and Han, 1990] (also known as the CH algorithm). Surazhsky et al. implemented the MMP algorithm for exact geodesics and further proposed an approximation algorithm for approximate geodesics in [Surazhsky et al. 2005]. The distinct advantage of their approximation algorithm is to reduce the running time to . However, a usual triangulated manifold is essentially inscribed in the surface of an object. The solutions of the existing exact algorithms are “exact” only with respect to the triangulated approximation of the object rather than the surface of the object. For clarification, we also refer to the “exact algorithm” as that without restriction of bounded error in this paper.

This paper presents a novel discrete geodesic computational scheme for parametric and nonparametric manifolds based on geodesic curvature flow. Consequently, three algorithms are proposed, algorithm 1 for exact geodesic computation on manifolds (please see definition below for ‘exact geodesics’), algorithm 2 for approximate geodesic computation on manifolds, and algorithm 3 for approximate geodesic computation on unorganized point sets. Given the increasingly widespread use of geodesics, our focus has been to improve its computational efficiency. Compared with the other geodesic computation algorithms, the distinct advantage of our approximation algorithm is that it can achieve linear time complexity.

The contributions of this paper include:

1. Approximation algorithm. We have developed an approximation algorithm that computes geodesics with an approximate *O*(*n*) complexity. Moreover, to assess the approximate accuracy, we give the estimation of the upper error bound at *O*(*εm*), where ε is a small ratio detailed below, and *m* denotes the number of faces covered by a geodesic path.
2. Smooth manifolds. Our proposed algorithms yield the geodesic paths which tend to the surfaces of manifolds rather than their triangulated approximations.

Most existing methods aim at polyhedral surfaces, e.g. triangle meshes. However, there have existed some data that is represented in an implicit form, e.g. medical volume data. In addition, due to the advances of 3D shape acquisition devices, point clouds are becoming fundamental shape representation. In the machine learning community, unorganized pointsets are usually a natural representation for manifold learning. The 3rd contribution of the paper therefore is,

1. Performing on parametric and nonparametric manifolds. Our algorithms can be easily extended to nonparametric manifolds for exact and approximate geodesic computation.

Although all the algorithm description and our experiments are in 3D space in this paper, the proposed algorithms are valid for any dimensional manifolds.

Section 2. Geodesic research background

[Bose et al. 2011] give an extensive literature review of the “single source shortest path” problem. Herein, we only summarize a few of the popular algorithms and focus on the time complexity and error bound. Consequently, it is expected to clearly depict the research line of pursuing the reduction of the complexity and approximate error.

To find the exact geodesics on parametric surfaces, the early work mainly focused on the convex polyhedrons. [Sharir and Schorr, 1986] presented an algorithm for exact geodesic computation with the time complexity of . Their algorithm mainly replies on the three observations, (1) any shortest path intersecting an edge of a convex polygon enters and leaves it under the same angle; (2) no shortest path on a convex polygon could pass through a vertex of the polygon except the source and destination; (3) if the faces of a convex polygon covered by a geodesic path are known, the geodesic path must be the straight line after unfolding these faces around the adjacent edges to a plane. To decrease the complexity, [Schreiber and Sharir, 2008] proposed an exact solution of the geodesics on a convex polyhedral surface that further reduces the complexity to .

For the non-convex polyhedrons, the 2nd observation in [Sharir and Schorr, 1986] is not valid yet, that is, a geodesic path may pass through the vertices of a polyhedron. To deal with this challenge, the MMP algorithm proposed in [Mitchell, Mount and Papadimitriou, 1987] employs the technique of “continuous Dijkstra” that propagates the distance information out from the source in a Dijkstra-like manner. It costs  time. Later, the CH algorithm proposed in [Chen and Han, 1990] provides an exact solution of geodesics with the complexity of . Recently, [Xin and Wang, 2009] proposed an improved version of CH algorithm and further showed the improved version running many times faster than the original CH algorithm through the numerical experiments. In fact, when a geodesic passes through the vertices on a polyhedron, the existing algorithms have to take some special treatments, e.g. pseudo-sources or additional leaves in the sequence tree. Thanks to the geodesic curvature flow, our algorithm does not suffer from this issue.

