Initial value problem of discrete geodesics and its application

Cheng, Peng

2016

Cheng, P. (2016). Initial value problem of discrete geodesics and its application. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/67025>

<https://doi.org/10.32657/10356/67025>

INITIAL VALUE PROBLEM OF DISCRETE GEODESICS AND ITS APPLICATIONS

Peng CHENG

School of Computer Engineering

A dissertation submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2016

Acknowledgments

I would like to express my deep gratitude to my supervisors Prof. Miao Chunyan and Prof. Nadia Magnenat Thalmann for providing fertile ground on which this dissertation could grow. I am grateful to them for providing me with the perfect balance of guidance and freedom to my research. The challenging problems they put forward during the discussions pushed me go further and benefited me a lot, and their encouragement and support strengthen me to keep on the research path.

I am very grateful to Prof. He Ying for his great guidance on my study for differential geometry, his insightful comments and suggestions help clarify the theoretical part of this dissertation. Many thanks to Prof. Wang Jiaye, for his role model as a researcher and professor. Thanks to the seniors: Cai Yundong, Han Xiaogang, Li Shaohua, Liu Yong and Yu Han. They gave me many useful suggestions and advices on how to do research and how to spend a Ph.D student life. I am thankful to other colleagues in LILY center: Ailiya, Wu Qiong, Meng Lei, Hao Shuji... for their kind support and assistance. I also wish to thank my friend in IMI: Dong Lu, Zhang Xiaoyan, Fang Hui, Zhang Yuzhe, Xiao Feng, Wang Anran, Yuan Quan, Zhang Wenjing, Zhu Hongyuan, and Li Qiaohong for their friendship. They have helped provide a supportive and enjoyable research environment. I would like to thank my great colleagues Wang Xiaoning, Zhang Minqi, and Wu Jiajun in Geometry Processing Group of GameLab for the constructive discussions and help on my research.

I would like to thank Prof. Tu Changhe who first introduced me the interesting research topics on discrete geodesics. I also wish to thank my friends in Shandong University Wang Lu, Gong Wenjuan, and Lu Haiyan for their friendship.

I would also like to acknowledge the Ph.D. scholarship awarded by the Institute for Media Innovation (IMI), Nanyang Technological University, Singapore. IMI has orga-

nized many seminars, workshops, and conferences, which provide each of us with opportunities to learn from and discuss with great researchers around the world. I appreciate IMI for providing excellent facilities and environment for my research. I am honored to be studying in IMI.

Last but not the least, I would like to extend my deepest gratitude to my beloved families for their lasting encouragement and support that make my heart strong and make my life beautiful. And I would also like to express my appreciation to my husband Dr. Hai Zhen for his care, love, and being always on my side.

Contents

Acknowledgments	i
List of Figures	v
List of Tables	xii
Abstract	xv
1 Introduction	1
1.1 Background and Motivations	1
1.2 Objectives and Research Scope	4
1.3 Summary of Contributions	6
1.4 Dissertation Organization	7
2 Differential Geometry Background	9
2.1 Vector Fields and Parallel Transport	9
2.2 Moving Frame	10
2.2.1 The FrenetSerret Frame on Curve	10
2.2.2 The Darboux Frame on Surface	11
2.3 Geodesics on Smooth Surface	11
2.3.1 Definition	12
2.3.2 Geodesic Equation	12
2.3.3 Geodesic Curvature	13
2.3.4 Properties	13
2.4 The Discrete Geodesic Problem	15

2.4.1	The Shortest Geodesic and the Boundary Value Problem	16
2.4.2	The Straightest Geodesic and the Initial Value Problem	17
2.5	Geodesic Coordinates	17
3	Literature Review	19
3.1	Shortest Geodesic Algorithms	19
3.2	Straightest Geodesic Algorithms	20
3.2.1	Straightest Geodesic on Parametric Surface	22
3.2.2	Straightest Geodesic on Polygon Mesh	23
3.2.3	Convergence of Straightest Geodesic	24
3.3	Constrained Texture Mapping	25
3.4	The Discrete Exponential Map	26
4	Solving the Initial Value Problem of Discrete Geodesics	27
4.1	Introduction	27
4.2	Our Algorithm	29
4.2.1	Overview	29
4.2.2	Algorithm	31
4.2.3	Tangent Computation	33
4.2.4	Parallel Transport	35
4.3	Experimental Results	35
4.3.1	Runtime Performance	35
4.3.2	Robustness	37
4.3.3	Convergence Analysis	38
4.4	Discussion	42
4.4.1	First-order ODE vs Second-order ODE	42
4.4.2	Comparison with Straightest Geodesic	44
4.4.3	Adaptive Step Length	47
4.5	Conclusion	47

5	Constrained Texture Mapping via Voronoi Diagram Base Domain	49
5.1	Introduction	49
5.2	Proposed Method	51
5.2.1	Definitions	52
5.2.2	Overview	52
5.2.3	Building Voronoi Diagram Base Domain on Surface	55
5.2.4	Local Bilinear Mapping	57
5.2.5	Discussion	58
5.3	Experimental Results and Discussion	60
5.4	Conclusion	64
6	Conclusions and Future Directions	65
6.1	Conclusions	65
6.2	Future Directions	66
6.2.1	Intrinsic Integral Invariants for Solving the Initial Value Problem of Discrete Geodesics	67
6.2.2	Constrained Texture Mapping via Optimised Voronoi Diagram Base Domain	68
	References	70
	Appendix	82
A	Derivation of Numerical Geodesic of Cubic Bezier Patch	83
A.1	General Parametric Surface	83
A.2	Cubic <i>Bézier</i> Patch	85
B	Numerical Method Computing Geodesics on Parametric Surface	89

List of Figures

1.1	Three shortest geodesic paths (the yellow, brown, and green paths) between two points [PS06].	2
1.2	An initial value problem geodesic example: wireframe sculpture on Rhino surface generated by a single geodesic path of our proposed method. . . .	3
1.3	Initial value problem geodesic path encodes the intrinsic property of surface locally around a vertex.	4
2.1	Geodesic curvature. Let $\mathbf{p}$ be a point on a smooth surface S and $\mathbf{N}$ the normal at p . Two curves γ_1 and γ_2 are tangential at $\mathbf{p}$ and their tangent direction is $\mathbf{T}$. The tangent $\mathbf{T}$, the normal $\mathbf{N}$, and the binormal $\mathbf{B}$ define a orthonormal coordinate basis at $\mathbf{p}$. The curve γ_1 is a geodesic at p , since $\ddot{\gamma}_1(s)$ is parallel to $\mathbf{N}$, where s is the arc-length parameter. However, γ_2 is not a geodesic, since $\ddot{\gamma}_2$ has a non-zero component on $\mathbf{B}$, i.e., the geodesic curvature $\kappa_g = \ddot{\gamma} \cdot \mathbf{B} \neq \mathbf{0}$ is not vanishing.	14
2.2	On a smooth surface, there is a uniquely determined geodesic passing through a given point $\mathbf{p}$ in an arbitrary tangent direction $\vec{\omega} \in T_{\mathbf{p}}$. Here we illustrate two geodesics γ_1 and γ_2 on a torus, corresponding to two tangent directions $\vec{\omega}_1$ and $\vec{\omega}_2$. Note that the curve γ_2 can return to the starting point $\mathbf{p}$, whereas γ_1 cannot.	15
2.3	Shortest geodesic and straightest geodesic on an eight surface. The red curve is the shortest geodesic path between the two green vertices s and t . The blue curve is the straightest geodesic defined by the vertex s and a tangent direction at s . The tangent direction is computed at vertex s as the tangent direction of the red curve.	16

3.1	(a) A straightest geodesic has equal left and right curve angles $\theta_l = \theta_r$. (b) The shortest geodesic between s and t cannot be extended through a spherical vertex p , since making it off p shortens the length. (c) Unfold the spherical vertex p to $\mathbb{R}^2$	22
4.1	The discrete <i>shortest</i> geodesic does not solve the initial value problem. Let v be a saddle vertex, whose cone angle is more than 2π . When a shortest geodesic path, say $\gamma(s, v)$, passes through v , it splits into many outgoing geodesic paths: any line segment $\overline{pv}$ in the fan-shaped area (in grey) together with $\gamma(s, v)$, is a shortest path from s to p . Therefore, the initial value problem does not have a unique solution, if one considers the <i>shortest</i> geodesic paths.	28
4.2	Our method solves the initial value problem of discrete geodesics. For each point (in red) on the Rhino model, we compute geodesic paths in 60 tangent directions, which are evenly sampled on the tangent plane. Each tangent direction is guaranteed to produce a unique geodesic path on the triangle mesh. Our method is numerically stable and works well on meshes with large and/or skinny triangles.	30
4.3	A curved PN-triangle is a three-sided cubic Bézier patch that interpolates both the positions and normals of the three vertices of a planar triangle.	31
4.4	A typical iteration in our algorithm.	32
4.5	A curved PNG1-triangle is G^1 continuous with its three neighboring patches. The patch is obtained by blending two cubic Bézier patches for each triangle edge (see (a)). For patches that meet at a vertex v , control points corresponding to the vertex are all projected to the tangent plane at v (see (b)). Thus, the tangent direction remains unchanged when the geodesic crosses the edge boundary or passes through a vertex. As a result, we can continue to solve the first-order ODE of the tangent on the neighboring patch (see (c)).	36
4.6	The time complexity is inversely proportional to the step length δ . The vertical axis shows the running time of computing 60 geodesic paths of length $\sqrt{3}$ at 100 random source points. The time for constructing the PNG1-triangles, which is dependent on mesh complexity, is excluded. . .	37

4.7	We solve the initial value problem on a wide range of 3D models and we assign each source point 60 evenly distributed tangent directions. Note that our method works well on highly curved regions as well as sharp features. The images are generated at high resolution to allow for zoom-in examination.	38
4.8	Solving the initial value problem on the Rhino model with two different length constraints. (see (a) and (b)). Wireframe sculpture made with initial value geodesic path with different length. (see (c) and (d)). Note that the source point is located at the belly button and different sculpture with different source point or initial tangent direction.	39
4.9	Given the user-specified point (green), tangent direction and length, our algorithm computes the unique solution of the initial value problem. . .	40
4.10	Our method also works for the non-orientable surfaces.	40
4.11	Computing local parameterization using our method. The user specifies a point $\mathbf{p}$ on the 3D mesh and a radius r . We then apply our method to compute 60 geodesic paths passing through $\mathbf{p}$ and having length r . Each geodesic path uniquely corresponds to a radial line of a polar coordinate system defined on $T_{\mathbf{p}}$, allowing us to build a polar coordinate system on 3D meshes. We visualize the parameterization by checkerboard texture mapping on a rectangular patch. The bottom row shows the parameterized patch on $\mathbb{R}^2$	41

- 4.12 Convergence test on four synthetic models. Sphere $(\cos u \sin v, \sin u \sin v, \cos v)$, $u \in [0, 2\pi]$, $v \in [0, \pi]$; Ellipsoid $(2 \cos u \sin v, 2 \sin u \sin v, 3 \cos v)$, $u \in [0, 2\pi]$, $v \in [0, \pi]$; Paraboloid $(2v^2, 4v, 6u)$, $u, v \in [-1, 1]$; Hyperboloid $(2 \cos v \sqrt{1+u^2}, 2 \sin v \sqrt{1+u^2}, 4u)$, $u \in [-1, 1]$, $v \in [0, 2\pi]$; Torus $((3 + \cos v) \cos u, (3 + \cos v) \sin u, \sin v)$, $u \in [0, 2\pi]$, $v \in [0, \pi]$. When the mesh resolution is sufficiently high, the Hausdorff distance between the PNG1-triangles $\widehat{M}$ and the smooth surface S approaches zero, meaning that the curved PNG1-triangles converge to the smooth surfaces S . We also observe that the discrete geodesic $\gamma \in M$ converges to the geodesic $\widehat{\gamma} \in \widehat{M}$, since both the length difference $|\text{len}(\gamma) - \text{len}(\widehat{\gamma})|$ and the area of the gap between γ and $\widehat{\gamma}$ approach zero. The horizontal axis h is the average edge length of M 43
- 4.13 The straightest geodesic [PS06] (red) which is highly dependent on the mesh triangulation and resolution, converges to the ground truth (blue) only when the mesh resolution is sufficiently high. Our method with step length $\delta = \frac{h}{16}$ produces very good result (green) even when the mesh resolution is very low. Note that all geodesics are given the same initial condition. 45
- 4.14 Comparison of our method (green) to straightest geodesic (red) and analytic geodesic (blue) on the torus model with 600 faces. Note that all geodesics are with the same initial condition. We set the step length $\delta = \frac{h}{2}, \frac{h}{4}, \frac{h}{8}$ and $\frac{h}{16}$, respectively, where h is the average edge length. We can clearly see that our result converges to the ground truth when the step length δ approaches zero. We evaluate the quality of a geodesic curve by measuring the area between it and the ground truth. As shown in (e), the gap is proportional to the length of the geodesic (the horizontal axis). . . 46
- 5.1 Texture mapping results on mask model(499,955) of our proposed method with only four constraints of "eyes", "nose" and "mouse". All the textures are from Beijing opera faces. (a,e) A military general and regent of the state of Shu Han; (b,f) A warlord in the late Sui dynasty; (c,g) One of the four ugly in ancient China; (d,h) A military general between the late Eastern Han dynasty and early Three Kingdoms. 51

5.2	The workflow of the proposed Voronoi Diagram base domain method for constrained texture mapping. (a) The input mesh surface; (b) The mesh surface boundary (red curve), point constraints (green spheres) and corresponding Voronoi Diagram in the texture image; (c) The texture image plane boundary (red curve), point constraints (green spheres), and corresponding Voronoi Diagram on mesh surface; (d) The parametrization result; (e), (f) Different views of texture mapping results.	54
5.3	Tags of each vertex indicated with different colors. (a),(b) The tags on a mask model (499,955) in front and side views; (c),(d) The tags on a refined mask model (31762,62467). Red and blue colors correspond to the Voronoi cells of both left and right eyes, yellow color indicates nose, and the green color corresponds to mouth.	55
5.4	The exponential map at a point v (red sphere) projects geodesic curves (blue lines) originating at v in (a) to straight vectors, which emanate from from the origin of the tangent plane T_p . (b) Iso-contours (green lines) of the geodesic distance.	57
5.5	Geodesic distance field on refined mask model (31762,62467).	58
5.6	Piecewise scale of angle for each pair of the Voronoi diagrams.	59
5.7	Texture mapping results via our proposed method on Beijing Opera faces. (a) Example of the texture image and constraints (green spheres); (b) The mask mesh surface (499,955) and constraints (green spheres); (c), (e), (f), (g), (h), (i) Front views of the texture mapping results; (d),(j),(k),(l),(m),(n) Side views of the texture mapping results.	61
5.8	Results of proposed method with checkerboard texture. (a) Input checkerboard image, constraints (green sphere), and boundary(red curve); (b) Mesh surface, constraints(green sphere), and boundary(red curve); (c) Parametrization result; (d), (e) Different views of texture mapping results.	62
5.9	Results of proposed method with curve constraints. (a) Texture image with curve constraints(green lines); (b) Mesh surface with curve constraints(green lines)(c) Parametrization result; (d) Texture mapping result.	62

5.10	Comparison with constrained harmonic map. (a) Texture image and constraints; (b) Mask surface and constraints; (c) Texture mapping results with constrained harmonic mapping; (d) Texture mapping result with proposed method; (e) Parametrization result of constrained harmonic mapping; (f) Parametrization result of proposed method.	63
6.1	Geodesic disk around a point on triangular mesh.	68

List of Tables

4.1	The statistics of surfaces that converge to its smooth counterpart.	44
4.2	Fixed step length vs adaptive step length. We compute 60 geodesic paths evenly emanating from 100 randomly selected source points and having the length $\sqrt{3}$. The running time t is measured in seconds. δ is the step length, h is the average edge length and ΔL measures the length difference of the geodesic on the curved PNG1-triangles and the discrete geodesic on the piecewise flat surfaces.	48

Abstract

Computing geodesic paths and distances is a common operation in computer graphics and computer-aided geometric design. The existing discrete geodesic algorithms are mainly designed for solving the boundary value problem, i.e., finding the shortest path between two given points. However, the *shortest* geodesic paths do not simulate properties of geodesics on smooth surface, for example, the *shortest* geodesic paths define neither the discrete geodesic curvature nor the parallel transport of vector fields. When a *shortest* geodesic path passes through a saddle vertex, it does not provide a unique solution on triangle meshes. In this thesis, the initial value problem, i.e., finding a uniquely determined geodesic path from a given point in any direction, is studied.

Firstly, a first-order tangent ODE method for solving the initial value problem is proposed. The proposed method solves the initial value problem in an indirect manner. In particular, given a fixed point and an initial tangent direction on a triangle mesh M , we first compute a geodesic curve $\hat{\gamma}$ on a piecewise smooth surface $\widehat{M}$, which well approximates the input mesh M and can be constructed at little cost. Then, we solve a first-order ODE of the tangent using the fourth-order Runge-Kutta method, and parallel transport it along $\hat{\gamma}$. When the geodesic curve reaches the boundary of the current patch, its tangent can be transported to the neighboring patch, since the piecewise smooth surface $\widehat{M}$ is constructed with G^1 -continuity along the common boundary of two adjacent patches. Finally, once the geodesic curve $\hat{\gamma}$ is available, we project it onto the underlying mesh M , producing the discrete geodesic path γ , which is guaranteed to be unique on M . It is worth noting that our method is different from the conventional methods of directly solving the geodesic equation (i.e., a second-order ODE of the position) on piecewise smooth surfaces, which is difficult to implement due to complicated representation of

the geodesic equation involving Christoffel symbols. The proposed method, based on the first-order ODE of the tangent vector, is intuitive and easy to implement. Our method is particularly useful for computing geodesic paths on low-resolution meshes which may have large and/or skinny triangles, since the conventional *straightest* geodesic paths are usually far from the ground truth.

