

Solving the initial value problem of discrete geodesics[☆]



Peng Cheng^{a,*}, Chunyan Miao^a, Yong-Jin Liu^b, Changhe Tu^c, Ying He^a

^a School of Computer Engineering, Nanyang Technological University, Singapore

^b TNList, Department of Computer Science and Technology, Tsinghua University, Beijing, China

^c School of Computer Science and Technology, Shandong University, Jinan, China

HIGHLIGHTS

- Shortest geodesic is not able to solve the initial value problem of discrete geodesic.
- Geodesic equation are second-order ODEs.
- We solve the initial value problem on triangle meshes by solving a first-order ODE
- The computed discrete geodesic path converges to the one on the smooth surface.

ARTICLE INFO

Keywords:

Discrete geodesics
Initial value problem
Geodesic polar coordinates

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 to solve the boundary value problem, i.e., to find the shortest path between two given points. In this paper, we focus on the initial value problem, i.e., finding a uniquely determined geodesic path from a given point in any direction. Since the *shortest* paths do not provide the unique solution on triangle meshes, we solve the initial value problem in an indirect manner: 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 \hat{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 vector 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 directly transported to the neighboring patch, thanks to the 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 are difficult to implement due to the 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 for implementation. 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.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Computing geodesic distances and geodesic paths plays an important role in many fields, such as CAD/CAM [1], path planning [2], shape analysis [3], parameterization [4,5], segmentation [6], and medial axis [7]. Geodesics on smooth surfaces are well understood in classic differential geometry. However, the dis-

crete 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, 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 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 problem has been studied extensively in the past three decades. To date, many elegant algorithms have been proposed. Representative works include the

[☆] This paper has been recommended for acceptance by Scott Schaefer and Charlie C.L. Wang.

* Corresponding author.

E-mail address: clearbunny@gmail.com (P. Cheng).

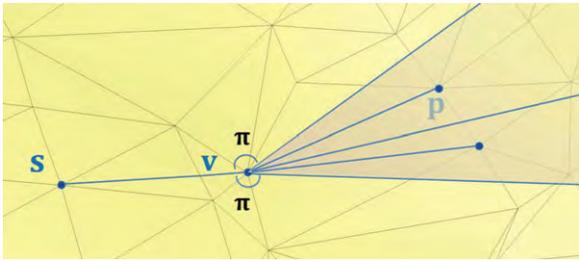


Fig. 1. The discrete *shortest* geodesic does not solve the initial value problem. Let v be a saddle vertex, whose curve angle is more than 2π . When a shortest geodesic path, say $\gamma(s, v)$, passes through v , it splits into many outgoing geodesic paths: any line segment \overline{pv} in the fan-shaped area (in gray) 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.

exact¹ algorithms (e.g., the Mitchell–Mount–Papadimitriou (MMP) algorithm [8] and the Chen–Han (CH) algorithm [9]), the PDE methods (e.g., the fast marching method [10] and the heat method [11,12]), and the graph-theoretic methods (e.g., the saddle vertex graph method [13]). These algorithms, however, are mainly designed to solve the boundary value problem, that is, to find the shortest path between two fixed endpoints.

Mitchell et al. [8] proved that the general form of a *shortest* geodesic path γ was an alternating sequence of vertices and (possibly empty) edges. The unfolded image of the path along any edge sequence is a straight line segment, and the curve angle of any vertex which γ passes through is greater than or equal to π . As Fig. 1 shows, when a shortest geodesic path γ passes through a saddle vertex (whose curve 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 paper, 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 $\hat{\gamma}$ on a piecewise smooth surface \hat{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 vector by 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 vector can be directly transported to the neighboring patch, thanks to the 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 the triangle mesh M . See Fig. 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 observe 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. In addition, our method can be easily adapted to work on non-orientable surfaces.