Moreover, in practice, considering the time complexity of exact geodesic computation, it is straightforward to develop an approximation algorithm for fast and accurate geodesic approximations with bounded error. [Surazhsky et al., 2005] presented an approximation algorithm for bounded geodesic approximations from one source to all destinations on a mesh. The time complexity can be expected as . The accuracy is satisfied by the relative error threshold. The other popular approximation algorithm is the fast marching method (FMM) in [Kimmel and Sethian, 1998]. The FMM algorithm carries out by solving the discrete version of the Eikonal equation over a regular grid. It costs . time as well as the above approximation algorithm. However, performing the FMM on a triangular mesh might result in the remarkable errors that are introduced by the numerical support of obtuse triangles. The accuracy essentially depends on the quality of the underlying triangulation. The near-degenerate triangulations usually lead to a big error, e.g. long and skinny triangles. The approximate error is estimated as , where  denotes the length of the longest edge and  denotes the widest angle. It can be noted that the approximate error is indeed unbounded. In practice, since the FMM is easy to implement, it is therefore applied to the nonparametric manifolds as well, such as point cloud [Mémoli and Sapiro, 2001]. The resulting geodesic is an approximation over a coarse grid built in an offset band of the point cloud surface. To restrict the discrete curves on the point cloud surface, [Hofer and Pottmann, 2004] proposed an energy minimization scheme for geodesic computation and restrict the solution on a Moving Least Squares surface. However, their method usually requires an initial estimation of the geodesic path. Our work aims at the challenging issues of the complexity and approximate error and proposes a linear time approximate algorithm with bounded error.

Section 2. Geodesic on Mesh

1. **Proposed Algorithms**

We firstly address the discrete geodesic computational scheme, i.e. geodesic curvature flow, and then outline the proposed 3 algorithms on meshes and point clouds.

* 1. **Geodesic curvature flow**

A geodesic curve has vanishing geodesic curvature everywhere. On this basis, a discrete and exact geodesic curve should have geodesic curvature zero at all of the points on it. To this end, the geodesic curvature flow is adopted mainly because the geodesic curvature flow can eliminate the geodesic curvature pointwise on a curve. For clarity, we first consider a regular surface *S* and two endpoints , between which we compute a geodesic curve on *S*. Let *C*(*s*,0) be an initial smooth parametric curve on *S* with  and , where *s* denotes the arc-length parameter of the curve. If these two endpoints are fixed, we can deform *C*(*s*,0) by the geodesic curvature flow on *S*. The curve convergences to a geodesic curve as fast as possible. The geodesic curvature flow is described as,

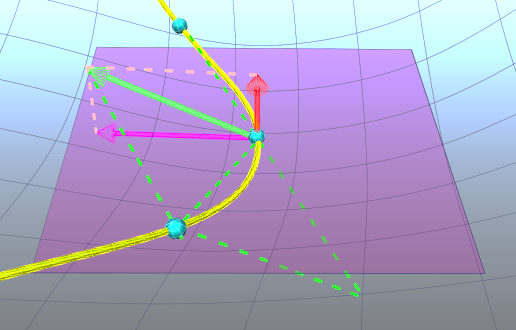
, (1)

where  denotes the geodesic curvature vector of *C*(*s*,*t*). Furthermore, the geodesic curvature vector can be expressed as,

, (2)

where,  is the 2nd order derivative of *C*(*s*,*t*) to arc-length *s*, and  is the normal vector to *S*. This flow is also known as the Euclidean curve shortening flow [Sethian, 1999] and has widely been applied in computational physics, image processing and material sciences. It is proved that the flow of Eq.(1) can eliminate the geodesic curvature pointwise on *C*(*t*) by moving *C*(*t*) in the gradient direction of the functional of the *C*(*t*)’s Euclidean length.