Moreover, we employ the initial value problem geodesic to solve the constrained texture mapping problem. Constrained texture mapping builds extra correspondence between mesh features and texture details. However, to match each feature between texture image and mesh surface, many constraints are required to be satisfied. Manually specifying and editing such constraints is time-consuming and is thus not applicable for practical problems. We propose a Voronoi diagram based domain method to address the positional constrained texture mapping problem. Specially, the first step is to generate a Voronoi diagram on the mesh surface via taking user-defined feature points as Voronoi sites. Then, we build a Voronoi diagram with corresponding feature points on the image plane. Finally, we create an exponential mapping between every pair of Voronoi cells. The proposed method provides a valid one-to-one mapping, which not only satisfies user-defined constraints but also preserves the metric structure of the original mesh. It allows high-level semantic constraints. For example, in order to map a human face image onto a mask surface, we only need four naturally meaningful constraints of “eyes”, “nose”, and “mouth”.

Chapter 1

Introduction

1.1 Background and Motivations

Computing geodesic distances and geodesic paths has been widely studied in computational geometry and geometric modeling community and many other fields, such as CAD/CAM [SRD08], path planning [DMT91], and shape analysis [SRIC01]. As one of the fundamental problems in the study of computational geometry, the discrete geodesic problem has been studied extensively in the past decades. For the discrete shortest geodesic problem or the boundary value problem, the typical exact algorithms are the Mitchell-Mount-Papadimitriou (MMP) algorithm [MMP87] and the Chen-Han (CH) algorithm [CH90]. Except for the exact algorithms, previous studies have developed approximate algorithms. A typical approximate algorithm is the fast marching method [KS98], which is widely used in image processing. The discrete straightest geodesics are particularly useful in visualization and simulation [PS99, LKMD01] in computer graphics. To our knowledge, the discrete straightest geodesic problem receives less attention than the shortest geodesic, where representative methods for the problem include the intrinsic curve angle method [PS06] and the sectional line method [RKSDH⁺03].

Geodesics on smooth surfaces are well defined and studied in classic differential geometry. However, due to the discontinuity of surface curvatures, the discrete geodesic

problem, i.e., computing geodesic distances and paths on discrete domains such as polygonal meshes, is fundamentally different from its smooth counterpart, and has not been well studied. The discrete *shortest* geodesic is not equivalent to the discrete *straightest* geodesic, which bisects the vertex angles. For example, geodesics is both straightest and locally shortest on smooth surfaces, but such a nice property does not hold on polygonal meshes. The discrete shortest geodesic is widely adopted for many application in computer graphics community, as it is locally shortest path and provides a metric on polygonal surfaces. However, the discrete shortest geodesic does not simulate the properties of geodesics on smooth surface, such as vanishing geodesic curvature, the Gauss-Bonnet theorem and parallel transport of vector fields. In addition, the discrete shortest geodesic does not provide a unique solution through a saddle vertex, and thus we cannot trace the path of a moving particle along mesh surface by using the shortest geodesic. See Figure 1.1.

Figure 1.1: Three shortest geodesic paths (the yellow, brown, and green paths) between two points [PS06].

To robustly and accurately approximate geodesics on mesh surface, a first-order ODE method for solving the initial value problem on triangle meshes is proposed. Given a fixed point and an initial tangent direction on a triangle mesh M , we first compute a geodesic

curve $\hat{\gamma}$ on a piecewise smooth surface $\widehat{M}$, which well approximates the input mesh M and can be constructed at little cost. Then, we solve the ODE of the unit tangent vector by the fourth-order Runge-Kutta method, and parallel transport it along $\hat{\gamma}$. Due to the G^1 -continuity along the common boundary of two adjacent patches, we can transport the tangent to neighboring patch when the geodesic path reaches the boundary of current patch. Finally, we project the geodesic curve $\hat{\gamma}$ on piecewise smooth surface onto the underlying mesh. The projected discrete geodesic path γ , is guaranteed to be unique on the triangle mesh M . Figure 1.2 plots a wireframe sculpture of rhino generated by one geodesic path of our method.

Figure 1.2: An initial value problem geodesic example: wireframe sculpture on Rhino surface generated by a single geodesic path of our proposed method.

Generally, our method is different from the conventional methods, which directly solve the geodesic equation on the piecewise smooth surface. The geodesic equation is a second-order ODE of the position and involves Christoffel symbols as parameters. Therefore, it requires C^2 smooth surface construction, which is tedious and difficult to implement. In contrast, based on the first-order ODE of the tangent vector, the proposed method is intuitive and easy to implement. We observe that our result converges to the smooth counterpart with respect to the measure of “gap” length and “gap” area with the ground truth on quadratic surfaces and torus surface. Our method is particularly useful for

computing geodesic paths on low-resolution meshes with large and/or skinny triangles, where the conventional *straightest* geodesic paths are usually far from the ground truth. Moreover, we see the benefits of our method as a shape descriptor to distinct shape locally. See Figure 1.3.

Figure 1.3: Initial value problem geodesic path encodes the intrinsic property of surface locally around a vertex.

1.2 Objectives and Research Scope

Geodesic on smooth surfaces is an intrinsic property of surface in terms of distance metric, geodesic curvature, and parallel transport of vector field. However, the discrete geodesic problem is fundamentally different from its smooth counterpart, due to the difference between smooth and discrete domains. For the discrete shortest geodesic, the gap lies in that it does not simulate the differential geometry property of smooth geodesic on surface. Moreover, the discrete shortest geodesic does not provide a unique solution on mesh. For the discrete straightest geodesic, the gap is that it does not approximate smooth geodesic well on a coarse mesh or meshes with noise. This research aims to fill the gap between discrete geodesics and smooth geodesics on surface. In this section, we summarize our research objectives in two aspects: a robust and accurate method for

solving the initial value problem of discrete geodesics and how the method benefits the constrained texture mapping problem.

Solving the Initial Value Problem of Discrete Geodesics. In general, a robust and accurate method for solving the initial value problem is required to satisfy the following criteria:

- It provides a unique solution for each instance of the geodesic path. Assuming a particle travels on surface and subjects to no any forces except for a force perpendicular to the surface, its path on the surface should be determined only by the intrinsic property of the surface. Therefore, we would be able to trace its path uniquely.
- The output geodesic path is a good approximation of its smooth counterpart.
- An appropriate metric is required to measure the quality of the approximated discrete geodesics.

Note that the straightest geodesic [PS06] is already a good approximation on a mesh with dense sample and good triangulation. In contrast, a robust method should be able to deal with the initial value geodesic problem on coarse-grained mesh surfaces, i.e. a mesh with poor triangulation, or with noise in geometry or topology. In this dissertation, a first order tangent ODE method is proposed to uniquely solve the initial value problem of discrete geodesics. The approximation quality of the proposed method is measured by using the “gap” length and area between our results and the ground truth on smooth quadratic surfaces and torus surface.

Constrained Texture Mapping via Voronoi Diagram Base Domain. One of the key challenges for the constrained texture mapping problem is to ease the tedious constraints editing work. We propose to solve the constrained texture mapping problem

by employing one point or curve constraint for each naturally meaningful feature, e.g., a nose. The detailed strategy is to implement each constraint not only globally as an overall constraints unit but also locally as a single constraint independently. By “globally”, we mean that the Voronoi diagrams on both the texture image and mesh surface control the consistence between constraints, while by “locally” we refer to that a discrete exponential mapping isometrically encodes the mesh surface property, especially in the area near the constraint. As our initial value problem geodesic path inherits the intrinsic property of the surface it lies in, the discrete exponential map from initial value geodesics can better encode the local surface property compared to the shortest geodesics.

1.3 Summary of Contributions

The research presented in this dissertation has made major contributions to solve the initial value problem of discrete geodesics and apply it to the constrained texture mapping problem. The contributions are summarized as follows.

- We proposed a first-order tangent ODE method to solve the initial value problem of discrete geodesics. The proposed method achieves better approximation of geodesics on smooth surface with only G^1 continuity piecewise smooth surface. As the only coefficient of the tangent ODE is surface normal, the proposed method has no need to compute the higher order Christoffel symbols. Note that conventional methods solve the geodesic equation directly on the piecewise smooth surface. Therefore, the surface should be constructed with C^2 continuity, for that the geodesic equation is a second-order ODE of position, which involves Christoffel symbols. As for shortest geodesic on surface, it does not simulate the differential property of geodesics on smooth surface. Moreover, the shortest geodesic does not provide unique solution when through a saddle vertex.

- To evaluate the proposed first-order tangent ODE method, we construct a series of seven meshes for each of the four quadric mesh surfaces and the torus surface, which converge to their smooth counterpart. We measure the difference of the results of our method against the corresponding smooth geodesic using both the “gap” length and “gap” area. Experimental results show that our method results converge to the smooth counterpart under both measures when the mesh surface converges.
- We propose a Voronoi diagram base domain method for constrained texture mapping problem. Our method is able to achieve good texture mapping result with only one point or curve constraint for each naturally meaningful feature. Previous studies used the global energy minimization or image wrapping method for the constrained texture mapping problem. Then, to match a single feature between texture image and mesh surface, many constraints are often required to cope with, which could be time-consuming. In contrast, the proposed method does not suffer from the constraints editing problem. Moreover, it is guaranteed to produce foldover free mapping.

1.4 Dissertation Organization

The rest of this dissertation is organized as follows. Chapter 2 presents the differential geometry background of discrete geodesics. Chapter 3 reviews related work on both initial value problem and boundary value problem of discrete geodesics. Chapter 4 reports the first-order tangential ODE method for solving the initial value problem of discrete geodesics, including convergence experiments and comparison with straightest geodesics. Chapter 5 presents the Voronoi diagram based method for constrained texture mapping. Chapter 6 concludes the dissertation and present future directions on solving the initial value problem of discrete geodesics and its applications.

Chapter 2

Differential Geometry Background

In this chapter, we provide an introduction to geodesics in a differential geometrical view. We first present the moving frame and parallel transport, which will be employed for solving the initial value problem of discrete geodesics in Chapter 4. Then we introduce the definition and properties of geodesics on smooth surface. After that, we present the discrete geodesics problem on piecewise linear polygon mesh surface and discuss how the discrete case is different from the smooth case. Finally, we describe the geodesic coordinates, which will be used for the initial value problem geodesic application of constrained texture mapping in Chapter 5.

2.1 Vector Fields and Parallel Transport

Generally, a vector field assigns a vector to each point in a subset of space. This section presents specifically vector field and parallel transport along a parameterized curve embedded in surface.

For a parameterized curve γ in R^3 , let I be an interval of real numbers, a vector field is a function which assigns a vector $\mathbf{X}(t)$ to each $t \in I$ at the point $\gamma(t)$, i.e. for all $t \in I$, we have $\mathbf{X}(t) \in R^3_{\gamma(t)}$. For example, the velocity γ' of the parameterized curve γ is a vector field and its length $\|\gamma'\|$ is the speed of γ .

To introduce the concept of parallel transport, we first define covariant derivative of a vector field. Let $w(t)$ be the vector field along a parameterized curve γ embedded in surface S , for a point p in γ , the vector obtained by using the normal projection of $\frac{dw}{dt}$ onto the plane $T_p(S)$ is called the covariant derivative at p of the vector field w . A vector field w along a parameterized curve γ is parallel if $\frac{dw}{dt} = 0$ for every $t \in I$.

In terms of intrinsic geometry, geodesic is the curve with its tangent vector $\gamma'(t)$ being parallel along γ .

2.2 Moving Frame

To study the extrinsic geometry of smooth surface embedded in a homogeneous space, moving frame defines an ordered basis of vector space. This section presents the Frenet frame defined on space curve and the moving Darboux frame defined on surface.

2.2.1 The FrenetSerret Frame on Curve

Let $\gamma(s)$ be a space curve with arclength parameter s , $\mathbf{T}$ be the unit tangent vector, $\mathbf{N}$ be the unit normal vector, and $\mathbf{B} = \mathbf{T} \times \mathbf{N}$ be the unit binormal vector. The three unit vectors $\mathbf{T}$, $\mathbf{N}$, and $\mathbf{B}$ together forms the FrenetSerret frame, which provides an orthogonal basis spanning the Euclidian space R^3 . The FrenetSerret formulas describe the derivatives of the FrenetSerret frame in terms of the curvature κ and torsion τ of γ . See Equation 2.1.

$$\begin{cases} \frac{d\mathbf{T}}{ds} = \kappa\mathbf{N}, \\ \frac{d\mathbf{N}}{ds} = -\kappa\mathbf{T} + \tau\mathbf{B}, \\ \frac{d\mathbf{B}}{ds} = \tau\mathbf{N}, \end{cases} \quad (2.1)$$

2.2.2 The Darboux Frame on Surface

For any non-umbilic point p of a surface embedded in Euclidean space, the Darboux frame is defined on every tangential direction $\mathbf{T}$ of its tangent space T^p .

Let S be an oriented surface embedded in R^3 , p be a non-umbilic point on S . The Darboux frame is defined on every tangential direction in its tangent space T^p . The Darboux frame with respect to a tangential direction $\mathbf{T}$ consists of a triple of unit vectors $\mathbf{T}, t, u$, where T is the tangent, u is the normal, and t is the tangent normal. The triple is defined by the embedded curve $\gamma(s)$ through p in the direction $\mathbf{T}$, and parameterized by arc length s :

$$\begin{cases} \mathbf{T} = \gamma'(s), \\ u = u(\gamma(s)), \\ t = u \times \mathbf{T}, \end{cases} \quad (2.2)$$

The Darboux frame is a positively oriented orthonormal basis that defines on each point of the curve, and thus forms a natural moving frame along the embedded curve. Applying the Frenet-Serret formulas, we have the differential of the Darboux frame as in Equation 2.3.

$$\begin{cases} \frac{d\mathbf{T}}{ds} = \kappa_g t + \kappa_n u, \\ \frac{dt}{ds} = -\kappa_g \mathbf{T} + \tau_{\mathbf{r}} \mathbf{u}, \\ \frac{du}{ds} = -\kappa_n \mathbf{T} - \tau_{\mathbf{r}} t, \end{cases} \quad (2.3)$$

2.3 Geodesics on Smooth Surface

Assuming a bug lives in the surface without knowledge of the embedding space, geodesics are the curves that the bug perceives to be straight. Further assuming a particle moves on the surface, and subjects to no forces except for a force that is perpendicular to the

surface, the particle would move along a geodesic. This section presents the background of geodesics in differential geometry [DC76, AZ67][Spi70].

2.3.1 Definition

The term “geodesic” comes from geodesy. Originally, a geodesic was the shortest route between two points on the Earth’s surface. In differential geometry, geodesic is the generalization of the notion *straight line on curved surface*. The following three equivalent properties in Definition 2.1 formally define geodesic.

Definition 2.1 *Let S be a smooth two-dimensional surface, and $\gamma(s) : [a, b] \rightarrow S$ be a curve on S parameterized by arc-length s . The curve γ is a geodesic if one of the equivalent properties holds:*

- (i) γ is a locally shortest curve.
- (ii) γ'' is perpendicular to the surface, i.e. parallel to its normal.
- (iii) γ has vanishing geodesic curvature.

2.3.2 Geodesic Equation

In terms of Riemannian geometry, geodesic is a locally distance minimizing curve. Let $p, q \in S$ be two points on S , and g be the metric tensor of S . The geodesic $\gamma(p, q)$ between p and q is locally distance minimizer, which can be obtained by minimizing the following energy functional

$$E(\gamma) = \frac{1}{2} \int_p^q g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t)) dt.$$

where t is a real number parameter of the geodesic curve γ . The Euler-Lagrange equation yields the geodesic equation

$$\frac{d^2 x^k}{dt^2} + \Gamma_{ij}^k \frac{dx^i}{dt} \frac{dx^j}{dt} = 0, \quad (2.4)$$

where $x^1(t)$ and $x^2(t)$ are the local coordinates of $\gamma(t)$, and Γ_{ij}^k are the Christoffel symbols of the second kind.

2.3.3 Geodesic Curvature

From an intrinsic point of view, geodesic is a *straightest* curve that its tangential acceleration only happens in the direction of the normal of the surface it lies in. The curve γ passes through a point $p \in S$, with tangent vector $\mathbf{T} = \dot{\gamma}$, normal vector $\mathbf{N}$, and binormal vector $\mathbf{B} = \mathbf{T} \times \mathbf{N}$. The curvature vector $\ddot{\gamma}$ is perpendicular to the tangent $\mathbf{T}$, i.e., $\ddot{\gamma} \perp \dot{\gamma}$. The normal curvature κ_n at the point p is the amount of the curve's curvature in the direction of the surface normal, i.e., $\kappa_n = \ddot{\gamma} \cdot \mathbf{N}$. The geodesic curvature κ_g is defined by projecting $\ddot{\gamma}$ onto the binormal vector $\mathbf{B}$, i.e., $\kappa_g = \ddot{\gamma} \cdot \mathbf{B}$ (See Figure 2.1). The curvature of the curve γ is related to both κ_g and κ_n , $\kappa = \|\ddot{\gamma}\| = \sqrt{\kappa_n^2 + \kappa_g^2}$. The curve γ is a *geodesic*, if its geodesic curvature vanishes $\kappa_g = 0$. The geodesic curvature measures how far the curve is from being a geodesic. A geodesic has vanishing geodesic curvature, meaning that it appears to curve in the ambient space whenever the surface curves and does not curve *within* the surface.

2.3.4 Properties

The properties of geodesics on smooth surface are summarized as follows.

- (i) On smooth surfaces, a geodesic is both locally shortest and straightest.
- (ii) A curve γ on surface is geodesic if and only if its normal is parallel to the surface normal everywhere.
- (iii) Given any two points on surface, there is at least one geodesic curve passing through the two points.