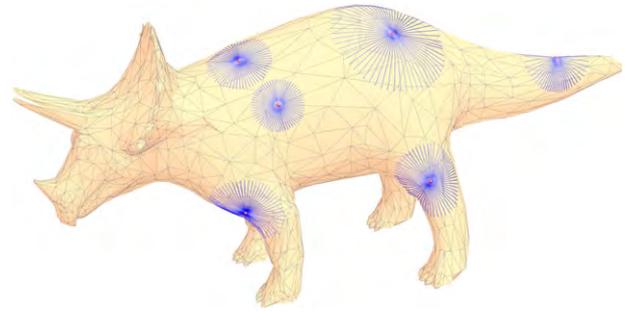


Fig. 2. Our method solves the initial value problem of discrete geodesics. For each point (in red) on the Rhino model, we compute geodesic paths in 60 tangent directions, which are evenly sampled on the tangent plane. Each tangent direction is guaranteed to produce a unique geodesic path on the triangle mesh. Our method is numerically stable and works well on meshes with large and/or skinny triangles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2. Related work

This section presents the related work on computing geodesic paths on discrete domains and discrete differential geometry. 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 to each other any longer.

Shortest geodesics have been extensively studied and also widely used in computer graphics community. Existing methods for computing exact shortest geodesic paths on polyhedral surfaces can be generally grouped into two categories, namely, the MMP algorithm and the CH algorithm. Both methods 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. The CH algorithm runs in $O(n^2)$ time and takes $O(n)$ space. Different extensions on the two algorithms have been developed, which aim at parallelization [14], performance improvement [15–18] and robustness [19], computing geodesic offsets [20,21], geodesic loops [22], and all-pairs geodesics [23].

Straightest geodesics receives relatively less attention than shortest geodesics. Polthier and Schmieß [24] 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 Schmieß also developed the geodesic flow method [25] to compute the evolution of the front of a point wave on a polyhedral surface. At each time step, the front is a topological circle on the surface [26], where each point moves a constant distance in orthogonal direction to the curve by the straightest geodesic path. Kumar et al. [27] 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 takes into account the variation of the tessellation normal along the geodesic path. Kumar et al.’s method can be considered as an extrinsic Euler method, which solves the geodesic equation with the first-order approximation. Therefore, their method tends to suffer from serious numerical issue and may deviate from the true geodesic curve after only a few iterations. Based on the fast marching method and the straightest geodesic [24], Martínez et al. [28] proposed an iterative algorithm for computing the shortest path between two fixed points.

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

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. [29] proved that convergence of the following properties were equivalent: surface normals, surface area, metric tensors, and Laplace–Beltrami operators. Dai et al. [30] 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. Lieutier and Thibert [31] proved that the convergence of *shortest* geodesics requires additional assumptions. The assumptions are 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.

Computing geodesics on parametric surfaces, such as Bézier, B-splines, and NURBS, has also been well studied. Beck et al. [32] computed geodesics on C^2 -continuous surfaces by solving four first-order ODEs. These equations were also used in [27] for computing geodesics on NURBS surfaces. Patrikalakis and Bardis [33] computed geodesic offsets of curves on rational B-splines using the initial-value integration. Maekawa [34] computed the geodesic path between a point and a curve on a parametric surface. See [35] for a detailed discussion. Chen et al. [36] solved the geodesic equation on rational Bézier surfaces with at least C^2 continuity, so that the Christoffel symbols were continuous across adjacent Bézier patches.

3. Preliminary

This section presents the background to differential geometry [37,38], which will be used in Section 4.

3.1. Geodesic equation

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 the locally distance minimizer, which can be obtained by minimizing the following functional

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

The Euler–Lagrange equation yields the geodesic equation

$$\frac{d^2 x^k}{dt^2} + \Gamma_{ij}^k \frac{dx^i}{dt} \frac{dx^j}{dt} = 0, \tag{1}$$

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. On smooth surfaces, a geodesic is both straightest and locally shortest. Furthermore, a geodesic is fully determined by an end point and a given tangent direction. See Fig. 3.

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

3.2. Geodesic curvature

Geodesics naturally generalize straight lines to curved surfaces. Let S be a regular surface and $\gamma(s) : [a, b] \rightarrow S$ be a curve on S parameterized by arc-length s . The curve γ passes through a point $p \in S$, with tangent vector $\mathbf{T} = \dot{\gamma}$, normal vector \mathbf{N} , and binormal