For the discrete case, the geodesic curve has to be approximated by a polyline containing a set of sample points. The sample’s number is fixed in general. Since we have no parametric form of the initial approximation of a geodesic curve *C*(*t*), the second order derivative of *C*(*s*,*t*) w.r.t *s* has to be approximated by using a vector triangle as shown in Fig.1. The  at point  can be estimated as,  where *∆h* denotes the interval of the successive samples. If the projection of  onto the tangent plane at  vanishes, the geodesic curvature of Eq.(2) will also vanish at  accordingly. When the geodesic curvature vanishes pointwise on *C*(*t*), the  in Eq.(1) converges to the steady state, i.e. the geodesic curve.



(*i*-1)

*Css* 

 *i*

*C* (*i*+1)

Figure 1. Illustration of vector triangle for estimating *Css*.

Moreover, we can write the discrete version of Eq.(1) in a matrix form as follows,



where, *m* denotes the number of sample points on *C*(*t*), *p* contains the coordinates of *m* sample points and *K* is a coefficient matrix. Then, the geodesic curvature of Eq.(2) can be rewritten as,



where,  denotes the normal vector at the ith sample point . As a result, the geodesic curvature flow of Eq.(1) can be rewritten as,

, (3)

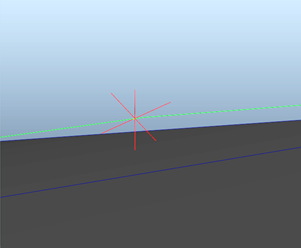
where μ denotes the iterative step length. In practice, with the two endpoints of *C*(*t*) as the constrained points  and , Eq.(3) can be converted into the following overdetermined linear system in least squares sense,

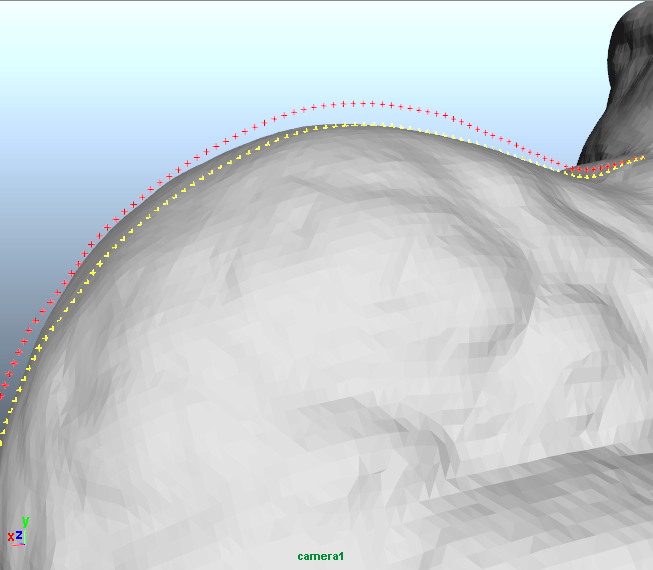
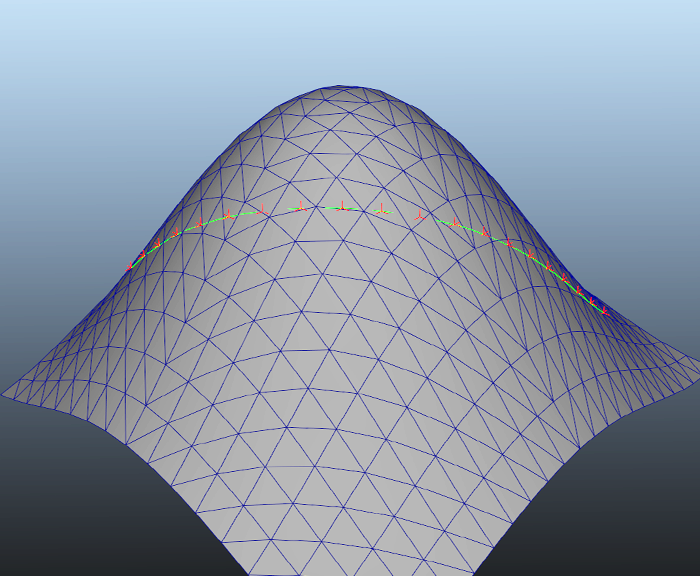
 (4)

where,  denotes an identity matrix of size 3 by 3. In general, we may enforce the positional constraints of these two endpoints by weighting the identity matrices.