Figure 2.1: Geodesic curvature. Let $\mathbf{p}$ be a point on a smooth surface S and $\mathbf{N}$ the normal at p . Two curves γ_1 and γ_2 are tangential at $\mathbf{p}$ and their tangent direction is $\mathbf{T}$. The tangent $\mathbf{T}$, the normal $\mathbf{N}$, and the binormal $\mathbf{B}$ define a orthonormal coordinate basis at $\mathbf{p}$. The curve γ_1 is a geodesic at p , since $\ddot{\gamma}_1(s)$ is parallel to $\mathbf{N}$, where s is the arc-length parameter. However, γ_2 is not a geodesic, since $\ddot{\gamma}_2$ has a non-zero component on $\mathbf{B}$, i.e., the geodesic curvature $\kappa_g = \ddot{\gamma} \cdot \mathbf{B} \neq 0$ is not vanishing.

- (iv) For a developable surface, if and only if geodesics on the surface are unfolded into straight lines, when the surface is unfolded to a plane.

Furthermore, a geodesic is fully determined by an end point and a given tangent direction. See Figure 2.2.

Theorem (Existence & Uniqueness) *Given a point $p \in S$ and a vector $\vec{w} \in T_p$, $v \neq 0$, there exists an $\epsilon > 0$ and a unique parameterized geodesic $\gamma : (-\epsilon, \epsilon) \rightarrow S$ such that $\gamma(0) = p$, and $\gamma'(0) = \vec{w}$.*

Figure 2.2: On a smooth surface, there is a uniquely determined geodesic passing through a given point $\mathbf{p}$ in an arbitrary tangent direction $\vec{w} \in T_{\mathbf{p}}$. Here we illustrate two geodesics γ_1 and γ_2 on a torus, corresponding to two tangent directions $\vec{w}_1$ and $\vec{w}_2$. Note that the curve γ_2 can return to the starting point $\mathbf{p}$, whereas γ_1 cannot.

2.4 The Discrete Geodesic Problem

Geodesics on smooth surface are well defined with three equivalent properties. See Section 2.3.4. In contrast, on triangular surfaces, the locally shortest property of geodesics leads to the shortest geodesic path, while the property that the curve normal is parallel to

the surface normal leads to the straightest geodesic path. In terms of geodesic equation, the shortest geodesic solves the boundary value problem, while the straightest geodesic solves the initial value problem of geodesic equation.

Figure 2.3: Shortest geodesic and straightest geodesic on an eight surface. The red curve is the shortest geodesic path between the two green vertices s and t . The blue curve is the straightest geodesic defined by the vertex s and a tangent direction at s . The tangent direction is computed at vertex s as the tangent direction of the red curve.

2.4.1 The Shortest Geodesic and the Boundary Value Problem

The shortest geodesic path is born discrete. The pioneer work is computing the "single-source-all-destination" shortest path on convex polyhedral by Sharir and Schorr [SS86]. On a triangular mesh, the shortest path satisfies the shortest property of smooth geodesic and referred as shortest geodesic. The shortest geodesic has several properties which make it widely useful in computer graphics community. First, the shortest geodesic distance is a non-negative, symmetric, and satisfies the triangle inequality. Therefore, it is a metric on discrete triangular mesh. Second, shortest geodesic is isometry invariant, and thus does not change over isometric transformation of reflection, translation, and rotation on surface.

In terms of differential geometry, the shortest geodesic solves the boundary value problem of the geodesic equation. The definition is as follows. And see the red curve in Figure 2.3 as an example.

Definition 2.2 *The Boundary Value Problem*

Instance: A triangular mesh M , two points s and t on the mesh.

Question: Find a shortest path in the Euclidean metric from s to t , such that the path stays on the mesh surface M .

2.4.2 The Straightest Geodesic and the Initial Value Problem

The straightest geodesic is defined on polygon mesh to uniquely solve the initial value problem [PS06]. One application is to trace moving particles that are restrict on a polygon surface. See Definition 2.3 in the following. And see the blue curve in Figure 2.3 as an example.

Definition 2.3 *The Initial Value Problem*

Instance: A triangular mesh M , a point p on the mesh and a tangential direction $t_p \in T_p(M)$ at p .

Question: Find the geodesic path $\gamma(p, t_p)$, whose tangent vector γ' is parallel on the mesh surface M .

The straightest geodesic simulates properties of smooth geodesic on surface. The definition by Polthier [PS06] allows discrete geodesic curvature and discrete Gauss-Bonnet theorem. Upon this definition, parallel translation of vectors and integration of vector fields on polygon surfaces is available. However, the straightest geodesic is not a measure on polygon surface.

2.5 Geodesic Coordinates

Based on the existence and uniqueness properties of geodesics on surface S , a useful polar coordinate for S can be constructed. Given a source point $s \in S$, an angle parameter $\theta \in [0, 2\pi]$, and a radius $r \in [0, r_{max}]$, a point $p \in S$ is parameterized by a polar coordinate

(θ_p, r_p) . The polar coordinate is determined by the unique geodesic connecting s and p , where θ_p represents the direction and r_p corresponds to the length of the geodesic.

Chapter 3

Literature Review

As mentioned above, discrete geodesic paths are fundamentally different from geodesics on smooth surfaces, since the shortest geodesics and the straightest geodesics are not equivalent any more. On the one hand, the shortest path problem arises in the field of computational geometry, and it is defined as the shortest path between two given points on a polyhedron; on the other hand, the straightest geodesic problem is defined in differential geometry, and the objective is to approximate properties of smooth geodesic. This chapter briefly reviews the related work on computing shortest and straightest geodesics on discrete domains, constrained texture mapping, and discrete geodesic polar maps.

3.1 Shortest Geodesic Algorithms

Shortest geodesics have been extensively studied and also widely used in many applications. In 1986, Sharir and Schorr [SS86] proposed the first algorithm with complexity of $O(n^3 \log n)$ for resolving “single-source-all-destination” shortest path on convex polyhedral surface.

Generally, the algorithms for computing exact shortest geodesic paths on polyhedral surfaces can be grouped into two categories, namely, the Mitchell-Mount-Papadimitriou (MMP) algorithm [MMP87] and the Chen-Han (CH) algorithm [CH90]. Though the

two types of algorithms are developed based on the continuous Dijkstra’s algorithm that iteratively propagates the discrete wavefront from the source to the destination, they differ in the wavefront organization and propagation scheme. The MMP algorithm has an $O(n^2 \log n)$ time complexity and an $O(n^2)$ space complexity for a mesh with n vertices, while the CH algorithm runs in $O(n^2)$ time and takes $O(n)$ space. However, Surazhsky et al. [SSK⁺05] implemented the MMP algorithm and showed experimentally that the MMP algorithm runs much faster than the CH algorithm. In the past few years, many different extensions have been developed based on the two algorithms, including parallelization [YXH14], improving performance [XWL⁺15, XW09, SSK⁺05, Liu13] and robustness [LZH07, BPS09, CK11], computing geodesic offsets or iso-contours [BK07, XYH11, LCT11a, LCT11b], anisotropic geodesics [LRF10, CHK13], geodesic loops [XHF12], and all-pairs geodesics [XYH12]. In addition, Ying et al. [YWH13] proposed a graph theory based method for meshes that have saddle vertices. As far as we know, the Xin-Wang method i.e. improved CH algorithm [XW09] is the most efficient method so far to compute exact shortest geodesic path and geodesic distance field.

In addition to the exact methods, there are also approximate methods [LMS97, HS98, VA00] for computing shortest geodesics with an error bound. Among them, the Fast March Method (FMM) [Set96] is one of the widely used methods, which solves the boundary value problems of the Eikonal equation with complexity $O(n \log n)$.

3.2 Straightest Geodesic Algorithms

Polthier and Schmies [PS06] studied the straightest geodesic and defined it as a curve that bisects the total vertex angle. Definition 3.1 gives a formal definition of the straightest geodesic.

Definition 3.1 (*Straightest Geodesic*). *A curve γ on a polyhedral surface M is*

called a *straightest geodesic* if for any point $p \in \gamma$ the left and right curve angles θ_l and θ_r at p are equal, that is, the geodesic curvature vanishes at p . See Figure 3.1(a).

A discrete geodesic γ which does not contain mesh vertices, is both straightest and locally shortest. In general, the straightest geodesic is not equivalent to the shortest geodesic on polyhedral surfaces. We summarize the differences between them as follows:

- The shortest geodesic distance is a metric, while the straightest geodesic distance is not, due to the violation of the triangle inequality. See Figure 3.1(c).
- The shortest geodesic path cannot pass through a spherical vertex p with total vertex angle $\theta < 2\pi$, since moving it off p will shorten the length. On the other hand, any path through p that bisects θ is a straightest geodesic. See Figure 3.1(b).
- The shortest geodesics through a hyperbolic vertex with total vertex angle $\theta > 2\pi$ have geodesic curvature κ_g in the interval $[-\pi(1 - \frac{2\pi}{\theta}), \pi(1 - \frac{2\pi}{\theta})]$. Among these shortest geodesic paths, only the one with zero geodesic curvature is the straightest geodesic.
- The shortest geodesics in metric geometry solve the boundary value problem of connecting two given points on a manifold with a locally shortest curve. The straightest geodesics do not solve the boundary value problem, since there exists shadow region in the neighborhood of a hyperbolic vertex, where points cannot be joined by a straightest geodesic. Instead, the straightest geodesics solve the initial value problem with a given position and an initial tangent direction.
- The shortest geodesic distances converge to geodesic distances on smooth surfaces at a quadratic rate [CWW13], whereas the straightest geodesics do not converge [LT09].

Figure 3.1: (a) A straightest geodesic has equal left and right curve angles $\theta_l = \theta_r$. (b) The shortest geodesic between s and t cannot be extended through a spherical vertex p , since making it off p shortens the length. (c) Unfold the spherical vertex p to $\mathbb{R}^2$.

3.2.1 Straightest Geodesic on Parametric Surface

Computing geodesics on parametric surfaces, such as Bézier, B-splines, and NURBS, has also been well studied. The problem is important in geometric design of ship shells, computation of medial axis transforms of trimmed surface patches, and cable installation on the sea floor. Beck et al. [BFH86] computed geodesics on a bicubic spline surface by solving the four first-order ODEs of geodesic (See Equation 3.1). These equations were also used in [RKSDH⁺03] for computing geodesics on NURBS surfaces. Patrikalakis and Bardis [PB89] computed geodesic offsets of curves on rational B-splines using the initial-value integration normal to an initial progenitor curve on the surface. The application of the computed offsets is to construct linkage curves for free-form procedural blending surfaces. Maekawa [Mae96] computed the geodesic path between a point and a curve on a parametric surface [PM09]. They employed a finite difference approximation to discretize the governing differential equations. Chen et al. [CHY⁺13] solved the geodesic equation by adjust parameters between connected Bézier patches. For all these methods, the surface needs to be constructed with precise C^2 parametric continuity, so that the

Christoffel symbols are thus continuous across adjacent Bézier patches.

$$\begin{cases} \frac{d\mathbf{u}}{ds} = u', \\ \frac{d\mathbf{v}}{ds} = v', \\ \frac{d\mathbf{u}'}{ds} = -\tau_{11}^1(u'^2) - \tau_{12}^1 u'v' - \tau_{22}^1(v'^2), \\ \frac{d\mathbf{v}'}{ds} = -\tau_{11}^2(u'^2) - \tau_{12}^2 u'v' - \tau_{22}^2(v'^2), \end{cases} \quad (3.1)$$

3.2.2 Straightest Geodesic on Polygon Mesh

To our knowledge, the straight geodesics have not been well studied compared to the shortest geodesics in computer graphics. Straightest geodesics receive relatively less attention than shortest geodesics in the area of computer graphics.

Polthier and Schmies [PS06] introduced the discrete geodesic curvature and defined the straightest geodesic on polyhedral surface as a path that has equal curve angle on both sides at each point. Then, they proposed the geodesic Euler method and the geodesic Runge-Kutta method for integrating a given vector field on a surface. Polthier and Schmies also developed the geodesic flow method [PS99] to compute the evolution of the front of a point wave on a polyhedral surface. At each time the front was a topological circle on the surface [LCT11a], where each point was moved a constant distance in orthogonal direction to the curve by the straightest geodesic. Kumar et al. [RKSDH⁺03] observed that the straightest geodesic obtained by tracing the path with equal left and right curve angles was far from the correct geodesic curve on the smooth surface. Therefore, they proposed a sectional plane method which took into account the variation of the tessellation normal along its path. Kumar et al.'s method can be considered as an extrinsic Euler method, which solved the geodesic equation with the first-order approximation. Therefore, their method tended to suffer from serious numerical issue and may deviate from the true geodesic curve after only a few iterations. Lee et al. [LHL06] also

presented the extrinsic sectional plane method, which computed the geodesic path by the intersection of the cutting plane and the surface at current point. Based on the fast marching method and the straightest geodesic [PS06], Martinez et al. [MVC05] proposed an iterative algorithm for computing the shortest path between two fixed points. For the visualization of geodesics on surface, Hotz and Hagen [HH00] employed the straightest property of geodesic and computed the approximation of smooth geodesic on parametric surface by projecting current tangent direction onto next tangent plane.

3.2.3 Convergence of Straightest Geodesic

Convergence of discrete differential operators and quantities has also been studied in the past years. Under the assumption of convergence of surfaces in Hausdorff distance, Hildebrandt et al. [HPW06] proved that convergence of the following properties were equivalent: surface normals, surface area, metric tensors, and Laplace Beltrami operators. Dai et al. [DLJ⁺07] derived the explicit formulae to the bounds of Hausdorff distance, normal distance, and Riemannian metric distortion between the smooth surface and the triangle mesh. They proved that the meshes induced from Delaunay triangulations of dense samples on a smooth surface were convergent to the smooth surface under both Hausdorff distance and normal fields. Moreover, the Riemannian metrics and the Laplace-Beltrami operators defined on Delaunay triangulations can also converge to those on smooth surfaces. Hildebrandt et al. [HPW06] and Memoli and Sapiro [MS01] showed that if a sequence of triangulations converge in Hausdorff distance to a smooth surface S , and the normals also converge to the normals of S , then the limit curve of a sequence of shortest paths was a shortest path of S . Lieutier and Thibert [LT09] concluded that the convergence of *shortest* geodesics required additional assumptions. The assumptions were concerning the rate of convergence of the normals and of the lengths of the edges of the triangulations. They pointed out that the edge lengths cannot converge faster to zero than the angles between the normals.

3.3 Constrained Texture Mapping

To our knowledge, *Lévy et al.* [Lév01] first studied constrained texture mapping problem. The key challenge that lies in constrained texture mapping is to avoid foldover and generate visual pleasing texture mapping. Floater and Hormann et al. [FH05] [SPR06, SHL⁺07] studied this problem in a geometric modeling view.

Existing methods for the problem can be grouped into two categories, namely, global optimization based methods [Lév01] [LPRM02] [DMA02] and image warping based methods [ESG01][KSG03][MZX15]. For the global optimization method [LM98, Lév01], its objective function consists of two parts, where the first part deals with the constraints defined by user, while the second part copes with the smoothness of the final mapping. The second part controls the smoothness of the final mapping. There is thus a balance between satisfying constraints and minimizing distortion of the mapping. In addition, hard constraints can be handled by using Lagrange multipliers. However, the optimization-based methods tend to suffer from the following two issues: 1) They do not guarantee a bijective embedding; and 2) They do not guarantee to converge.

As for the image-warping based methods [KSG03][MZX15], Delaunay triangulation and edge swaps are frequently used to meet positional constraints and to avoid fold-overs. Eckstein et al. [ESG01] constructed a 2D mesh in image plane and warped it onto mesh surface. The 2D mesh was constructed with the same topology of the mesh surface. Then mapping was created between corresponding vertices and triangles. One limitation of this method is that it is complicated and not robust, though it may handle a large set of constraints. Kraevoy et al. [KSG03] and Lee et al. [LYY08] performed embedding by adding a fixed rectangular virtual boundary. Then, they applied the Delaunay method to triangulate the region between true and virtual boundaries. A subsequent step of smoothing is required to reduce distortion, after aligning user-specified hard constraints.

3.4 The Discrete Exponential Map

The exponential map is an intrinsic local surface parameterization in differential geometry. One key challenge for computing an approximate exponential mapping exp_v is to trace smooth and accurate radial angles of each vertex. Schmidt et.al. [SGW06] extended the Dijkstra's graph distance algorithm [Dij59] and computed the exponential map via tracing angle difference while unfolding neighbor triangle. Schmidt [Sch13] then applied the discrete exponential map to generate planar parametrization of a painting stroke on surface. However, due to the linear interpolation of rotation angles, the method introduced large distortions. Sun et.al. [SZZ⁺13] presented an extended exponential map, which is defined on the user-sketched curve. They used a post harmonic mapping method to preserve the angle and area properties. Melvær et.al. [MR12] extended the Fast Marching Method on general polygonal meshes and achieved smoother geodesic polar coordinates by using a simple linear angle interpolation. Groot et. al. [GWB⁺14] computed this intrinsic and isotropic coordinate to edit repetitive texture patterns on surface.

Chapter 4

Solving the Initial Value Problem of Discrete Geodesics

4.1 Introduction

Computing geodesic distances and geodesic paths plays an important role in many fields, such as CAD/CAM [SRD08], path planning [DMT91], shape analysis [SRIC01], parameterization [PC05, FHR02], segmentation [Kos03], remeshing [SSG03, PC06] medial axis [Liu15], and many others. Geodesics on smooth surfaces are well understood in classic differential geometry. However, the discrete geodesic problem, i.e., computing geodesic distances and paths on discrete domains such as polygonal meshes, is fundamentally different from its smooth counterpart, due to the difference between smooth and discrete domains. For example, geodesic paths are both straightest and locally shortest on smooth surfaces, but such a nice property does not hold on polygonal meshes. The discrete *shortest* geodesic is not equivalent to the discrete *straightest* geodesic, which bisects the vertex angles, since the former is a metric but the latter is not.