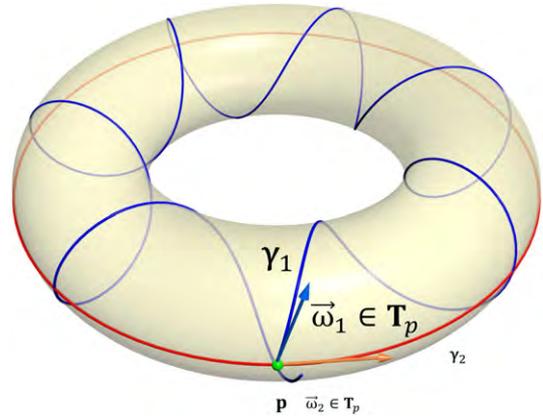


Fig. 3. On a smooth surface, there is a uniquely determined geodesic passing through a given point \mathbf{p} in an arbitrary tangent direction $\omega \in T_p$. Here we illustrate two geodesics γ_1 and γ_2 on a torus, corresponding to two tangent directions ω_1 and ω_2 . Note that the curve γ_2 can return to the starting point \mathbf{p} , whereas γ_1 cannot.

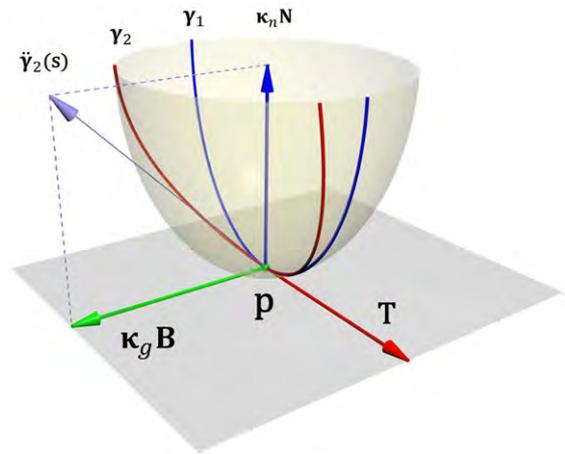


Fig. 4. Geodesic curvature. Let \mathbf{p} be a point on a smooth surface S , and \mathbf{N} the normal at \mathbf{p} . Two curves γ_1 and γ_2 are tangential at \mathbf{p} , and both their tangent direction is \mathbf{T} . The tangent \mathbf{T} , the normal \mathbf{N} , and the binormal \mathbf{B} define an orthonormal coordinate basis at \mathbf{p} . The curve γ_1 is a geodesic at \mathbf{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.

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 Fig. 4. 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, but does not curve *within* the surface.

4. Our algorithm

4.1. 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 using the weighted-average method with 1-ring neighborhood and Voronoi area weights [39,40]. The tangent plane of vertex v , denoted by T_v , passes through v and is perpendicular

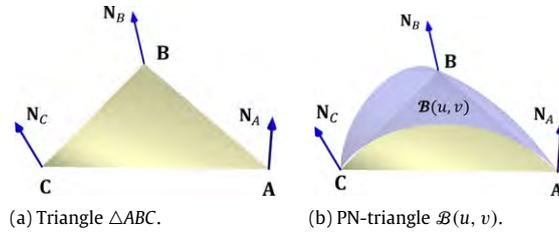


Fig. 5. A curved PN-triangle is a three-sided cubic Bézier patch that interpolates both the positions and normals of the three vertices of a planar triangle.

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 converge to those on S .

Given an arbitrary point $\mathbf{p} \in V$, and a tangent direction $\mathbf{w} \in T_{\mathbf{p}}$, our goal is to find a *unique* geodesic path which passes through \mathbf{p} in the direction \mathbf{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 \hat{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 [41] $\hat{f} \in \hat{M}$, which is a three-sided cubic Bézier patch corresponding to a triangle $f \in F$ of the underlying mesh M . The curved PN-triangle \hat{f} interpolates both the position and normal of f 's three corners. See Fig. 5. By using rational blending [42], adjacent Bézier patches can be glued so that they have G^1 continuity along the common edge. See Fig. 7. Note that other methods for locally construct piecewise smooth surfaces (e.g., [43]) can also be adopted.

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

4.2. Algorithm

Let \hat{f} be the curved patch containing \mathbf{p} . We compute a unique geodesic path $\hat{\gamma} \in \hat{f}$ using the fourth-order Runge–Kutta method and project $\hat{\gamma}$ onto the underlying triangle face f . Thanks to the G^1 continuity across the boundary, we can transfer the tangent direction to the neighboring patch, when $\hat{\gamma}$ crosses the boundary of \hat{f} . This allows us to extend $\hat{\gamma}$ to its neighboring patch, as shown in Fig. 7. We continue the procedure until the geodesic curve reaches the boundary or its length exceeds the user-specified threshold.