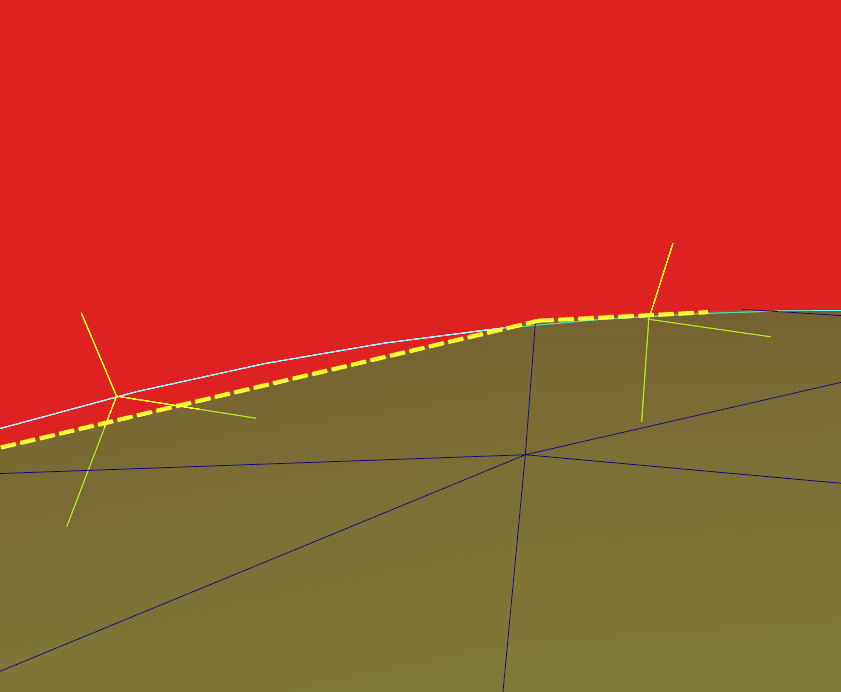
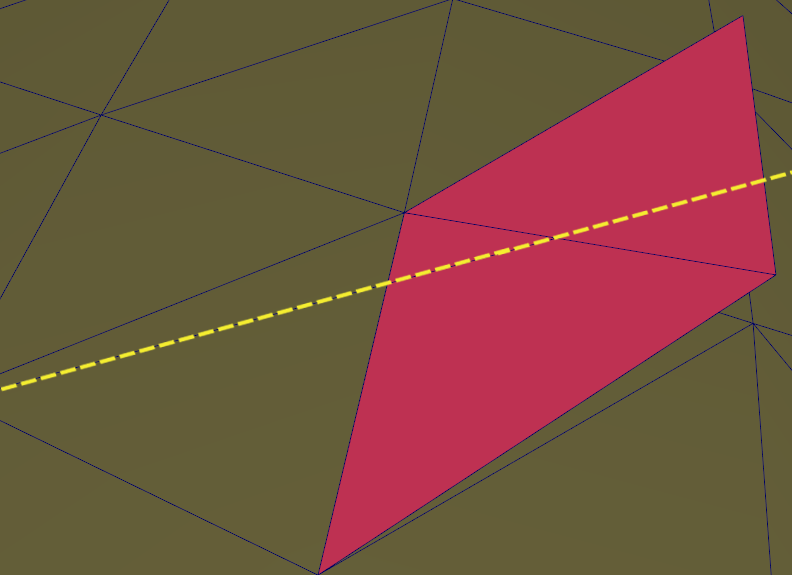
* 1. **Constraint of tangent space**

Because of the normal vectors of the sample points involved in Eq.(4), it can be noted that updating the sample points by solving Eq.(4) will result in the sample points deviating from the surfaces. This is shown in Fig.2a. The sample points are not required to be on the surfaces here. We seek the geodesic paths that accommodate the geometry of the underlying surfaces. To this end, we can restrict the sample points moving within their individual tangent spaces. This can be implemented as follows.



a. b. c.

d. e.