As a fundamental problem in computational geometry and geometric modeling, the discrete geodesic algorithms have been studied extensively in the past three decades. To date, many elegant algorithms have been proposed. Representative works include the *exact*¹ algorithms (e.g., the Mitchell-Mount-Papadimitriou (MMP) algorithm [MMP87]

¹If the numerical computation is exact, the computed geodesic distance is also exact.

Figure 4.1: The discrete *shortest* geodesic does not solve the initial value problem. Let v be a saddle vertex, whose cone angle is more than 2π . When a shortest geodesic path, say $\gamma(s, v)$, passes through v , it splits into many outgoing geodesic paths: any line segment $\overline{pv}$ in the fan-shaped area (in grey) together with $\gamma(s, v)$, is a shortest path from s to p . Therefore, the initial value problem does not have a unique solution, if one considers the *shortest* geodesic paths.

and the Chen-Han (CH) algorithm [CH90]), the PDE methods (e.g., the fast marching method [KS98], and the heat method [CWW13, CHK13]), and the graph-theoretic methods (e.g., the saddle vertex graph [YWH13]). These algorithms, however, are mainly designed for solving the boundary value problem, that is, finding the shortest path between two fixed endpoints.

Mitchell et al. [MMP87] proved that the general form of a *shortest* geodesic path γ is an alternating sequence of vertices and (possibly empty) edge sequences. The unfolded image of the path along any edge sequence is a straight line segment, and the curve angle of any vertex γ passing through is greater than or equal to π . As Figure 4.1 shows, when a shortest geodesic path γ passes through a saddle vertex (whose cone angle is more than 2π), γ splits into multiple outgoing geodesic paths. Therefore, the shortest geodesic paths, although well defined, are not able to solve the initial value problem of discrete geodesics, which aims at finding the *unique* geodesic path from a fixed point and in a given tangent direction.

In this Chapter, we present a method for solving the initial value problem on triangle meshes. To ensure a unique solution, we adopt an indirect strategy. Given a fixed point and an initial tangent direction on a triangle mesh M , we first compute a geodesic curve $\widehat{\gamma}$ on a piecewise smooth surface $\widehat{M}$, which well approximates the input mesh M and can be constructed at little cost. Then, we solve a first-order ODE of the tangent by the fourth-order Runge-Kutta method, and parallel transport it along $\widehat{\gamma}$. When the geodesic curve reaches the boundary of the current patch, its tangent can be transported to the neighboring patch, since the piecewise smooth surface $\widehat{M}$ is constructed with G^1 -continuity along the common boundary of two adjacent patches. Finally, once the geodesic curve $\widehat{\gamma}$ is available, we can project it onto the underlying mesh M , producing the discrete geodesic path, γ , which is guaranteed to be unique on the triangle mesh M (See Figure 4.2).

It is worth noting that our method is different from the conventional methods of directly solving the geodesic equation (i.e., a second-order ODE of the position) on the piecewise smooth surface, which are tedious and difficult to implement due to the complicated representation of the geodesic equation involving Christoffel symbols. Based on the first-order ODE of the tangent vector, the proposed method is intuitive and easy to implement. We believe that our method is particularly useful for computing geodesic paths on low-resolution meshes with large and/or skinny triangles, where the conventional *straightest* geodesic paths are usually far from the ground truth. Also, our method can be easily adapted to non-orientable surfaces.

4.2 Our Algorithm

4.2.1 Overview

Let $M = (V, E, F)$ be a triangle mesh, where V , E , and F are the set of vertices, edges, and faces, respectively. We estimate the vertex normal by using the weighted-average

Figure 4.2: Our method solves the initial value problem of discrete geodesics. For each point (in red) on the Rhino model, we compute geodesic paths in 60 tangent directions, which are evenly sampled on the tangent plane. Each tangent direction is guaranteed to produce a unique geodesic path on the triangle mesh. Our method is numerically stable and works well on meshes with large and/or skinny triangles.

method with 1-ring neighborhood and Voronoi area weights [MDSB03, Rus04]. The tangent plane of vertex v , denoted by T_v , passes through v and is perpendicular to v 's normal $\mathbf{N}(v)$. We assume the input mesh M is sampled from a smooth surface S so that it converges to S when the Hausdorff distance between S and M is sufficiently small and the normal fields on M converges to those on S .

Given an arbitrary point $\mathbf{p} \in V$, and a tangent direction $\vec{w} \in T_{\mathbf{p}}$, our goal is to find a *unique* geodesic path which passes through $\mathbf{p}$ in the direction of $\vec{w}$. As mentioned above, we cannot use the shortest path, which may produce multiple solutions. Since the initial value problem is guaranteed to have a unique solution on C^2 continuous surfaces, we consider a piecewise smooth surface $\widehat{M}$, which approximates M well and can be constructed from M with little cost. In our implementation, we construct for each triangle $f \in F$ a curved PN-triangle [VPBM01] $\widehat{f} \in \widehat{M}$, which is a three-sided cubic Bézier patch corresponding to a triangle $f \in F$ of the underlying mesh M . The curved PN-triangle $\widehat{f}$ interpolates both the position and normal of f 's three corners (See Figure 4.3). By using

rational blending [FMHF08, BFRA12], adjacent Bzier patches can be glued so that they have G^1 continuity along the common edge (See Figure 4.5). Note that other methods for locally construct piecewise smooth surfaces (e.g., [LWZ11]) can also be adopted.

Figure 4.3: A curved PN-triangle is a three-sided cubic Bzier patch that interpolates both the positions and normals of the three vertices of a planar triangle.

Throughout this Chapter, if an object is defined on both the underlying triangle mesh M and the piecewise smooth surface $\widehat{M}$, we distinguish its notations by the hat symbol. For example, point $\widehat{\mathbf{p}}$ is on the curved PNG1-triangle $\widehat{f}$ and point $\mathbf{p}$ is on the corresponding planar triangle f . Note that the tangent vectors are always defined on the curved surface $\widehat{M}$, thus, we simply denote them in the conventional way, i.e., using boldface font.

4.2.2 Algorithm

Let $\widehat{f}$ be the curved patch containing $\mathbf{p}$ and its tangent $\vec{w}$. We compute a unique geodesic path $\widehat{\gamma} \in \widehat{f}$ using the fourth-order Runge-Kutta method and project $\widehat{\gamma}$ onto the underlying triangle face f . Due to the G^1 continuity across the boundary, we can transfer the tangent direction to the neighboring patch, when $\widehat{\gamma}$ crosses the boundary of $\widehat{f}$. This allows us to extend $\widehat{\gamma}$ to its neighboring patch, as shown in Figure 4.5. We continue the procedure until the geodesic curve reaches the boundary or its length exceeds the user-specified threshold.

 (a) Computing $\hat{\mathbf{p}}_{i+1}$

 (b) Projecting $\hat{\mathbf{p}}_{i+1}$

 (c) Computing $\mathbf{T}'_{i+1}$

 (d) Parallel transport $\mathbf{T}'_{i+1}$ to $\hat{\mathbf{p}}_{i+1}$

Figure 4.4: A typical iteration in our algorithm.

To compute the geodesic curve $\widehat{\gamma}$ on $\widehat{M}$ numerically, our algorithm takes a step length δ as input and solves the initial value problem in an iterative manner. Each iteration has the following four steps as illustrated in Figure 4.4:

- (i) The point $\widehat{\mathbf{p}}_i$, the normal $\mathbf{N}_{\widehat{\mathbf{p}}_i}$, and a tangent vector $\mathbf{T}_i$ define a sectional plane π . Find a point $\widehat{\mathbf{p}}_{i+1} \in \pi \cap \widehat{M}$ such that $d(\widehat{\mathbf{p}}_i, \widehat{\mathbf{p}}_{i+1}) = \delta$, as shown in Figure 4.4.a.
- (ii) Let $\widehat{f}$ be the PNG1-triangle containing $\widehat{\mathbf{p}}_{i+1}$ and $f \in F$ the underlying triangle. Project $\widehat{\mathbf{p}}_{i+1}$ onto f along its normal direction and set $\mathbf{p}_{i+1}$ the foot point. If the foot point is outside f , we simply reset $\mathbf{p}_{i+1}$ as the intersection of $\mathbf{p}_i \mathbf{p}_{i+1}$ and the corresponding edge of f (See Figure 4.4.b).
- (iii) Compute the tangent $\mathbf{T}'_{i+1}$ using the fourth-order Runge-Kutta method (See Section 4.2.3 and Figure 4.4.c).
- (iv) Parallel transport the tangent $\mathbf{T}'_{i+1}$ along $\widehat{\gamma}$ to $\widehat{\mathbf{p}}_{i+1}$ (See Section 4.2.4 and Figure 4.4.d).

The output is a sequence of points $\{\mathbf{p}_0(=\mathbf{p}), \mathbf{p}_1, \mathbf{p}_2, \dots\}$, defining the discrete geodesic path through $\mathbf{p}$ in the given direction on the underlying triangle mesh (See Algorithm 1 for the pseudo-code).

4.2.3 Tangent Computation

Let $\gamma \in S$ be an arc-length parameterized geodesic curve on a smooth surface S . Since its geodesic curvature vanishes $\kappa_g = 0$, the curvature vector $\ddot{\gamma}$ is parallel to the surface normal. Therefore, the Frenet-Serret frame of γ coincides with the Darboux frame of

Protocol 1 Solving the Initial Value Problem of Discrete Geodesics

Require: A triangle mesh $M = (V, E, F)$, a point $\mathbf{p} \in V$, a tangent direction $\vec{w} \in T_{\mathbf{p}}$, the step length δ , and the maximal length L

Ensure: The sequence of points $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_i \in M$, representing the discrete geodesic path, whose length does not exceed L

- 1: Construct PNG1-triangle for each face $f \in F$;
 - 2: $\mathbf{p}_0 = \mathbf{p}$; $s_0 = 0$; $\mathbf{T}_0 = \vec{w}$;
 - 3: **for** $i = 0$; $i < \lceil \frac{L}{\delta} \rceil$; $i++$ **do**
 - 4: $\hat{\mathbf{p}}_{i+1} = \text{geodesic_path}(\hat{\mathbf{p}}_i, \mathbf{T}_i, \delta)$;
 - 5: $\mathbf{p}_{i+1} = \text{projection}(\hat{\mathbf{p}}_{i+1})$;
 - 6: $\mathbf{T}'_{i+1} = \text{Runge_Kutta}(\hat{\mathbf{p}}_i, \mathbf{T}_i, s_i, \delta)$;
 - 7: $\mathbf{T}_{i+1} = \text{parallell_transport}(\mathbf{T}'_{i+1}, \hat{\mathbf{p}}_i, \hat{\mathbf{p}}_{i+1})$;
 - 8: $s_{i+1} = s_i + \delta$;
 - 9: **end for**
-

surface S , i.e.,

$$\begin{cases} \frac{d\mathbf{T}}{ds} = \kappa_n \mathbf{N}, \\ \frac{d\mathbf{N}}{ds} = -\kappa_n \mathbf{T} + \tau \mathbf{B}, \\ \frac{d\mathbf{B}}{ds} = \tau \mathbf{N}, \end{cases} \quad (4.1)$$

where τ is the torsion of γ .

Since $\kappa_n = -\frac{d\mathbf{N}}{ds} \bullet \mathbf{T}$, we have the following first-order ODE on tangent $\mathbf{T}$:

$$\frac{d\mathbf{T}}{ds} = -\left(\frac{d\mathbf{N}}{ds} \bullet \mathbf{T}\right)\mathbf{N} \quad (4.2)$$

Given the initial value $s_0 = 0$ and $\mathbf{T}_0 = \vec{w}$, we solve the above ODE by the fourth order Runge-Kutta method, where the total accumulated error is order $O(\delta^4)$. Define function $f(s, \mathbf{T}) = -\left(\frac{d\mathbf{N}}{ds} \bullet \mathbf{T}\right)\mathbf{N}$. We compute $\mathbf{T}_i$, for $i = 1, 2, \dots$, as follows:

$$\mathbf{T}'_{i+1} = \mathbf{T}_i + \frac{\delta}{6}(\vec{v}_1 + 2\vec{v}_2 + 2\vec{v}_3 + \vec{v}_4), \quad (4.3)$$

$$s_{i+1} = s_i + \delta, \quad (4.4)$$

where

$$\vec{v}_1 = f(s_i, \mathbf{T}_i), \quad (4.5)$$

$$\vec{v}_2 = f\left(s_i + \frac{\delta}{2}, \mathbf{T}_i + \frac{\delta}{2}\vec{v}_1\right), \quad (4.6)$$

$$\vec{v}_3 = f\left(s_i + \frac{\delta}{2}, \mathbf{T}_i + \frac{\delta}{2}\vec{v}_2\right), \quad (4.7)$$

$$\vec{v}_4 = f(s_i + \delta, \mathbf{T}_i + \delta\vec{v}_3). \quad (4.8)$$

We iteratively update the tangent direction using the above equations and trace the geodesic.

4.2.4 Parallel Transport

Note that the computed tangent $\mathbf{T}'_{i+1}$ is located at $\hat{\mathbf{p}}_i$. So we need to parallel transport $\mathbf{T}'_{i+1}$ to $\hat{\mathbf{p}}_{i+1}$ along the geodesic curve $\gamma(\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_{i+1})$.

Let α be the angle between $\mathbf{T}'_{i+1}$ and $\mathbf{T}_i$ at $\hat{\mathbf{p}}_i$. Projecting $\overrightarrow{\hat{\mathbf{p}}_i \hat{\mathbf{p}}_{i+1}}$ onto the tangent plane $T_{\hat{\mathbf{p}}_{i+1}}$, we obtain a reference tangent direction $\mathbf{T}$. With the normal $\mathbf{N}$ at $\hat{\mathbf{p}}_{i+1}$, we compute the binormal $\mathbf{B} = \mathbf{T} \times \mathbf{N}$. Finally, we compute $\mathbf{T}_{i+1} = \mathbf{T} + \cos \alpha \mathbf{B}$. Figure 4.4.d illustrates the parallel transport of tangent vectors.

4.3 Experimental Results

4.3.1 Runtime Performance

We implemented our algorithm in C++ and tested it on an Intel Core2 2.83GHz CPU. To make the results *unitless*, we uniformly scale all models to a unit cube. We measured the running time of computing 60 geodesic paths evenly emanating from 100 randomly selected source points and having the length $\sqrt{3}$. Theoretically, the time complexity of solving the ODE (lines 3-9 in Algorithm 1) depends only on the maximal length L and the step length δ . In practice, we observe that the mesh complexity affects the running time slightly since the point location is triangulation dependent (See Figure 4.6).

Figure 4.5: A curved PNG1-triangle is G^1 continuous with its three neighboring patches. The patch is obtained by blending two cubic Bézier patches for each triangle edge (see (a)). For patches that meet at a vertex v , control points corresponding to the vertex are all projected to the tangent plane at v (see (b)). Thus, the tangent direction remains unchanged when the geodesic crosses the edge boundary or passes through a vertex. As a result, we can continue to solve the first-order ODE of the tangent on the neighboring patch (see (c)).

Figure 4.6: The time complexity is inversely proportional to the step length δ . The vertical axis shows the running time of computing 60 geodesic paths of length $\sqrt{3}$ at 100 random source points. The time for constructing the PNG1-triangles, which is dependent on mesh complexity, is excluded.

4.3.2 Robustness

To further evaluate the robustness of our method, we computed all-direction initial value geodesic paths on several common 3D models in graphics community, as shown in Figure 4.7. We observe that most geodesic paths on real-world models do not return to the initial point, thus, they can be arbitrarily long. Computational results show that our method is numerically stable and works well on meshes with large and skinny triangles as well as sharp features. For example, geodesics cross sharp edges on the fandisk model and the saddle vertices on the *bunny ear* behave smoothly (See Figures 4.2 and 4.8). According to the aforementioned evaluations on convergence on the quadric surfaces (Section 4.3.3) and robustness on real-world models, we conclude that our method is numerically stable and insensitive to mesh triangulation and resolution.

Figure 4.11 demonstrates our algorithm on computing the local exponential map on triangle meshes, which naturally define polar coordinates on 3D surfaces. Figure 4.9 shows the results of the initial value problem with user-specified starting point, tangent direction and length. Our method can be also applied to non-orientable surfaces (See

Figure 4.10). Note that the existing methods, such as the MMP algorithm [MMP87] and the CH algorithm [CH90], do not have such a feature, since they adopt the half-edge data structure to maintain the windows (a key data structure that encodes the geodesic distance to the source) on oriented edges.

Figure 4.7: We solve the initial value problem on a wide range of 3D models and we assign each source point 60 evenly distributed tangent directions. Note that our method works well on highly curved regions as well as sharp features. The images are generated at high resolution to allow for zoom-in examination.

4.3.3 Convergence Analysis

To evaluate the convergence of our method, we constructed four quadric mesh surfaces, namely, sphere, ellipsoid, paraboloid, and hyperboloid and torus surface, which converge to their smooth counterparts (See Figure 4.12). For each surface, we have seven mesh surfaces with increasing sample rate. All mesh vertices are on the smooth surfaces and

Figure 4.8: Solving the initial value problem on the Rhino model with two different length constraints. (see (a) and (b)). Wireframe sculpture made with initial value geodesic path with different length. (see (c) and (d)). Note that the source point is located at the belly button and different sculpture with different source point or initial tangent direction.

Figure 4.9: Given the user-specified point (green), tangent direction and length, our algorithm computes the unique solution of the initial value problem.

Figure 4.10: Our method also works for the non-orientable surfaces.