To compute the geodesic curve $\hat{\gamma}$ on \hat{M} numerically, our algorithm takes a step length δ as input and solves the initial value problem in an iterative manner. Each iteration contains the following four steps, as illustrated in Fig. 6:

- (i) The point $\hat{\mathbf{p}}_i$, the normal $\mathbf{N}_{\hat{\mathbf{p}}_i}$, and a tangent vector \mathbf{T}_i define a sectional plane π . Find a point $\hat{\mathbf{p}}_{i+1} \in \pi \cap \hat{M}$ such that $d(\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_{i+1}) = \delta$, as shown in Fig. 6(a).
- (ii) Let \hat{f} be the PNG1-triangle containing $\hat{\mathbf{p}}_{i+1}$ and $f \in F$ the underlying triangle. Project $\hat{\mathbf{p}}_{i+1}$ 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 $\mathbf{p}, \mathbf{p}_{i+1}$ and the corresponding edge of f . See Fig. 6(b).
- (iii) Compute the tangent \mathbf{T}'_{i+1} using the fourth-order Runge–Kutta method. See Section 4.3 and Fig. 6(c).
- (iv) Parallel transport the tangent \mathbf{T}'_{i+1} along $\hat{\gamma}$ to $\hat{\mathbf{p}}_{i+1}$. See Section 4.4 and Fig. 6(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.

Algorithm 1 Solving the Initial Value Problem of Discrete Geodesics

Input: 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
Output: A 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 is no more than L

- 1: Construct the 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: $\mathbf{T}_{i+1} = \text{parallel_transport}(\mathbf{T}'_{i+1}, \hat{\mathbf{p}}_i, \hat{\mathbf{p}}_{i+1})$;
 - 8: $s_{i+1} = s_i + \delta$;
 - 9: **end for**
-

4.3. 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 surface S , i.e.,

$$\begin{cases} \frac{d\mathbf{T}}{ds} = \kappa_n \mathbf{N}, \\ \frac{d\mathbf{N}}{ds} = -\kappa_n \mathbf{T} + \tau \mathbf{B}, \\ \frac{d\mathbf{B}}{ds} = \tau \mathbf{N}, \end{cases} \quad (2)$$

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} :

$$\frac{d\mathbf{T}}{ds} = -\left(\frac{d\mathbf{N}}{ds} \bullet \mathbf{T}\right) \mathbf{N}. \quad (3)$$

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

$$\mathbf{T}'_{i+1} = \mathbf{T}_i + \frac{\delta}{6}(\mathbf{v}_1 + 2\mathbf{v}_2 + 2\mathbf{v}_3 + \mathbf{v}_4), \quad (4)$$

$$s_{i+1} = s_i + \delta, \quad (5)$$

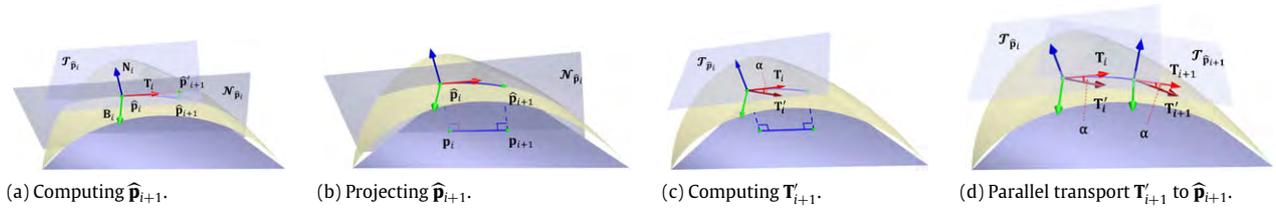


Fig. 6. A typical iteration in our algorithm.