Figure 2. Illustration of the geodesic curvature flow performance. a) the red points are the solution of Eq.(4), while the yellow points being the sample points on the desired geodesic. It is clear that the solution of Eq.(4) deviates from the surface; b,c) show the sample points may not lie on the faces; d) shows that the plane (red) passing through the geodesic (white) cuts the faces of the mesh. The dash lines denote the cutting lines on faces; e) shows the result of unfolding the faces into a plane. The red faces are unfolded here.

Firstly, let us construct the tangent space at a sample  based on the 1-ring neighbourhood of  as,

 (5)

where, *N* consists of the normal vectors of all faces incident to , and *W* is a weight matrix and is set to an identity matrix in this paper. *B* is positive semi-definite and symmetric. Applying SVD to Eq.(5) yields, , where the eigenvalues  are all real and nonnegative, the corresponding eigenvectors  are stored in the columns of *U*. We refer to the tangent space spanned by the eigenvectors corresponding to the smallest eigenvalue of Σ.

Then, the sample points  can be updated by,

 (6)

where,  denotes the normal vector of the above-mentioned tangent space, ρ is an iterative stepsize and  denotes the projection of  onto a triangle of the mesh.

In essential, the tangent space constraint of Eq.(6) moves a point within the tangent space and therefore it tends to preserve the characterization of the underlying surfaces. Moreover, Eq.(4) can further introduce Eq.(6) as the constraint for a least-squares solution as follows,

 (7)

where,  is a column vector consisting of the coordinates of all the projections. Note that the system of Eq.(7) requires of the current locations of the sample points for computing . This is appropriate for implementation in an approximation way, that is, we may select an initial path for computing  and , and then update the samples *p* by solving Eq.(7) accordingly. Figure 2b and 2c further show the resulting geodesic lying on a surface. It can be noted that the sample points may not lie on faces. The line segment of two successive sample points must intersect with faces. Selecting such three points of one sample point and two intersection points before and after to form a plane, we can cut the related faces by this plane to generate the cutting lines on the faces. Combining all the cutting lines forms the projection of the geodesic on the triangulated meshes. This is shown in Fig.2d. Moreover, we further unfold the faces covered by the geodesic on a plane in Fig.2e. It can be noted that the cutting lines form a straight line on this plane.

**Remark**

Essentially, the solution of Eq.(7) is of an approximation to the geodesic in least squares sense. To be consistent with the existing literature, we here clarify what we mean by an ‘exact’ solution in this paper. Figure 2d and 2e indicate that the projection of the solution of the geodesic curvature flow Eq.(7) is actually the solution of existing exact algorithms (e.g. MMP exact algorithm). In fact, the solution of Eq.(7) approaches more towards the original surface of an object than its triangulated approximation. This will be further illustrated in the experiments of Section 5. In this sense, the results from Eq.(7) can be regarded as the “exact” solution and we thus call our algorithm the geodesic curvature flow based ‘exact algorithm’ in this paper.

* 1. **Outlines of the proposed algorithms**

**“Continuous Dijkstra” Strategy**

Our algorithm still employs the “continuous Dijkstra” technique. Resorting to the geodesic curvature flow of Eq.(7), the implementation is becoming very easy. Starting from one source, the distance information propagates from the 1st ring to the 2nd ring, and so on. Unlike the MMP and FMM algorithms, our method only requires the edge information instead of the accurate distance information to the source, since the accurate geodesics can be obtained by solving the system of Eq.(7). This is illustrated in Fig.3.

Let  be a source and the vertices of  on the 2nd ring and  on the 3rd ring in Fig.3. The dash lines denote the exact geodesic curves. Assume that the 3 geodesic curves of ,  and  have been computed here. When computing the geodesic between  and , it can be noted that  has 3 edges linking the 2nd ring.