Figure 4.11: Computing local parameterization using our method. The user specifies a point $\mathbf{p}$ on the 3D mesh and a radius r . We then apply our method to compute 60 geodesic paths passing through $\mathbf{p}$ and having length r . Each geodesic path uniquely corresponds to a radial line of a polar coordinate system defined on $T_{\mathbf{p}}$, allowing us to build a polar coordinate system on 3D meshes. We visualize the parameterization by checkerboard texture mapping on a rectangular patch. The bottom row shows the parameterized patch on $\mathbb{R}^2$.

we compute the vertex normals analytically. Table 4.1 shows the statistics of each model with number of vertices and faces. Figure 4.12 shows all-direction geodesics from a vertex and the corresponding geodesic circles on the quadric surfaces with the lowest sample rate.

Our method computes the discrete geodesic path γ by projecting the unique geodesic $\hat{\gamma}$, which is defined on the piecewise smooth surface $\widehat{M}$, onto the underlying piecewise linear surface M . Thus, it is natural to consider the “gap” between γ and $\hat{\gamma}$ and the length difference $|\text{len}(\gamma) - \text{len}(\hat{\gamma})|$ as the convergence measures. For each surface, we solve the initial value problem at 10 points and we assign 60 evenly distributed tangent directions for each point. Let h be the average edge length of each mesh surface, the average length difference is $\Delta = L|\text{len}(\gamma) - \text{len}(\hat{\gamma})| = \sum_{i=1}^{10} \sum_{j=1}^{60} \frac{L(\tilde{\gamma}_{ij}) - L(\gamma_{ij})}{600}$, and the average gap area is $\text{Area} = \sum_{i=1}^{10} \sum_{j=1}^{60} \sum_k \frac{\max(h_{ij}^k) L(\gamma_{ij}^k)}{600}$, where $L(\gamma_{ij}^k)$ is the length of geodesic within one triangle and $\max(h_{ij}^k)$ is the maximum distance between $\tilde{\gamma}$ and γ within a triangle. Figure 4.12(a) shows that when the mesh resolution is sufficiently high, the curved PNG1-triangles converge to the smooth surfaces. As for the length difference ΔL and gap area between γ and $\hat{\gamma}$, Figure 4.12(b)(c) shows that both the gap area and length difference converge to zero as the average edge length h decreases. Since γ is on the triangle mesh, and $\hat{\gamma}$ is on the PNG1-triangle surface, the convergence here is with respect to ground truth on PNG1-triangle surface.

4.4 Discussion

4.4.1 First-order ODE vs Second-order ODE

Computing geodesics on parametric surfaces, such as B-spline surfaces and NURBSs, has been studied in the literature [PB89][RKSDH⁺03][CHY⁺13][Mae96]. These methods directly solve the standard geodesic equation (2.4), which is a second-order ODE of the

Figure 4.12: Convergence test on four synthetic models. Sphere $(\cos u \sin v, \sin u \sin v, \cos v)$, $u \in [0, 2\pi]$, $v \in [0, \pi]$; Ellipsoid $(2 \cos u \sin v, 2 \sin u \sin v, 3 \cos v)$, $u \in [0, 2\pi]$, $v \in [0, \pi]$; Paraboloid $(2v^2, 4v, 6u)$, $u, v \in [-1, 1]$; Hyperboloid $(2 \cos v \sqrt{1 + u^2}, 2 \sin v \sqrt{1 + u^2}, 4u)$, $u \in [-1, 1]$, $v \in [0, 2\pi]$; Torus $((3 + \cos v) \cos u, (3 + \cos v) \sin u, \sin v)$, $u \in [0, 2\pi]$, $v \in [0, \pi]$. When the mesh resolution is sufficiently high, the Hausdorff distance between the PNG1-triangles $\widehat{M}$ and the smooth surface S approaches zero, meaning that the curved PNG1-triangles converge to the smooth surfaces S . We also observe that the discrete geodesic $\gamma \in M$ converges to the geodesic $\widehat{\gamma} \in \widehat{M}$, since both the length difference $|\text{len}(\gamma) - \text{len}(\widehat{\gamma})|$ and the area of the gap between γ and $\widehat{\gamma}$ approach zero. The horizontal axis h is the average edge length of M .

Table 4.1: The statistics of surfaces that converge to its smooth counterpart.

Surface	Sphere	Ellipsoid	Paraboloid	Hyperboloid	Torus
(V_1, F_1)	(642, 1,280)	(382, 760)	(256, 450)	(465, 900)	(300, 600)
(V_2, F_2)	(4,482, 5,120)	(1,562, 3,120)	(961, 1,800)	(1,830, 3,600)	(1,200, 2,400)
(V_3, F_3)	(19,842, 20,480)	(6,322, 12,640)	(3,721, 7,200)	(7,260, 14,400)	(4,800, 9,600)
(V_4, F_4)	(81,282, 81,920)	(25,442, 50,880)	(14,641, 28,800)	(28,920, 57,600)	(19,200, 38,400)
(V_5, F_5)	(327,042, 327,680)	(102,082, 204,160)	(58,081, 115,200)	(115,440, 230,400)	(76,800, 153,600)
(V_6, F_6)	(1,310,082, 1,310,720)	(408,962, 817,920)	(231,361, 460,800)	(461,280, 921,600)	(307,200, 614,400)
(V_7, F_7)	(5,242,242, 5,242,280)	(1,637,122, 3,274,240)	(923,521, 1,843,200)	(1,844,160, 3,686,400)	(1,228,800, 2,457,600)

position. One, of course, can borrow their approaches to the PNG1-triangles, which are cubic Bézier patches. However, due to complicated representation of the Christoffel symbols on triangular patches, the implementation would be tedious. Moreover, surfaces must be at least C^2 continuous to ensure that the Christoffel symbols are continuous. Our method solves the first-order ODE of the tangent vector on G^1 continuity patches, which is intuitive and easy to implement. Moreover, as Eqn. (4.2) shows, the derivative of tangent depends on the change of normal, which can be computed efficiently on cubic Bézier patches.

4.4.2 Comparison with Straightest Geodesic

To further evaluate the accuracy of our method, we compared our discrete initial value geodesic paths with different computation step length δ and the straightest geodesic [PS06] on the torus model, where the analytic geodesic is available. Although the straightest geodesic converges to the smooth counterpart when the mesh surface resolution is sufficiently high, on a low-resolution torus mesh, it deviates from the analytic geodesic fast as the area is doubly curved with big different curvature size (See Figure 4.13).

It is worth noting that our method is fundamentally different from the straightest geodesic: our geodesic path is a *polyline* inside a triangle, whereas a straightest geodesic path, when restricting to a triangle, must be *straight line*. We observe that straightest

geodesics are too “rigid” to approximate analytic geodesics on low-resolution meshes. As Figure 4.14 shows, the straightest geodesic is far from the ground truth on a low-resolution torus model, which has only 300 faces. Our method, in sharp contrast, is more flexible, providing a high quality approximation when the step length is sufficiently small. We also observe that our results and the straightest geodesics are similar when the input mesh has a sufficiently high resolution and fairly good triangulation (See Figure 4.13 (d) and (e)). This is not a surprise, since the vertex normals for each such triangle do not vary too much. Thus, our method is particularly useful for computing geodesic paths for low-resolution meshes which may contain many large and/or skinny triangles.

Figure 4.13: The straightest geodesic [PS06] (red) which is highly dependent on the mesh triangulation and resolution, converges to the ground truth (blue) only when the mesh resolution is sufficiently high. Our method with step length $\delta = \frac{h}{16}$ produces very good result (green) even when the mesh resolution is very low. Note that all geodesics are given the same initial condition.

Figure 4.14: Comparison of our method (green) to straightest geodesic (red) and analytic geodesic (blue) on the torus model with 600 faces. Note that all geodesics are with the same initial condition. We set the step length $\delta = \frac{h}{2}, \frac{h}{4}, \frac{h}{8}$ and $\frac{h}{16}$, respectively, where h is the average edge length. We can clearly see that our result converges to the ground truth when the step length δ approaches zero. We evaluate the quality of a geodesic curve by measuring the area between it and the ground truth. As shown in (e), the gap is proportional to the length of the geodesic (the horizontal axis).

4.4.3 Adaptive Step Length

Both the performance and the accuracy of our algorithm depend on the step length, i.e., the smaller the step length, the better accuracy we obtain, at the price of higher computational cost. The results reported above are based on fixed step length. In practice, adaptive step length may improve the performance without compromising the quality. Note that the normal variance within an iteration is small given a short step length δ , hereby the adjustment of tangent direction is not significant. As suggested by the adaptive length in the Runge-Kutta method [Pre07], we adopt the following heuristic to determine the step length in our implementation. If the length of $d\mathbf{N}$ is less than a user-specified tolerance tol , we simply compute the locally straightest geodesic for this iteration. Otherwise, we adjust the tangent direction with an adaptive step length δ_a

$$\delta_a = 0.9 \times \delta \times \frac{tol}{\|d\mathbf{N}\|}. \quad (4.9)$$

Empirically, we set the local error tolerance $tol = \frac{h}{2\phi}$. We adopt absolute error estimate so that the difference of tangent direction is bounded. Table 4.2 shows the performance and quality measurements of our algorithm with fixed step length and adaptive step length.

4.5 Conclusion

This Chapter presents a method to solve the initial value problem of discrete geodesics. Since the commonly used shortest geodesic [MMP87] does not produce a unique solution, we solve the problem by computing a geodesic curve on a piecewise smooth surface $\widehat{M}$, which well approximates M and can be constructed at little cost. Each patch in $\widehat{M}$ is a three-sided cubic Bézier spline, thus, it has a unique geodesic curve through the given point in any direction. We solve a first-order ODE of the tangent by the Runge-Kutta method and adopt parallel transport to move the tangent along the geodesic curve. When

Table 4.2: Fixed step length vs adaptive step length. We compute 60 geodesic paths evenly emanating from 100 randomly selected source points and having the length $\sqrt{3}$. The running time t is measured in seconds. δ is the step length, h is the average edge length and ΔL measures the length difference of the geodesic on the curved PNG1-triangles and the discrete geodesic on the piecewise flat surfaces.

Surface	(V , F)	h	Fixed Step Length ($\Delta L, t$)			
			$\delta = \frac{h}{2}$	$\delta = \frac{h}{4}$	$\delta = \frac{h}{8}$	$\delta = \frac{h}{16}$
Sphere	(642, 1280)	0.150730	(0.023123, 0.190147)	(0.011562, 0.327172)	(0.009730, 0.636252)	(0.001264, 1.299852)
Ellipsoid	(382, 760)	0.467991	(0.040078, 0.161148)	(0.019068, 0.260773)	(0.018068, 0.485002)	(0.004623, 1.005054)
Hyperboloid	(465, 900)	0.132355	(0.031381, 0.229931)	(0.015646, 0.258773)	(0.007801, 0.460958)	(0.003913, 0.958326)
Torus	(300, 600)	0.182588	(0.057618, 0.139749)	(0.030948, 0.206672)	(0.012446, 0.425581)	(0.005751, 0.888576)
Surface	(V , F)	h	Adaptive Step Length ($\Delta L, t$)			
Sphere	(642, 1280)	0.150730	(0.002584, 0.277009)			
Ellipsoid	(382, 760)	0.467991	(0.004917, 0.244246)			
Hyperboloid	(465, 900)	0.132355	(0.005089, 0.248684)			
Torus	(300, 600)	0.182588	(0.006547, 0.213495)			

the geodesic curve reaches the boundary of the current patch, its tangent can be transported to the neighboring patch, due to the G^1 -continuity along the common boundary of two adjacent patches. We project the geodesic curve from $\widehat{M}$ onto the underlying mesh M to obtain the unique solution of the initial value problem. We define the error metric as the gap between the two geodesic curves and observe that the computed discrete geodesic converges linearly to its smooth counterpart with increasing mesh resolution. Experimental results show that our method is particularly useful for computing geodesic paths for low-resolution meshes which may contain many large and/or skinny triangles.

Chapter 5

Constrained Texture Mapping via Voronoi Diagram Base Domain

5.1 Introduction

Texture mapping is a common technique to enhance visual realism in computer graphics. Generally, the concept of parametrization which is a one-to-one mapping from 2D image domain to surface, provides a mathematical formulation for studying this problem. In order to build meaningful correspondence between surface geometry features and texture details, *Lévy et al.* [Lév01, ESG01, KSG03, ZWT⁺05] formulated this problem as a constrained optimization problem, where the constraint conditions are the correspondence between mesh feature points and 2D image points. However, one major limitation of their methods is that many constraints are required for only one meaningful feature correspondence. Moreover, users need to manually edit the complex constraints to reduce texture distortion and to avoid foldovers. This is often time-consuming and difficult to implement.

The two challenges of texture mapping or mesh parametrization problem is to avoid foldovers and to minimize deformation. In this chapter, we aim to address the above challenges in the constrained texture mapping problem. We propose a Voronoi diagram base domain method, where for each natural semantic feature (e.g., nose), we only need

one point or curve correspondent constraint to achieve satisfied texture mapping result. Firstly, we construct a Voronoi diagram with all the constraints in 2D image as Voronoi seeds. Then, we construct the other Voronoi diagram by considering all the constraints on mesh surface as Voronoi seeds. As the constraints from 2D image to mesh surface are one-to-one mapped, the Voronoi cells are then one-to-one mapped between the two Voronoi diagrams. Finally, for each pair of the Voronoi cells, we can build the texture mapping by a local bilinear map or exponential map.

Due to the property of “closeness” of Voronoi cells, our method results can avoid global foldover among the Voronoi cells, and the local exponential map is able to minimize texture distortion near the feature seeds. Experimental results show that our method can effectively generate satisfying texture mapping results of human faces with only four natural constraints of “eyes”, “nose” and “mouth”. For example, to map a tiger face onto human face, Matchmaker [KSG03] had to specify 27 point constraints and to pay attention to post-processing for smoothing or removing redundant Steiner vertices. The proposed method, however, is able to produce good results with only four point or curve constraints.

We summarize the advantages of our proposed method as follows.

- Our method employs very few constraints for each of the natural semantic features (e.g., a point or curve constraint for the feature “nose”), thus, the time used for specifying and editing feature constraints can be largely reduced.
- Our method is efficient for computing the texture mapping, as there is no global optimization required for the computation.
- Our method does not introduce any intermediate surfaces or controlling nets used for tuning the mapping or feature constraints.

- The feature constraints for our method can be generalized to curves, because the Voronoi seeds can be curves.

Figure 5.1: Texture mapping results on mask model(499,955) of our proposed method with only four constraints of "eyes", "nose" and "mouse". All the textures are from Beijing opera faces. (a,e) A military general and regent of the state of Shu Han; (b,f) A warlord in the late Sui dynasty; (c,g) One of the four ugly in ancient China; (d,h) A military general between the late Eastern Han dynasty and early Three Kingdoms.

5.2 Proposed Method

We aim to resolve the texture mapping problem. In this section, we introduce a novel method based on global Voronoi diagram base domain and local exponential map. The proposed method uses few natural constraints while maintain a satisfied mapping result.

Constraints are exactly satisfied while preserving as much as possible the metrics of the original 3D geometry. Our proposed method can generate the satisfying feature constraints and meanwhile preserve the metrics of original 3D geometry as much as possible. Our method can thus achieve pretty good mapping results via employing very few natural feature constraints.

5.2.1 Definitions

A triangular mesh surface M is represented as the pair (T, X) , where T represents the topology or connectivity information, and $X = \{x_1, x_2, \dots, x_N\}$, where $x_i \in R^3$ is the geometric position of surface points. The proposed method can deal with surfaces that have complicated topology or even contain holes. The input texture image I is represented as a 2D planar space of $I(u, v)$, $u, v \in [0, 1]$. We suppose the target texture part has been segmented from the the input texture image. In our implementation, the boundary of the target texture is interactively sketched by the user.

Although we categorize constraints into two classes, i.e. points and curves, they are equally treated as Voronoi sites in our implementation. Therefore, we do not differentiate between them in our symbol system. Let $C_M = \{C_M^1, C_M^2, \dots, C_M^m\}$ be the set of point and curve constraints on surface M , and $C_I = \{C_I^1, C_I^2, \dots, C_I^m\}$ be the set of constraints in the texture space I . We construct two Voronoi diagrams V_M and V_I in the two spaces individually. The symbol V_M indicates the Voronoi diagram on mesh surface with C_M as the Voronoi sites, and V_I is the one on 2D image plane with C_I as Voronoi sites. Finally, we compute an exponential map exp_i for each pair of Voronoi cell V_M^i and V_I^i .

5.2.2 Overview

Our approach takes a triangular mesh surface M and target texture image I as input. We suppose mesh surface has natural boundary and texture has already been segmented.

In our implementation, the boundaries for the mesh surface and texture image will be sketched by the user. Then the user may interactively specify the constraints on both image and surface by sketching curves or clicking points. Our approach computes a one-to-one mapping from M to I , which means for each vertex x_i in M , there is a $I(u_i, v_i)$ in I , and vice versa. Our method works in the following two steps (Figure 5.2 provides an illustration for this):

- **Building Voronoi Diagram Base Domain.** Firstly, we build a 2D Voronoi diagram V_I in the image space. The texture boundary is either segmented from the input or interactively sketched by the users. The Voronoi sites are the feature constraints C_I with both points and curves. Secondly, we build the Voronoi diagram V_M on mesh surface separately. The details will be described in Section 5.2.3. With the two Voronoi diagrams V_I and V_M and corresponding feature constraints, we can construct the correspondence between Voronoi cells in V_I and V_M , where Voronoi cells V_M^i on mesh surface M and Voronoi cells V_I^i in texture image I are one-to-one mapped.
- **Local Bilinear Map.** For the local map from Voronoi cell V_I^i to V_M^i , we assume the area near the Voronoi sites (features) is more important, while that relatively far away from the Voronoi sites is less important. We employed the exponential map, or geodesic polar coordinates, to deal with this local map. Because the distortion of exponential map becomes much smaller when a point is close to the source point (Voronoi sites in our implementation). The setting makes our method effectively avoid the distortion, and thus leads to a good mapping around the feature (See Section 5.2.4 for the implementation details).