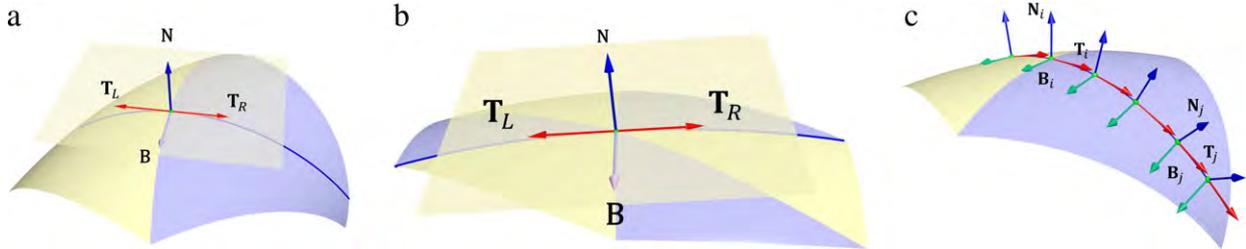


Fig. 7. G^1 continuity of a curved PNG1-triangle. Along the boundary of two neighbor patches, the patch is obtained by blending the two cubic Bézier patches for the triangle edge (see (a)). For patches that meet at a vertex v , the control points corresponding to the vertex are all projected to the tangent 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)).

where

$$\mathbf{v}_1 = f(s_i, \mathbf{T}_i), \quad (6)$$

$$\mathbf{v}_2 = f\left(s_i + \frac{\delta}{2}, \mathbf{T}_i + \frac{\delta}{2}\mathbf{v}_1\right), \quad (7)$$

$$\mathbf{v}_3 = f\left(s_i + \frac{\delta}{2}, \mathbf{T}_i + \frac{\delta}{2}\mathbf{v}_2\right), \quad (8)$$

$$\mathbf{v}_4 = f(s_i + \delta, \mathbf{T}_i + \delta\mathbf{v}_3). \quad (9)$$

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

4.4. 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 $\mathcal{T}_{\hat{\mathbf{p}}_{i+1}}$, 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 $\mathbf{T}_{i+1} = \mathbf{T} + \cos \alpha \mathbf{B}$. Fig. 6(d) illustrates the parallel transport of tangent vector.

5. Experimental results

We implemented our algorithm in C++ and evaluated it on a PC with an Intel Core2 2.83 GHz CPU and 4 GB main memory. To make the results *unitless*, we uniformly scale all models to be bounded by a unit cube. We measured the running time of computing 60 geodesic paths, which evenly emanate from 100 randomly selected source points and have 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 also affects the running time slightly, since the point location is triangulation dependent. See Fig. 8.

Fig. 9 shows the all-direction geodesic paths on several common 3D models in graphics community. 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

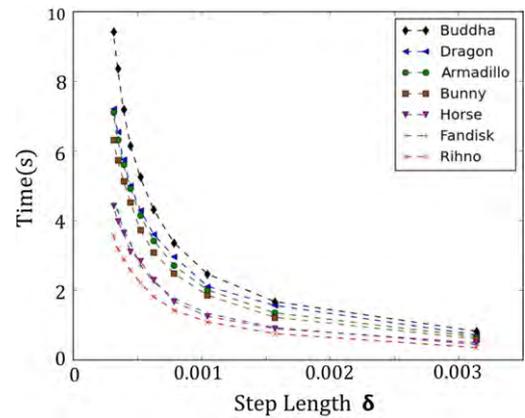


Fig. 8. The time complexity is inversely proportional to the step length δ . The vertical axis shows the running time of computing 60 geodesic paths of length $\sqrt{3}$ at 100 random source points. The time for constructing the PNG1-triangles, which is dependent on mesh complexity, was excluded.

method is numerically stable and works well on meshes with large and skinny triangles as well as sharp features. See Figs. 2 and 12. Fig. 11 demonstrates the results of our algorithm for computing the local exponential map on triangle meshes, which naturally define polar coordinates on 3D surfaces. Our method can be also applied to non-orientable surfaces, as shown in Fig. 10.