We select the edge , since the included angle between  and the tangent line at  of the geodesic  is the largest compared to the other two included angles. It can be observed that compared to the two curves of the edge  plus the geodesic  and the edge  plus the geodesic , the curves of the edge  plus the geodesic  looks like straighter. Therefore, the curve of  plus the geodesic  is regarded as the initial solution of the geodesic  for the following system of Eq.(7). As a result, our algorithm does not need the “backtracing” used in the MMP, CH and FMM algorithms. This is because when adding a new vertex, we may select the “straightest” path connected to the last visited ring to form the initial estimation of the geodesic. The system of Eq.(7) can yield an exact solution of the geodesic accordingly.











Figure 3. Illustration of “continuous Dijkstra” strategy. The dash lines denote the exact geodesics derived from the source . The  is a new vertex while the neighbours .being visited.

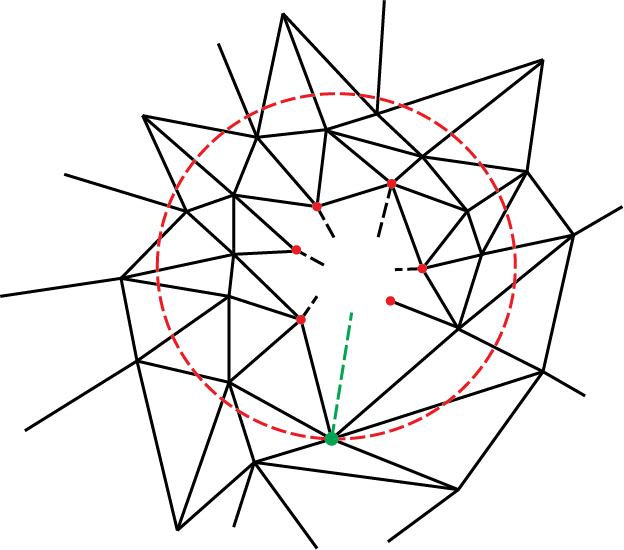


Figure 4. Illustration of the causality. The red points are visited. In the neighbours of the visited point set, the green point is the farthest to the source . The red circle indicates the range of the propagation.

**Causality**

Our method propagates information from the interior rings to external ones. However, it is required to run on arbitrary triangular meshes rather than on regular grids. There is therefore no guarantee to propagate information in such a way, that is, from smaller distances to the source to larger distances. To maintain the causality, we may view the vertices adjacent to the boundary of the visited vertex set as the neighbours. We can select the farthest vertex to the boundary from them, and further set it as the limitation that any vertices with the distance to the source less than the distance of this vertex to the source will be involved. It can be observed that the distance limitation may cover the vertices lying on the different rings. Within this limitation, our method still propagates information from the interior rings to external ones. The causality is guaranteed by this distance limitation.

Another issue is raising, that is, how to compute the distance. For convenience, we preserve the lengths of the edges on the meshes and compute the distances accordingly instead of re-computing the distances based on the exact geodesic curves. Although the resulting distances are not accurate, this does not result in significant computational complexity in our experiments, in particular the error can be estimated and controlled by adjusting the window size in our approximation algorithm. The implementation is illustrated in Fig.4.

**Algorithm Outline**

For exact geodesics, the proposed algorithm is summarized as follows.

**Algorithm 1**: Exact Geodesic Computation on Triangulated Manifolds

* Input: Triangulated manifold *S*, and a source ;
* Loop:
* Select the farthest vertex on the 1-ring neighbourhood of the visited vertex set as the distance limitation  as shown in Fig.4;
* Loop: (from the interior jth ring to the external ones within )
  + At the jth ring, select a vertex  that is not visited, and obtain the “straightest” edge linking the  and the visited vertex set as shown in Fig.3;
  + Combine this edge with the existing geodesic curve to form the initial estimation of the geodesic from  to ;
  + Solve the system of Eq.(7) to yield the geodesic of ;
  + Update the neighbourhoods for the samples  separately;
  + Until all the vertices on the jth ring within  are visited;
  + Goto the next ring, ;