Figure 5.2: The workflow of the proposed Voronoi Diagram base domain method for constrained texture mapping. (a) The input mesh surface; (b) The mesh surface boundary (red curve), point constraints (green spheres) and corresponding Voronoi Diagram in the texture image; (c) The texture image plane boundary (red curve), point constraints (green spheres), and corresponding Voronoi Diagram on mesh surface; (d) The parametrization result; (e), (f) Different views of texture mapping results.

5.2.3 Building Voronoi Diagram Base Domain on Surface

To compute bisectional curve between Voronoi sites, we measure the surface distance on mesh by computing geodesics from Voronoi sites to all the other vertices. It is generally expensive to compute the geodesic distance [SSK⁺05, MMP87]. To generate a good segmentation of the mesh surface, we expect a smooth distance field, which is not necessarily exact. We extend existing algorithm for computing geodesic distance on mesh surface and propose an efficient and effective method. The proposed method floods the distance and angle field of site points to all the other points with only one traverse for individual vertices. Moreover, geodesic distance filed flooding converges fast when there are more source points (feature constraints). In our implementation, we generate the Voronoi diagram and exponential map at the same time. Algorithm 2 summarizes the steps of our method, where tag_i is a flag, which indicates the Voronoi cell V_M^i that the vertex x_i lies in (See Figure 5.3 for an example), d_i is the geodesic distance from v_i to C_M^i (See Figure 5.5), and α_i is the exponential angle of the corresponding geodesic curve.

Figure 5.3: Tags of each vertex indicated with different colors. (a),(b) The tags on a mask model (499,955) in front and side views; (c),(d) The tags on a refined mask model (31762,62467). Red and blue colors correspond to the Voronoi cells of both left and right eyes, yellow color indicates nose, and the green color corresponds to mouth.

In Algorithm 2, we follow [MR12] to update the geodesic distance and angle. In particular, the function $Distance(i)$ updates the geodesic distance from vertex v_i to a

Protocol 2 Voronoi Diagram Base Domain on Surface.

Require: A mesh surface $M(T, X)$ and feature constraints C_M on surface.

Ensure: For each vertex $x_i \in X$, we compute a tag C_m^i indicating which Voronoi cell it belongs to, a distance d_i from x_i to C_m^i , an angle α_i denotes the corresponding geodesic direction.

for $i = 0; i < \|X\|; i++$ **do**

$d_i = \infty$;

$\alpha_i = 0$;

end for

for $i = 0; i < \|C_M\|; i++$ **do**

$d_{C_M^i} = 0$;

$tag_{C_M^i} = i$;

$Heap.push(C_M^i)$;

end for

while $Heap.notEmpty()$ **do**

$s = Heap.getSmallest()$;

for each $i \in Neighbor(s)$ **do**

$New_{d_i} = Distance(i)$;

if $New_{d_i} < d_i$ **then**

$heap.push(i)$;

$d_i = New_{d_i}$;

$tag_i = tag_s$;

$\alpha_i = Angle(i)$;

end if

end for

end while

seed vertex C_M^i , and the function $Angle(i)$ updates the angle of geodesic from C_M^i to v_i . About the details of the two updates, we refer to [MR12]. Once we obtain the Voronoi cell tag tag_i and the geodesic distance d_i to its related seed vertex for each vertex v_i , the Voronoi points of V_M can be computed by matching the geodesic distance. Then the Voronoi edges can be traced from the start Voronoi point to the end one (in the counterclockwise order of the corresponding Voronoi cell).

5.2.4 Local Bilinear Mapping

Our goal is to map a 2D image area V_I^i to a surface area V_M^i , so that the constraint point C_I^i in the image aligns with constraint point C_M^i on mesh. The exponential map and geodesic polar coordinates [DC76] have been used to define a 2D coordinate system on the surface around C_M^i . The exponential map projects a point v on M to the tangent plane T_v at v . For any unit vector $v \in T_p$, there exists one and only one geodesic $g(v)$ with $g_v(0) = p$ and $g'_v(0) = v$ is exist and unique (See Figure 5.4 for the illustration of this concept).

Figure 5.4: The exponential map at a point v (red sphere) projects geodesic curves (blue lines) originating at v in (a) to straight vectors, which emanate from the origin of the tangent plane T_p . (b) Iso-contours (green lines) of the geodesic distance.

One key challenge for computing an approximate exponential mapping exp_v is to trace smooth and accurate radial angles for each vertex. Schmidt et.al. [SGW06] extended

Dijkstra’s graph distance algorithm [Dij59] and traced the angle difference with the rotation angle while unfolding the neighbor triangle. Melvr et.al. [MR12] produced more smooth geodesic polar coordinates via using a simple linear angle interpolation. As a smooth mapping is preferred, we follow [MR12] and employ the linear angle interpolation method in our implementation. To match the Voronoi cell V_I^i and V_M^i , we perform a piecewise linear scale of angle in the exponential map (See Figure 5.6 and Equation 5.1 for the scale parameters).

Figure 5.5: Geodesic distance field on refined mask model (31762,62467).

$$\frac{\alpha_i}{\sum_{i=1}^5 \alpha_i} = \frac{\beta_i}{\sum_{i=1}^5 \beta_i} \quad (5.1)$$

5.2.5 Discussion

Although the mapping from Voronoi cells of V_M to that of V_I is one to one, the shape of the mapped Voronoi cell pair could be different due to the curvedness of surface. Sheffer et al.[SPR06] studied the alternative complexes based domains for dealing with the cross-parametrization problem from surface to surface.

Kraevoy et.al. [KSG03] addressed the constrained texture mapping problem by matching the Delauney triangulation of constrains to surface planar parametrization with vir-

Figure 5.6: Piecewise scale of angle for each pair of the Voronoi diagrams.

tual boundary. The matching is time-consuming, and Steiner points have to be computed to avoid blocking of the match. However, the method presented in [KSG03] cannot completely filter out the foldovers, because the consistent neighboring ordering was not considered when building matched triangulations. In order to bound the distortion of cross parametrization, Kwok et.al. [KZW12] constructed base domain as triangulation induced from Voronoi diagram on surface.

As using only few feature constraints is able to achieve satisfying texture mapping, the shape of both Voronoi diagrams of our method is much simpler. Then, we can simplify the problem largely. Specifically, for constrained texture mapping of human face, we can adjust constraints on both the image and mesh surface to match the face anthropometric measurements, which reflect the facial physical structure, and to make them consistent [ZB06].

5.3 Experimental Results and Discussion

In this section, we apply the proposed method to several examples for constrained texture mapping, and show its validity, efficiency, and robustness.

First of all, we evaluated our method with texture images of Beijing Drama faces and a coarse mesh surface of human face mask. As shown in Figure 5.7, all the results are generated with four constraints of “eyes”, “nose”, and “mouth”. In Figure 5.8, we applied the checkerboard texture to the proposed method and show the smoothness of the parametrization. Although exponential map is neither conformal nor area-preserving, the mapping result is smooth, as shown in (c), (d), and (e) of Figure 5.8. Moreover, we find that the exponential map did not result in foldover, even though it is not guaranteed to be foldover-free on a large neighborhood.

Then we evaluated the proposed method with curve constraints. Curves are another natural way to indicate constraints. To loose the curve constraints, we will not utilize the exact curve as hard constraints. Figure 5.9 shows the results using four curve constraints of “eyes”, “nose” and “mouth”. In this experiment, we only sampled the midpoint and the two endpoints for each curve constraints as point constraints.

Finally, we compare the proposed method with constrained harmonic mapping. Although constrained harmonic mapping has a more smooth parametrization, the texture mapping result expressed big distortion. As shown in Figure 5.10, the constraint features corresponding to eyes and mouth are clearly distorted. Moreover, for a harmonic mapping, the constraints tends to result in foldovers due to the minimization of Dirichlet energy. Users thus have to carefully maintain the structure consistence among all the constraints.

Figure 5.7: Texture mapping results via our proposed method on Beijing Opera faces. (a) Example of the texture image and constraints (green spheres); (b) The mask mesh surface (499,955) and constraints (green spheres); (c), (e), (f), (g), (h), (i) Front views of the texture mapping results; (d), (j), (k), (l), (m), (n) Side views of the texture mapping results.

Figure 5.8: Results of proposed method with checkerboard texture. (a) Input checkerboard image, constraints (green sphere), and boundary (red curve); (b) Mesh surface, constraints (green sphere), and boundary (red curve); (c) Parametrization result; (d), (e) Different views of texture mapping results.

Figure 5.9: Results of proposed method with curve constraints. (a) Texture image with curve constraints (green lines); (b) Mesh surface with curve constraints (green lines); (c) Parametrization result; (d) Texture mapping result.

Figure 5.10: Comparison with constrained harmonic map. (a) Texture image and constraints; (b) Mask surface and constraints; (c) Texture mapping results with constrained harmonic mapping; (d) Texture mapping result with proposed method; (e) Parametrization result of constrained harmonic mapping; (f) Parametrization result of proposed method.

5.4 Conclusion

In this chapter, we proposed a new and effective Voronoi diagram base area method for matching positional constraints to texture maps. Our method can provide a valid one-to-one mapping not only satisfies user-defined constraints but also preserves the metric structure of the mesh surface. The experimental results have demonstrated that our proposed method is able to successfully map a face with only four point or curve constraints. The surface Voronoi diagram and exponential map which is at the heart of the algorithm do not rely on the planarity of the embedding. Therefore, the proposed method can work robustly for parameterizations on the sphere and other domains. Applying constraints to morph texture image on surface [ZB06] would be an interesting problem for future research. In addition, another promising direction is to detect and match features automatically [GKMT01].

Although our local exponential mapping is neither conformal nor area-preserving, the texture mapping result is better than constrained harmonic mapping, as shown in figure 5.10. We thus conclude that better parametrization will not essentially make better texture mapping.

Chapter 6

Conclusions and Future Directions

6.1 Conclusions

In this dissertation, we focus on the initial value problem of discrete geodesics and its applications. We made contributions to the discrete geodesics problem in both theory and application. For the discrete geodesic theory, we proposed the first-order tangent ODE method for uniquely solving the initial value problem, and experimental results have demonstrated that our method works well, even on the coarse mesh. As for the application of the initial value problem geodesics, our proposed method achieves pretty good texture mapping results with very few positional constraints, due to the local discrete exponential mapping on Voronoi diagram base domain.

In particular, in order to cope with the initial value problem of discrete geodesics on polygonal meshes, we proposed a first-order ODE of tangent direction along the geodesic path to avoid constructing C^2 parametric continuity surface and computing Christoffel symbols. Instead, we construct G^1 continuity piecewise smooth surface which well approximates the mesh surface and can be accelerated on graphics hardware. As we construct each patch to a three-sided cubic Bézier spline, the geodesic curve through the given point in any direction is unique. Therefore, we solve the first-order ODE of the tangent by the Runge-Kutta method and adopt parallel transport to move the tangent along the geodesic curve. As we have blended the patch along the common boundary with

G^1 continuity, the tangent direction vector can be directly transported to the neighboring patch. Finally, we obtain the unique solution of the initial value problem by projecting the geodesic curve from piecewise smooth surface onto the underlying mesh surface. Experimental results show that the computed discrete geodesic converges linearly to its smooth counterpart with increasing mesh resolution. Our method is particularly useful for computing geodesic paths for low-resolution meshes which may contain many large and/or skinny triangles.

We further applied the initial value problem geodesic on constrained texture mapping problem. According to our experimental results, better parametrization will not necessarily lead to better texture mapping. Thus, we did not employ the traditional global optimization and embedding strategy to smooth the parameterizations. Instead, we proposed a Voronoi diagram base domain method for matching positional constraints to texture maps. For each Voronoi area, we applied the discrete exponential map. Thus the final mapping not only isometrically preserves the metric structure of mesh surface but also satisfies the user-defined constraints. We proved that the result texture mapping of our method is one-to-one. In addition, it is able to successfully mapping a face with only four constraints.

6.2 Future Directions

Experimental results have demonstrated that the proposed methods are efficient and effective for the initial value problem of geodesic and its application, and also outperform the well-established baseline methods. Next, we present interesting future directions for improving the dissertation. Firstly, for solving the initial value problem of discrete geodesics, both the geodesics on smooth surface and the discrete shortest geodesics are intrinsic, but the proposed first-order tangent ODE method employs the extrinsic normal direction as coefficient. In addition, for a method that approximates geodesics on the

smooth counterpart, we expect a parameter to measure and control the quality of the approximation. Secondly, the proposed constraint texture mapping method preserves the metric structure of mesh surface through a global Voronoi diagram base domain and local exponential map. However, the angle distortion could be large near the Voronoi boundary, when the matched Voronoi cells between image and surface largely differs in shape.

6.2.1 Intrinsic Integral Invariants for Solving the Initial Value Problem of Discrete Geodesics

Differential invariants of curves and surfaces are widely used in geometry processing. However, the geometry operators of differential invariants [MDSB03] cannot adapt to the choice of resolution and are sensitive to noise. Manay et al. [MHYS04, MCH⁺06] first introduced integral invariants to robustly estimate curvatures on surface. Pottmann et al. [PWHY09] discussed the theoretical stability and numerical robustness of integral invariants, such as curve length, surface area, and volume on a local kernel of disk, sphere, and ball. Figure 6.1 shows a geodesic disk kernel of a point on surface.

With respect to solving the initial problem of discrete geodesics, the merits of integral invariants is threefold. Firstly, integral invariants of curve length and surface area are defined intrinsically. Secondly, the kernel of integral invariants is naturally controlled by a radius parameter which can serve as multi-scale measure. Thirdly, Yang et al. showed that integral invariants are particular useful for the computation of principle directions and curvature [LPW⁺06, YLHP06] as well as geodesic curvature [PWY⁺07]. Therefore, we propose a intrinsic integral invariants method for solving the initial value problem of discrete geodesic. The proposed method is intrinsic and naturally has a kernel radius parameter that can be employed to measure and control the quality of geodesic path.

Figure 6.1: Geodesic disk around a point on triangular mesh.

6.2.2 Constrained Texture Mapping via Optimised Voronoi Diagram Base Domain

Energy minimization is typically used for texture mapping in computer graphics. Generally, the energy functions compromise between the minimization of angle distortion and the minimization of area distortion. However, according to our experimental results, minimizing the energy function does not necessarily lead to good mapping results. Moreover, the semantic connections between texture images and mesh surfaces describes the extra requirements for the parametrization. The extra requirements are not considered in the energy functions.

To date, parametrization methods usually take the positional constraints as either hard or soft constraints, where hard constraints need to be adhered to strictly in the optimization process, while soft constraints can be loosely followed. Though each of the two types of constraints can be handled independently, the integrated semantic structure of all the constraints is useful for the mapping task. However, the structure of the constraints as a integrated unit is also important when we treat each constrain as a feature.

Therefore, exploring the consistence of the integrated structure of all the constraints can potentially improve the texture mapping between images and mesh surfaces.

For the proposed constrained texture mapping method in Chapter 5, the consistence of integrated constraint structure relies on the similarity of the matched Voronoi cells between texture image and mesh surface. We propose to compute a quasi-Voronoi-Diagram on mesh surface by minimizing the shape difference between Voronoi diagrams of mesh surface and image plane. Then the angle distortion near *Voronoi area* boundary can be minimized, and thus better texture mapping result can be achieved, even when there is a large Voronoi diagram inconsistency.

References

- [AZ67] Aleksandr Danilovich Aleksandrov and Viktor A Zalgaller. *Intrinsic geometry of surfaces*, volume 15. American Mathematical Society Providence, RI, 1967.
- [BFH86] James M Beck, Rida T Farouki, and John K Hinds. Surface analysis methods. *Computer Graphics and Applications, IEEE*, 6(12):18–36, 1986.
- [BFRA12] Maria Boschioli, Christoph Fünfzig, Lucia Romani, and Gudrun Albrecht. G 1 rational blend interpolatory schemes: A comparative study. *Graphical models*, 74(1):29–49, 2012.
- [BK07] David Bommes and Leif Kobbelt. Accurate computation of geodesic distance fields for polygonal curves on triangle meshes. In *VMV*, volume 7, pages 151–160, 2007.
- [BPS09] Mukund Balasubramanian, Jonathan R Polimeni, and Eric L Schwartz. Exact geodesics and shortest paths on polyhedral surfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(6):1006–1016, 2009.
- [CH90] Jindong Chen and Yijie Han. Shortest paths on a polyhedron. In *Proceedings of the sixth annual symposium on Computational geometry*, pages 360–369. ACM, 1990.

- [CHK13] Marcel Campen, Martin Heistermann, and Leif Kobbelt. Practical anisotropic geodesy. In *Computer Graphics Forum*, volume 32, pages 63–71. Wiley Online Library, 2013.
- [CHY⁺13] Xi Chen, Si-Yuan He, Ding-Feng Yu, Hong-Cheng Yin, Wei-Dong Hu, and Guo-Qiang Zhu. Geodesic computation on nurbs surfaces for utd analysis. *Antennas and Wireless Propagation Letters, IEEE*, 12:194–197, 2013.
- [CK11] Marcel Campen and Leif Kobbelt. Walking on broken mesh: Defect-tolerant geodesic distances and parameterizations. In *Computer Graphics Forum*, volume 30, pages 623–632. Wiley Online Library, 2011.
- [CWW13] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. Geodesics in heat: a new approach to computing distance based on heat flow. *ACM Transactions on Graphics (TOG)*, 32(5):152, 2013.
- [DC76] Manfredo Perdigao Do Carmo. *Differential geometry of curves and surfaces*. Pearson, 1976.
- [Dij59] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [DLJ⁺07] Junfei Dai, Wei Luo, Miao Jin, Wei Zeng, Ying He, Shing-Tung Yau, and Xianfeng Gu. Geometric accuracy analysis for discrete surface approximation. *Computer Aided Geometric Design*, 24(6):323–338, 2007.
- [DMA02] Mathieu Desbrun, Mark Meyer, and Pierre Alliez. Intrinsic parameterizations of surface meshes. In *Computer Graphics Forum*, volume 21, pages 209–218. Wiley Online Library, 2002.

- [DMT91] Leo Dorst, Indur Mandhyan, and Karen Trovato. The geometrical representation of path planning problems. *Robotics and Autonomous Systems*, 7(2-3):181–195, 1991.
- [ESG01] Ilya Eckstein, Vitaly Surazhsky, and Craig Gotsman. Texture mapping with hard constraints. In *Computer Graphics Forum*, volume 20, pages 95–104. Wiley Online Library, 2001.
- [FH05] Michael S Floater and Kai Hormann. Surface parameterization: a tutorial and survey. In *Advances in multiresolution for geometric modelling*, pages 157–186. Springer, 2005.
- [FHR02] Michael S Floater, Kai Hormann, and Martin Reimers. Parameterization of manifold triangulations. In *Approximation Theory X: Abstract and Classical Analysis*, pages 197–209. Vanderbilt University Press, Nashville, 2002.
- [FMHF08] Christoph Fünfzig, Kerstin Müller, Dianne Hansford, and Gerald Farin. PNG1 triangles for tangent plane continuous surfaces on the GPU. In *Proceedings of graphics interface 2008*, pages 219–226. Canadian Information Processing Society, 2008.
- [GKMT01] Taro Goto, Sumedha Kshirsagar, and Nadia Magnenat-Thalmann. Automatic face cloning and animation using real-time facial feature tracking and speech acquisition. *Signal Processing Magazine, IEEE*, 18(3):17–25, 2001.
- [GWB⁺14] Erwin Groot, Brian Wyvill, Loïc Barthe, Ahmad Nasri, and Paul Lalonde. Implicit decals: Interactive editing of repetitive patterns on surfaces. In

- Computer Graphics Forum*, volume 33, pages 141–151. Wiley Online Library, 2014.
- [HH00] Ingrid Hotz and Hans Hagen. Visualizing geodesics. In *Proceedings of the conference on Visualization'00*, pages 311–318. IEEE Computer Society Press, 2000.
- [HPW06] Klaus Hildebrandt, Konrad Polthier, and Max Wardetzky. On the convergence of metric and geometric properties of polyhedral surfaces. *Geometriae Dedicata*, 123(1):89–112, 2006.
- [HS98] John Hershberger and Subhash Suri. Practical methods for approximating shortest paths on a convex polytope in $\mathbb{R}^3$. *Computational Geometry*, 10(1):31–46, 1998.
- [Kos03] AF Koschan. Perception-based 3D triangle mesh segmentation using fast marching watersheds. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–27. IEEE, 2003.
- [KS98] Ron Kimmel and James A Sethian. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences*, 95(15):8431–8435, 1998.
- [KSG03] Vladislav Kraevoy, Alla Sheffer, and Craig Gotsman. Matchmaker: Constructing constrained texture maps. In *ACM SIGGRAPH 2003 Papers, SIGGRAPH '03*, pages 326–333, New York, NY, USA, 2003. ACM.
- [KZW12] Tsz-Ho Kwok, Yunbo Zhang, and Charlie CL Wang. Constructing common base domain by cues from voronoi diagram. *Graphical Models*, 74(4):152–163, 2012.

- [LCT11a] Yong-Jin Liu, Zhanqing Chen, and Kai Tang. Construction of iso-contours, bisectors, and Voronoi diagrams on triangulated surfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1502–1517, 2011.
- [LCT11b] Yong-Jin Liu, Zhanqing Chen, and Kai Tang. Construction of iso-contours, bisectors, and voronoi diagrams on triangulated surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1502–1517, 2011.
- [Lév01] Bruno Lévy. Constrained texture mapping for polygonal meshes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 417–424, New York, NY, USA, 2001. ACM.
- [LHL06] Sungyeol Lee, Joonhee Han, and Haeyoung Lee. Straightest paths on meshes by cutting planes. In *Geometric Modeling and Processing-GMP 2006*, pages 609–615. Springer, 2006.
- [Liu13] Yong-Jin Liu. Exact geodesic metric in 2-manifold triangle meshes using edge-based data structures. *Computer-Aided Design*, 45(3):695–704, 2013.
- [Liu15] Yong-Jin Liu. Semi-continuity of skeletons in 2-manifold and discrete Voronoi approximation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, DOI: 10.1109/TPAMI.2015.2430342, 2015.
- [LKMD01] Haeyoung Lee, Laehyun Kim, Mark Meyer, and Mathieu Desbrun. Meshes on fire. In *Computer Animation and Simulation 2001*, pages 75–84. Springer, 2001.
- [LM98] Bruno Lévy and Jean-Laurent Mallet. Non-distorted texture mapping for sheared triangulated meshes. In *Proceedings of the 25th Annual Conference*

- on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 343–352, New York, NY, USA, 1998. ACM.
- [LMS97] Mark Lanthier, Anil Maheshwari, and Jörg-Rüdiger Sack. Approximating weighted shortest paths on polyhedral surfaces. In *Proceedings of the thirteenth annual symposium on Computational geometry*, pages 274–283. ACM, 1997.
- [LPRM02] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérôme Maillot. Least squares conformal maps for automatic texture atlas generation. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 362–371. ACM, 2002.
- [LPW⁺06] Yang Liu, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang. Geometric modeling with conical meshes and developable surfaces. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 681–689. ACM, 2006.
- [LRF10] Yaron Lipman, Raif M Rustamov, and Thomas A Funkhouser. Biharmonic distance. *ACM Transactions on Graphics (TOG)*, 29(3):27, 2010.
- [LT09] André Lieutier and Boris Thibert. Convergence of geodesics on triangulations. *Computer Aided Geometric Design*, 26(4):412–424, 2009.
- [LWZ11] Yuen-Shan Leung, Charlie CL Wang, and Yunbo Zhang. Localized construction of curved surfaces from polygon meshes: A simple and practical approach on GPU. *Computer-Aided Design*, 43(6):573–585, 2011.
- [LYY08] Tong-Yee Lee, Shao-Wei Yen, and I-Cheng Yeh. Texture mapping with hard constraints using warping scheme. *Visualization and Computer Graphics, IEEE Transactions on*, 14(2):382–395, 2008.

- [LZH07] Yong-Jin Liu, Qian-Yi Zhou, and Shi-Min Hu. Handling degenerate cases in exact geodesic computation on triangle meshes. *The Visual Computer*, 23(9-11):661–668, 2007.
- [Mae96] Takashi Maekawa. Computation of shortest paths on free-form parametric surfaces. *Journal of mechanical design*, 118(4):499–508, 1996.
- [MCH⁺06] Siddharth Manay, Daniel Cremers, Byung-Woo Hong, Anthony J Yezzi, and Stefano Soatto. Integral invariants for shape matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1602–1618, 2006.
- [MDSB03] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003.
- [MHYS04] Siddharth Manay, Byung-Woo Hong, Anthony J Yezzi, and Stefano Soatto. *Integral invariant signatures*. Springer, 2004.
- [MMP87] Joseph SB Mitchell, David M Mount, and Christos H Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987.
- [MR12] Eivind Lyche Melvær and Martin Reimers. Geodesic polar coordinates on polygonal meshes. In *Computer Graphics Forum*, volume 31, pages 2423–2435. Wiley Online Library, 2012.
- [MS01] Facundo Mémoli and Guillermo Sapiro. Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. *Journal of computational Physics*, 173(2):730–764, 2001.

- [MVC05] Dimas Martínez, Luiz Velho, and Paulo C Carvalho. Computing geodesics on triangular meshes. *Computers & Graphics*, 29(5):667–675, 2005.
- [MZX15] Yuewen Ma, Jianmin Zheng, and Jian Xie. Foldover-free mesh warping for constrained texture mapping. *Visualization and Computer Graphics, IEEE Transactions on*, 21(3):375–388, 2015.
- [PB89] NM Patrikalakis and L Bardis. Offsets of curves on rational b-spline surfaces. *Engineering with Computers*, 5(1):39–46, 1989.
- [PC05] Gabriel Peyré and Laurent Cohen. Geodesic computations for fast and accurate surface remeshing and parameterization. In *Elliptic and Parabolic Problems*, pages 157–171. Springer, 2005.
- [PC06] Gabriel Peyré and Laurent D Cohen. Geodesic remeshing using front propagation. *International Journal of Computer Vision*, 69(1):145–156, 2006.
- [PM09] Nicholas M Patrikalakis and Takashi Maekawa. *Shape interrogation for computer aided design and manufacturing*. Springer Science & Business Media, 2009.
- [Pre07] William H Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
- [PS99] Konrad Polthier and Markus Schmies. Geodesic flow on polyhedral surfaces. In *Data Visualization*, pages 179–188. Springer, 1999.
- [PS06] Konrad Polthier and Markus Schmies. *Straightest geodesics on polyhedral surfaces*. ACM, 2006.

- [PWHY09] Helmut Pottmann, Johannes Wallner, Qi-Xing Huang, and Yong-Liang Yang. Integral invariants for robust geometry processing. *Computer Aided Geometric Design*, 26(1):37–60, 2009.
- [PWY⁺07] Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, Yu-Kun Lai, and Shi-Min Hu. Principal curvatures from the integral invariant viewpoint. *Computer Aided Geometric Design*, 24(8):428–442, 2007.
- [RKSDH⁺03] GVV Ravi Kumar, Prabha Srinivasan, V Devaraja Holla, KG Shastri, and BG Prakash. Geodesic curve computations on surfaces. *Computer Aided Geometric Design*, 20(2):119–133, 2003.
- [Rus04] Szymon Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Proceedings of Symposium on 3D Data Processing, Visualization and Transmission*, pages 486–493. IEEE, 2004.
- [Sch13] Ryan Schmidt. Stroke parameterization. In *Computer Graphics Forum*, volume 32, pages 255–263. Wiley Online Library, 2013.
- [Set96] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.
- [SGW06] Ryan Schmidt, Cindy Grimm, and Brian Wyvill. Interactive decal compositing with discrete exponential maps. *ACM Transactions on Graphics (TOG)*, 25(3):605–613, 2006.
- [SHL⁺07] Alla Sheffer, K Hormann, B Levy, M Desbrun, K Zhou, E PRAUN, and H HOPPE. Mesh parameterization: Theory and practice. *ACM SIG-GRAPPH, course notes*, 2007.

- [Spi70] Michael Spivak. A comprehensive introduction to differential geometry, vol. 2. *I (Boston, Mass., 1970)*, 1970.
- [SPR06] Alla Sheffer, Emil Praun, and Kenneth Rose. Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision*, 2(2):105–171, 2006.
- [SRD08] J Sánchez-Reyes and R Dorado. Constrained design of polynomial surfaces from geodesic curves. *Computer-Aided Design*, 40(1):49–55, 2008.
- [SRIC01] Peter-Pike J Sloan, Charles F Rose III, and Michael F Cohen. Shape by example. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 135–143. ACM, 2001.
- [SS86] Micha Sharir and Amir Schorr. On shortest paths in polyhedral spaces. *SIAM Journal on Computing*, 15(1):193–215, 1986.
- [SSG03] Oren Sifri, Alla Sheffer, and Craig Gotsman. Geodesic-based surface remeshing. In *In Proc. 12th Intl. Meshing Roundtable*. Citeseer, 2003.
- [SSK⁺05] Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J Gortler, and Hugues Hoppe. Fast exact and approximate geodesics on meshes. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 553–560. ACM, 2005.
- [SZZ⁺13] Qian Sun, Long Zhang, Minqi Zhang, Xiang Ying, Shi-Qing Xin, Jiazhi Xia, and Ying He. Texture brush: an interactive surface texturing interface. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 153–160. ACM, 2013.

- [VA00] Kasturi R Varadarajan and Pankaj K Agarwal. Approximating shortest paths on a nonconvex polyhedron. *SIAM Journal on Computing*, 30(4):1321–1340, 2000.
- [VPBM01] Alex Vlachos, Jörg Peters, Chas Boyd, and Jason L Mitchell. Curved pn triangles. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 159–166. ACM, 2001.
- [XHF12] Shi-Qing Xin, Ying He, and Chi-Wing Fu. Efficiently computing exact geodesic loops within finite steps. *Visualization and Computer Graphics, IEEE Transactions on*, 18(6):879–889, 2012.
- [XW09] Shi-Qing Xin and Guo-Jin Wang. Improving Chen and Han’s algorithm on the discrete geodesic problem. *ACM Transactions on Graphics (TOG)*, 28(4):104, 2009.
- [XWL⁺15] Chun-Xu Xu, Tuanfeng Y. Wang, Yong-Jin Liu, Ligang Liu, and Ying He. Fast wavefront propagation (FWP) for computing exact discrete geodesics on meshes. *IEEE Transactions on Visualization and Computer Graphics*, 2015.
- [XYH11] Shi-Qing Xin, Xiang Ying, and Ying He. Efficiently computing geodesic offsets on triangle meshes by the extended Xin–Wang algorithm. *Computer-Aided Design*, 43(11):1468–1476, 2011.
- [XYH12] Shi-Qing Xin, Xiang Ying, and Ying He. Constant-time all-pairs geodesic distance query on triangle meshes. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 31–38. ACM, 2012.

- [YLHP06] Yong-Liang Yang, Yu-Kun Lai, Shi-Min Hu, and Helmut Pottmann. Robust principal curvatures on multiple scales. In *Symposium on Geometry Processing*, pages 223–226, 2006.
- [YWH13] Xiang Ying, Xiaoning Wang, and Ying He. Saddle vertex graph (SVG): A novel solution to the discrete geodesic problem. *ACM Transactions on Graphics*, 32(6):Article No. 170, 2013.
- [YXH14] Xiang Ying, Shi-Qing Xin, and Ying He. Parallel Chen-Han (PCH) algorithm for discrete geodesics. *ACM Transactions on Graphics (TOG)*, 33(1):9, 2014.
- [ZB06] Yu Zhang and Norman I Badler. Synthesis of 3d faces using region-based morphing under intuitive control. *Computer Animation and Virtual Worlds*, 17(3-4):421–432, 2006.
- [ZWT⁺05] Kun Zhou, Xi Wang, Yiying Tong, Mathieu Desbrun, Baining Guo, and Heung-Yeung Shum. Texturemontage. *ACM Transactions on Graphics (TOG)*, 24(3):1148–1155, 2005.

Appendix A

Derivation of Numerical Geodesic of Cubic Bezier Patch

A.1 General Parametric Surface

Let $r(u, v)$ be an orthogonal parameterization of surface $S(u, v)$. Given an point $P_0 = r(u_0, v_0)$ on the surface and a tangent direction $T_0 = \frac{r_u}{\|r_u\|} \cos \theta + \frac{r_v}{\|r_v\|} \sin \theta$ at P_0 . Here θ is the angle between T_0 and r_u . Geodesic curve $\gamma(u, v)$ on surface starts at $\gamma(u_0, v_0) = P_0$ and extends along the direction of $\gamma'(u_0, v_0) = T_0$ is uniquely defined by geodesic equation. When the parametric chart is orthogonal, geodesic equation can be derived to a first-order ODE by Liouville's equation of geodesic curvature,

$$\left\{ \begin{array}{l} \frac{d\theta}{ds} = \frac{1}{2\sqrt{G}} \frac{\partial \ln E}{\partial v} \cos \theta - \frac{1}{2\sqrt{E}} \frac{\partial \ln G}{\partial u} \sin \theta, \\ \frac{du}{ds} = \frac{1}{\sqrt{E}} \cos \theta, \\ \frac{dv}{ds} = \frac{1}{\sqrt{G}} \sin \theta. \end{array} \right. \quad (\text{A.1})$$

where E and G are the coefficients of the first fundamental form of surface with $E = r_u \cdot r_u$ and $G = r_v \cdot r_v$. With the initial conditions of $u(0) = u_0, v(0) = v_0$ and $\theta(0) = \theta_0$, unique solution of the equation will be $u = u(s), v = v(s), \theta = \theta(s)$. As geodesic is second-order property within a infinitesimal neighborhood of the point, we adopt a fourth-order

Runge-Kutta method numerically solve this geodesic ODE. Let

$$\left\{ \begin{array}{l} \frac{d\theta}{ds} = \frac{1}{2\sqrt{G}} \frac{\partial \ln E}{\partial v} \cos \theta - \frac{1}{2\sqrt{E}} \frac{\partial \ln G}{\partial u} \sin \theta = F_\theta, \\ \frac{du}{ds} = \frac{1}{\sqrt{E}} \cos \theta = F_u, \\ \frac{dv}{ds} = \frac{1}{\sqrt{G}} \sin \theta = F_v. \end{array} \right. \quad (\text{A.2})$$

where F_θ, F_u, F_v are constant at each point. To a certain iteration step length δ , the numerical geodesic on surface is a point series $\{P_i\}$.