Note that proceeding to solve Eq.(7), it is necessary to compute the projections  and  in terms of the current locations of the sample points. Since the sample points always move away from the triangle faces except the source and destination, we have to re-determine their 1-ring neighbourhood individually. Nonetheless, the sample points usually don’t move out of their original 1-ring neighbourhoods. For next iteration, we may update the neighbourhood for each sample point *q* in terms of their individual original 1-ring neighbours. The simple approach is to firstly seek for the nearest vertex  to the sample *q* from the *q*’s 1-ring neighbours, and then seek for the nearest vertex  to the *q* from the ’s 1-ring neighbours and so on until the nearest vertex  is reached. After that, we can compute the tangent space at the  and the *q*’s projection onto a face. The 1-ring neighbourhood of the  is regarded as the neighbourhood of the sample *q*. In our implementation, the sample points rarely move out of their 1-ring neighbourhood.

Strictly speaking, the Algorithm 1 cannot provide an exact solution. But the existing exact algorithms work on triangulated mesh that is indeed the approximation of an object. The existing exact algorithms are exact only with regard to the triangulated approximation rather than the object. To the object, our proposed algorithm essentially approximates more to the surface of the object rather than its counterpart – triangulated mesh. To distinguish from the following algorithms, we still categorize it into the exact algorithm in this paper.

The challenging issue of the Algorithm 1 is to solve the large sparse linear system of Eq.(7), which is the most time consuming. In order to speed up the proposed algorithm within some error tolerance, it is natural to employ a small-sized window to Eq.(7), so as to avoid such the large sparse matrices. When we obtain an initial estimation of the geodesic as shown in Fig.3, we may use a w-sized window to cover the last *w* sample points and then apply Eq.(7) to this small patch for the local geodesic patch. The final geodesic is formed by replacing the small patch with the resulting local geodesic one. This is summarized as follows.

**Algorithm 2:** Approximate Geodesic Computation on Triangulated Manifolds

* Input: Triangulated manifold *S*, a source  and window size *w*;
* Loop:
* Select the farthest vertex on the 1-ring neighbourhood of the visited vertex set as the distance limitation  as shown in Fig.4;
* Loop: (from the interior jth ring to the external ones within )
  + At the jth ring, select a vertex  that is not visited, and obtain the “straightest” edge linking the  and the visited vertex set as shown in Fig.3;
  + Combine this edge with the existing geodesic curve to form the initial estimation of the geodesic from  to , and then select a w-sized patch starting from  to  on this estimation;
  + Solve the system of Eq.(7) to yield the geodesic of , and then combine the geodesics  and  to form the approximate geodesic ;
  + Update the neighbourhoods of the samples  separately;
  + Until all the vertices on the jth ring within  are visited;
  + Goto the next ring, ;

Section 3. Geodesic on Point Cloud

Moreover, the above algorithm can further be extended into unorganized pointsets, e.g. point clouds. Herein, the main issue is to determine the neighbourhood for each sample point on a discrete geodesic curve. A simple method is to select the *d* nearest points as the neighbours by the ANN searching [Arya et al. 1998]. Additionally, in our case, the source may be specified in advance while the destinations being undetermined in general, since the unorganized points are always rough and bumpy. Thus, we may employ a regular grid to cover point clouds, and set the mean of the points within a specified cell of the grid as the destination node on the surface. This makes the destinations evenly distributed over the point cloud surface. Unlike the triangulated meshes, the number of the destinations should be *n*, while the unorganized points are always much more than the destinations. In fact, the regular grid can further benefit the constraint of the causality as well, since the propagation can be carried out on the regular grid instead of others. This is illustrated in Fig.5.

However, the drawback is to build up a whole grid with request for an additional  space, where *D* denotes the dimensionality of manifolds. It can be noted that this is a sparse multidimensional array. To avoid such large space complexity, the simplest approach is to store the nonempty cells into an array. Thanks to the ANN searching, we may then conveniently determine the neighbouring cells of a query cell by it. Note that the ANN searching is performed on a grid as well as on a point cloud. The algorithm for computing geodesic on unorganized pointsets is summarized as below.