Protocol 3 Runge-Kutta Numerical Geodesic on Surface

Require: Orthogonal parametric surface $r(u, v)$, the initial point $P_0 = r(u_0, v_0)$, tangent direction the tangent direction $T_0 = \frac{r_u}{\|r_u\|} \cos \theta_0 + \frac{r_v}{\|r_v\|} \sin \theta_0$ at P_0 , the step length δ , and the maximal geodesic distance L

Ensure: Numerical geodesic point series $\{P_i\}$

$u = u_0;$

$v = v_0;$

$P_0 = r(u, v);$

$Length = 0;$

for $i = 0; i < \lceil \frac{L}{\delta} \rceil; i++$ **do**

$F_\theta^1 = F_\theta(u, v, \theta);$

$F_u^1 = F_u(u, v, \theta);$

$F_v^1 = F_v(u, v, \theta);$

$F_\theta^2 = F_\theta(u + \frac{1}{2}\delta F_u^1, v + \frac{1}{2}\delta F_v^1, \theta + \frac{1}{2}F_\theta^1);$

$F_u^2 = F_u(u + \frac{1}{2}\delta F_u^1, v + \frac{1}{2}\delta F_v^1, \theta + \frac{1}{2}F_\theta^1);$

$F_v^2 = F_v(u + \frac{1}{2}\delta F_u^1, v + \frac{1}{2}\delta F_v^1, \theta + \frac{1}{2}F_\theta^1);$

$F_\theta^3 = F_\theta(u + \frac{1}{2}\delta F_u^2, v + \frac{1}{2}\delta F_v^2, \theta + \frac{1}{2}F_\theta^2);$

$F_u^3 = F_u(u + \frac{1}{2}\delta F_u^2, v + \frac{1}{2}\delta F_v^2, \theta + \frac{1}{2}F_\theta^2);$

$F_v^3 = F_v(u + \frac{1}{2}\delta F_u^2, v + \frac{1}{2}\delta F_v^2, \theta + \frac{1}{2}F_\theta^2);$

$F_\theta^4 = F_\theta(u + F_u^3\delta, v + F_v^3\delta, \theta + F_\theta^3);$

$F_u^4 = F_u(u + \delta F_u^3, v + \delta F_v^3, \theta + F_\theta^3);$

$F_v^4 = F_v(u + \delta F_u^3, v + \delta F_v^3, \theta + F_\theta^3);$

$\theta_{i+1} = \theta_i + \frac{1}{6}\delta(F_\theta^1 + 2F_\theta^2 + 2F_\theta^3 + F_\theta^4);$

$u = u + \frac{1}{6}\delta(F_u^1 + 2F_u^2 + 2F_u^3 + F_u^4);$

$v = v + \frac{1}{6}\delta(F_v^1 + 2F_v^2 + 2F_v^3 + F_v^4);$

$P_i = r(u, v);$

end for

A.2 Cubic Bézier Patch

For an example, we derive the exact formula of numerical geodesic on cubic *Bézier* patch,

$$B(\alpha, \beta) = \sum_{i=0}^3 \sum_{j=0}^{3-i} \mathbf{C}_{i,j,k} \frac{3!}{i!j!k!} \alpha^i \beta^j \gamma^k \quad (\text{A.3})$$

Where $i + j + k = 3$ and $\alpha + \beta + \gamma = 1$. To simplify the expression, we shear the parameters α, β to a orthogonal coordinate system of u, v by,

$$\begin{cases} \alpha = \frac{u}{c}, \\ \beta = \frac{v}{a}. \end{cases} \quad (\text{A.4})$$

where $a = \|BC\|$ and $c = \|AB\| \tan \theta$. Then B_u, B_v are derived as:

$$\begin{aligned} B_u = & \frac{1}{c} \sum_{i=1}^3 \sum_{j=0}^{3-i} \frac{3!}{(i-1)!j!k!} \mathbf{C}_{i,j,k} \left(\frac{\mathbf{u}}{c}\right)^{i-1} \left(\frac{\mathbf{v}}{a}\right)^j \left(1 - \frac{\mathbf{u}}{c} - \frac{\mathbf{v}}{a}\right)^k \\ & - \frac{1}{c} \sum_{i=0}^2 \sum_{j=0}^{2-i} \frac{3!}{i!j!(k-1)!} \mathbf{C}_{i,j,k} \left(\frac{\mathbf{u}}{c}\right)^i \left(\frac{\mathbf{v}}{a}\right)^j \left(1 - \frac{\mathbf{u}}{c} - \frac{\mathbf{v}}{a}\right)^{k-1} \end{aligned} \quad (\text{A.5})$$

$$\begin{aligned} B_v = & \frac{1}{a} \sum_{j=1}^3 \sum_{i=0}^{3-j} \frac{3!}{i!(j-1)!k!} \mathbf{C}_{i,j,k} \left(\frac{\mathbf{u}}{c}\right)^i \left(\frac{\mathbf{v}}{a}\right)^{j-1} \left(1 - \frac{\mathbf{u}}{c} - \frac{\mathbf{v}}{a}\right)^k \\ & - \frac{1}{a} \sum_{j=0}^2 \sum_{i=0}^{2-j} \frac{3!}{i!j!(k-1)!} \mathbf{C}_{i,j,k} \left(\frac{\mathbf{u}}{c}\right)^i \left(\frac{\mathbf{v}}{a}\right)^j \left(1 - \frac{\mathbf{u}}{c} - \frac{\mathbf{v}}{a}\right)^{k-1} \end{aligned} \quad (\text{A.6})$$

Or

$$\begin{aligned}
 B_u &= \frac{3}{c}C_{0,3,0}\left(\frac{u}{c}\right)^2 - \frac{3}{c}C_{1,2,0}\left(\frac{u}{c}\right)^2 \\
 &\quad + \frac{6}{c}C_{1,2,0}\frac{u}{c}\left(1 - \frac{u}{c} - \frac{v}{a}\right) - \frac{6}{c}C_{2,1,0}\frac{u}{c}\left(1 - \frac{u}{c} - \frac{v}{a}\right) \\
 &\quad + \frac{3}{c}C_{2,1,0}\left(1 - \frac{u}{c} - \frac{v}{a}\right)^2 - \frac{3}{c}C_{3,0,0}\left(1 - \frac{u}{c} - \frac{v}{a}\right)^2 \\
 &\quad + \frac{6}{c}C_{0,2,1}\frac{u}{c}\frac{v}{a} - \frac{6}{c}C_{1,1,1}\frac{u}{c}\frac{v}{a} \\
 &\quad + \frac{6}{c}C_{1,1,1}\frac{v}{a}\left(1 - \frac{u}{c} - \frac{v}{a}\right) - \frac{6}{c}C_{2,0,1}\frac{v}{a}\left(1 - \frac{u}{c} - \frac{v}{a}\right) \\
 &\quad + \frac{3}{c}C_{0,1,2}\left(\frac{v}{a}\right)^2 - \frac{3}{c}C_{1,0,2}\left(\frac{v}{a}\right)^2 \\
 &= \frac{3}{c^3}(C_{0,3,0} - C_{1,2,0})u^2 + \frac{6}{c^2}(C_{1,2,0} - C_{2,1,0})u\left(1 - \frac{u}{c} - \frac{v}{a}\right) \\
 &\quad + \frac{3}{c}(C_{2,1,0} - C_{3,0,0})\left(1 - \frac{u}{c} - \frac{v}{a}\right)^2 + \frac{6}{ac^2}(C_{0,2,1} - C_{1,1,1})uv \\
 &\quad + \frac{6}{ac}(C_{1,1,1} - C_{2,0,1})v\left(1 - \frac{u}{c} - \frac{v}{a}\right) + \frac{3}{a^2c}(C_{0,1,2} - C_{1,0,2})v^2
 \end{aligned} \tag{A.7}$$

$$\begin{aligned}
 B_v &= \frac{3}{a}C_{0,2,1}\left(\frac{u}{c}\right)^2 - \frac{3}{a}C_{1,2,0}\left(\frac{u}{c}\right)^2 \\
 &\quad + \frac{6}{a}C_{1,1,1}\frac{u}{c}\left(1 - \frac{u}{c} - \frac{v}{a}\right) - \frac{6}{a}C_{2,1,0}\frac{u}{c}\left(1 - \frac{u}{c} - \frac{v}{a}\right) \\
 &\quad + \frac{3}{a}C_{2,0,1}\left(1 - \frac{u}{c} - \frac{v}{a}\right)^2 - \frac{3}{a}C_{3,0,0}\left(1 - \frac{u}{c} - \frac{v}{a}\right)^2 \\
 &\quad + \frac{6}{a}C_{0,1,2}\frac{u}{c}\frac{v}{a} - \frac{6}{a}C_{1,1,1}\frac{u}{c}\frac{v}{a} \\
 &\quad + \frac{6}{a}C_{1,0,2}\frac{v}{a}\left(1 - \frac{u}{c} - \frac{v}{a}\right) - \frac{6}{a}C_{2,0,1}\frac{v}{a}\left(1 - \frac{u}{c} - \frac{v}{a}\right) \\
 &\quad + \frac{3}{a}C_{0,0,3}\left(\frac{v}{a}\right)^2 - \frac{3}{a}C_{1,0,2}\left(\frac{v}{a}\right)^2 \\
 &= \frac{3}{ac^2}(C_{0,2,1} - C_{1,2,0})u^2 + \frac{6}{ac}(C_{1,1,1} - C_{2,1,0})u\left(1 - \frac{u}{c} - \frac{v}{a}\right) \\
 &\quad + \frac{3}{a}(C_{2,0,1} - C_{3,0,0})\left(1 - \frac{u}{c} - \frac{v}{a}\right)^2 + \frac{6}{a^2c}(C_{0,1,2} - C_{1,1,1})uv \\
 &\quad + \frac{6}{a^2}(C_{1,0,2} - C_{2,0,1})v\left(1 - \frac{u}{c} - \frac{v}{a}\right) + \frac{3}{a^3}(C_{0,0,3} - C_{1,0,2})v^2
 \end{aligned} \tag{A.8}$$

The geodesic ODE is then,

$$\left\{ \begin{array}{l} \frac{d\theta}{ds} = \frac{1}{2\sqrt{B_v \cdot B_v}} \frac{\partial \ln(B_u \cdot B_u)}{\partial v} \cos \theta - \frac{1}{2\sqrt{B_u \cdot B_u}} \frac{\partial \ln(B_v \cdot B_v)}{\partial u} \sin \theta, \\ \frac{du}{ds} = \frac{1}{\sqrt{B_u \cdot B_u}} \cos \theta, \\ \frac{dv}{ds} = \frac{1}{\sqrt{B_v \cdot B_v}} \sin \theta. \end{array} \right. \quad (\text{A.9})$$

Appendix B

Numerical Method Computing Geodesics on Parametric Surface

Given a parametric surface $S(u, v)$, a curve $r(t)$ lying on the surface is expressed in parametric form as:

$$\begin{cases} u = u(t), \\ v = v(t). \end{cases} \quad (\text{B.1})$$

Let the first fundamental form coefficients $E = r_u \cdot r_u$, $F = r_u \cdot r_v$, and $G = r_v \cdot r_v$, we have the arc length ds^2 of the curve and the length lying on the surface.

$$ds^2 = E du^2 + 2F du dv + G dv^2. \quad (\text{B.2})$$

$$L = \int (E \dot{u}^2 + 2F \dot{u} \dot{v} + G \dot{v}^2)^{1/2} dt \quad (\text{B.3})$$

Here, the dots operation denotes differentiation with respect to the parameter t . To minimize the length L , let $\Gamma = E \dot{u}^2 + 2F \dot{u} \dot{v} + G \dot{v}^2$, $u(t), v(t)$ will satisfy the Euler equations.

$$\begin{cases} \frac{d}{dt} \frac{\partial \Gamma}{\partial \dot{u}} - \frac{\partial \Gamma}{\partial u} = 0, \\ \frac{d}{dt} \frac{\partial \Gamma}{\partial \dot{v}} - \frac{\partial \Gamma}{\partial v} = 0. \end{cases} \quad (\text{B.4})$$

To simplify Equation B.4, we choose arc length parameter s along the curve. Then Equation B.4 can be expressed into a system of nonlinear differential equations. See Equation B.5.

$$\begin{cases} \frac{d^2u}{ds^2} + d_{11}\left(\frac{du}{ds}\right)^2 + 2d_{12}\left(\frac{du}{ds}\right)\left(\frac{dv}{ds}\right) + d_{22}\left(\frac{dv}{ds}\right)^2 = 0, \\ \frac{d^2v}{ds^2} + e_{11}\left(\frac{du}{ds}\right)^2 + 2e_{12}\left(\frac{du}{ds}\right)\left(\frac{dv}{ds}\right) + e_{22}\left(\frac{dv}{ds}\right)^2 = 0. \end{cases} \quad (\text{B.5})$$

where $n = \frac{r_u \times r_v}{|r_u \times r_v|}$ and the symbols $d_{11}, d_{12}, d_{22}, e_{11}, e_{12}, \text{ and } e_{22}$ are:

$$\begin{cases} d_{11} = \frac{n \cdot (r_{uu} \times r_v)}{|r_u \times r_v|}, \\ d_{12} = \frac{n \cdot (r_{uv} \times r_v)}{|r_u \times r_v|}, \\ d_{22} = \frac{n \cdot (r_{vv} \times r_v)}{|r_u \times r_v|}, \\ e_{11} = \frac{n \cdot (r_u \times r_{uu})}{|r_u \times r_v|}, \\ e_{12} = \frac{n \cdot (r_u \times r_{uv})}{|r_u \times r_v|}, \\ e_{22} = \frac{n \cdot (r_u \times r_{vv})}{|r_u \times r_v|}. \end{cases} \quad (\text{B.6})$$

Thus, the geodesics are traced by solving the initial value problem of four first-order differential equations in u, v, u_1, v_1 . Let $U_1 = -r_v \cdot \mathbf{t}$, $\mathbf{V}_1 = -\mathbf{r}_u \cdot \mathbf{t}$, the geodesic ordinary equation system is Equation B.8 subject to the initial conditions in Equation B.7.

$$\begin{cases} (u(0), v(0)) = (u^0, v^0) = \gamma(t_0), \\ u_1(0) = u_1^0 = \pm \frac{U_1}{(E U_1^2 + 2F U_1 V_1 + G V_1^2)^{1/2}}, \\ v_1(0) = v_1^0 = \pm \frac{V_1}{(E U_1^2 + 2F U_1 V_1 + G V_1^2)^{1/2}}. \end{cases} \quad (\text{B.7})$$

$$\left\{ \begin{array}{l} \frac{du}{ds} = u_1 = F_1(u, v, u_1, v_1), \\ \frac{dv}{ds} = v_1 = F_2(u, v, u_1, v_1), \\ \frac{du_1}{ds} = -d_{11}u_1^2 - 2d_{12}u_1v_1 - d_{22}v_1^2 = F_3(u, v, u_1, v_1), \\ \frac{dv_1}{ds} = -e_{11}u_1^2 - 2e_{12}u_1v_1 - e_{22}v_1^2 = F_4(u, v, u_1, v_1). \end{array} \right. \quad (\text{B.8})$$

When dealing with a curve $r(t)$ on cubic Bezier patch $B(u, v) = \sum_{i=0}^3 \sum_{j=0}^{3-i} \frac{3!b_{i,j,k}}{i!j!k!} u^i v^j (1 - u - v)^k$, the deferential of the curve r_u, r_v are derived as Equation B.9.

$$\left\{ \begin{array}{l} r_u = \sum_{i=1}^3 \sum_{j=0}^{3-i} \frac{3!b_{i,j,k}}{(i-1)!j!k!} u^{i-1} v^j (1 - u - v)^k - \sum_{i=0}^2 \sum_{j=0}^{2-i} \frac{3!b_{i,j,k}}{i!j!(k-1)!} u^i v^j (1 - u - v)^{k-1}, \\ r_v = \sum_{j=1}^3 \sum_{i=0}^{3-j} \frac{3!b_{i,j,k}}{i!(j-1)!k!} u^i v^{j-1} (1 - u - v)^k - \sum_{j=0}^2 \sum_{i=0}^{2-j} \frac{3!b_{i,j,k}}{i!j!(k-1)!} u^i v^j (1 - u - v)^{k-1}. \end{array} \right. \quad (\text{B.9})$$

The first fundamental coefficients of $B(u, v)$ are:

$$\left\{ \begin{aligned} E = r_u \cdot r_u &= \sum_{i=2}^3 \sum_{j=0}^{3-i} \frac{3!b_{i,j,k}}{(i-2)!j!k!} u^{i-2} v^j (1-u-v)^k \\ &\quad - \sum_{i=1}^2 \sum_{j=0}^{2-i} \frac{2 * 3!b_{i,j,k}}{(i-1)!j!(k-1)!} u^{i-1} v^j (1-u-v)^{k-1} \\ &\quad + \sum_{i=0}^1 \sum_{j=0}^{1-i} \frac{3!b_{i,j,k}}{(i-1)!j!(k-2)!} u^{i-1} v^j (1-u-v)^{k-2}, \\ F = r_u \cdot r_v &= \sum_{i=1}^3 \sum_{j=1}^{3-i} \frac{3!b_{i,j,k}}{(i-1)!(j-1)!k!} u^{i-1} v^{j-1} (1-u-v)^k \\ &\quad - \sum_{i=1}^2 \sum_{j=0}^{2-i} \frac{3!b_{i,j,k}}{(i-1)!j!(k-1)!} u^{i-1} v^j (1-u-v)^{k-1} \\ &\quad - \sum_{i=0}^2 \sum_{j=1}^{2-i} \frac{3!b_{i,j,k}}{i!(j-1)!(k-1)!} u^i v^{j-1} (1-u-v)^{k-1} \\ &\quad + \sum_{i=0}^1 \sum_{j=0}^{1-i} \frac{3!b_{i,j,k}}{i!j!(k-2)!} u^i v^j (1-u-v)^{k-2}, \\ G = r_v \cdot r_v &= \sum_{j=2}^3 \sum_{i=0}^{3-j} \frac{3!b_{i,j,k}}{i!(j-2)!k!} u^i v^{j-2} (1-u-v)^k \\ &\quad - \sum_{j=1}^2 \sum_{i=0}^{2-j} \frac{2 * 3!b_{i,j,k}}{i!(j-1)!(k-1)!} u^i v^{j-1} (1-u-v)^{k-1} \\ &\quad + \sum_{j=0}^1 \sum_{i=0}^{1-j} \frac{3!b_{i,j,k}}{i!j!(k-2)!} u^i v^j (1-u-v)^{k-2}. \end{aligned} \right. \quad (\text{B.10})$$