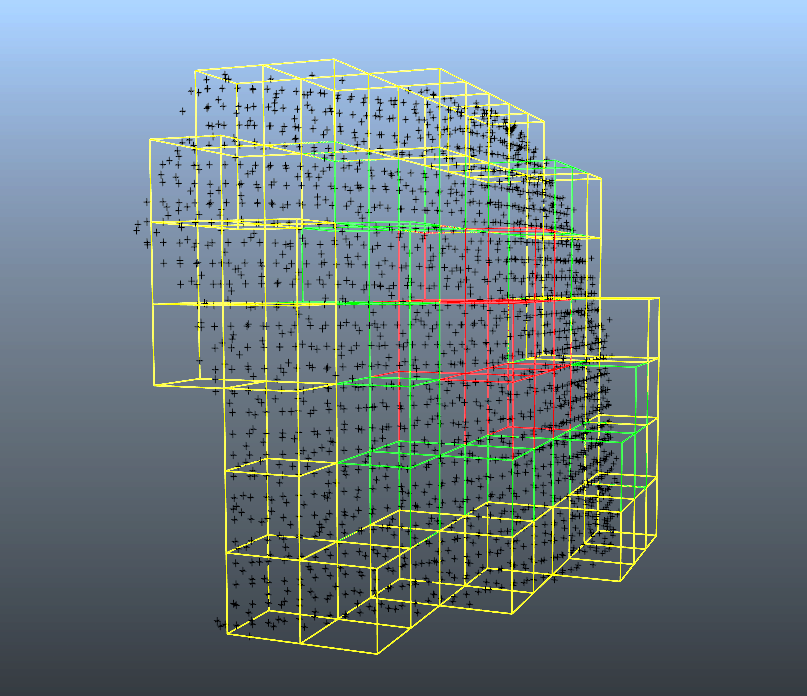


Figure 5. Illustration of a regular grid covering a point cloud. Propagation is from the red cells to the yellow ones. In practice, we store the nonempty cells in an array instead of the whole grid.

**Algorithm 3:** Geodesic Computation on Unorganized Pointsets

* Input: a Point cloud *S*, a source  within the cell , the intervals of a grid, i.e. Δ*x*, Δ*y*, Δ*z*, and the number of neighbours, *d*;
* Build up a searching tree for ANN performing on *S*, and another searching tree for ANN running on the grid;
* Loop:
* Determine the neighbouring cells of the visited cell set by ANN searching on the grid, denotes ;
* Loop: (within )
  + Set the mean of points within the ith neighbouring cell as the destination node, denotes , which is not visited;
  + Obtain the “straightest” edge linking the  and the visited cell set, i.e. the “straightest” edge to the visited destination node set, as shown in Fig.3;
  + Combine this edge with the existing geodesic curve to form the initial estimation of the geodesic from  to , and then select a w-sized patch starting from  to  on this estimation;
  + Determine the neighbourhoods of the nodes  on *S* by the ANN searching on point cloud *S*;
  + Solve the system of Eq.(7) to yield the geodesic of , and then union the geodesics  and  to form the approximate geodesic ;
  + Until all the nonempty cells are visited;

Section 4. Experiment

Chapter.5 winkle generation (40)

Section 1. textile engineering property retrieval(2 standard measurement method)

* 1. KESF system(Kawabata Evaluation System for Fabrics) (<http://158.132.122.156/portal/fom01/fom/kes.htm>)

A). Fabric Low-stress Mechanical Properties

* + 1. Tensile and Shear
    2. Pure Bending
    3. Compression
    4. Surface Characteristics
  1. FAST system(Fabric Assurance by Simple Testing)

(<http://158.132.122.156/portal/fom01/fom/fast.htm>)

* + 1. Compression
    2. Bending
    3. Extension
    4. Dimensional Stability

Section 2. Wrinkle category

Based on the KESF and FAST method, the main property of a piece of textile are: tension, shear, and compression. Thus the wrinkle can be categorized as 4 kinds(reference available in geometric cloth modeling section )

* 1. tension shear

2.2 compress

Section 3. training data acquisition and deformation detection

* 1. Definition our own feature- deformation features(scale invariant feature)
  2. Learning classifier- SVM(support vector machine or nearest neighborhood search)
  3. Detection

Chapter 6. Conclusion (5 pages)