Our method computes the discrete geodesic path γ by projecting the unique geodesic $\hat{\gamma}$, which is defined on the piecewise smooth surface \hat{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. We construct five synthetic surfaces, namely, sphere, ellipsoid, paraboloid, hyperboloid, and torus, with increasing resolutions. See Fig. 13. All mesh vertices are on the smooth surfaces, and we compute the vertex normals analytically. Fig. 13(a) shows that when the mesh resolution is sufficiently high, the curved PNG1-triangles converge to the smooth surfaces. For each surface, we solved the initial value problem at 10 points and assigned 60 evenly distributed tangent directions for each point. Then we computed the length difference $\Delta L = |\text{len}(\gamma) - \text{len}(\hat{\gamma})|$ and the area of the gap between γ and $\hat{\gamma}$. As Fig. 13(b) (c) shows,

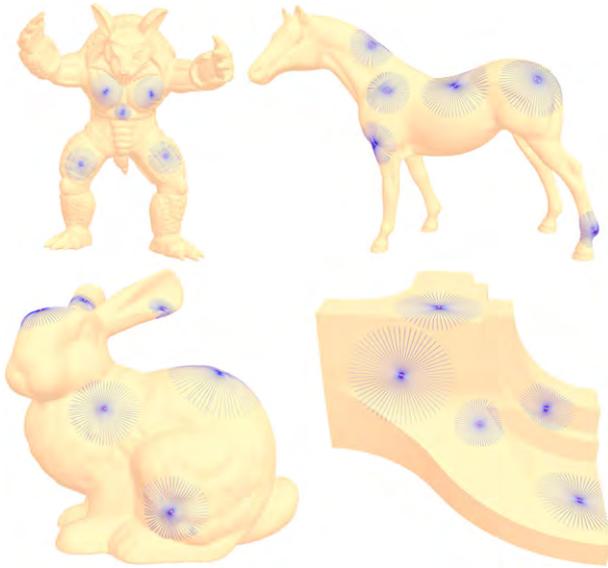


Fig. 9. We solved the initial value problem on a wide range of 3D models, and we assigned each source point 60 evenly distributed tangent directions. Note that our method works well on highly curved regions as well as sharp features. The images are generated at high resolution to allow for zoom-in examination.

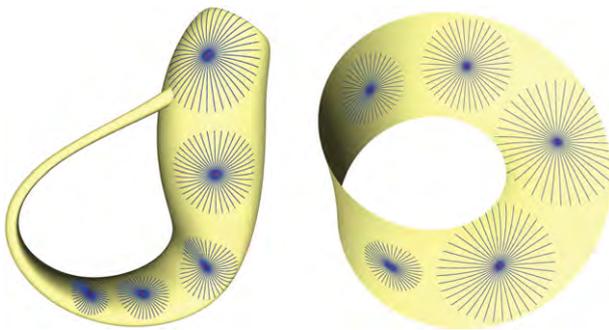


Fig. 10. Our method works for the non-orientable surfaces.

both the gap area and length difference converge to zero as the average edge length h decreases.

6. Discussion

6.1. 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 [33,27,36,34]. These methods directly solve the standard geodesic equation (1), which is a second-order ODE of the 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. In contrast, 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 Eq. (3) shows, the derivative of tangent depends on the change of normal, which can be computed efficiently on cubic Bézier patches.

6.2. Comparison with straightest geodesic

Polthier and Schmieß [24] defined the straightest geodesic as a curve that bisects the total vertex angle.

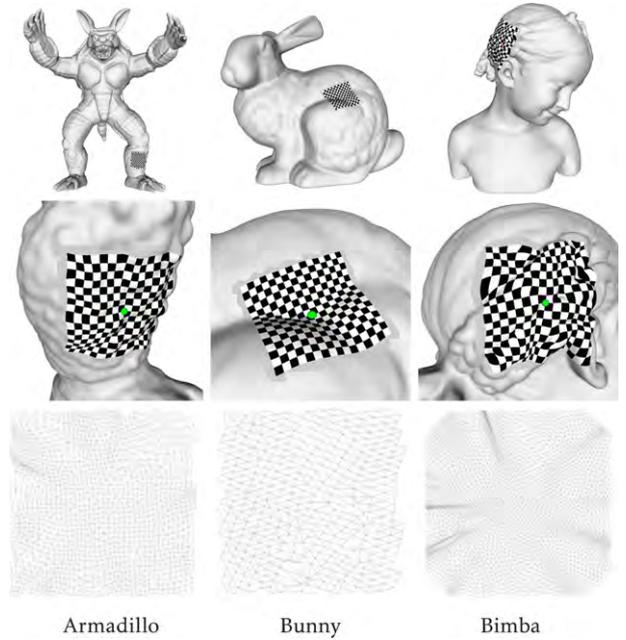


Fig. 11. Computing local parameterization using our method. The user specified a point \mathbf{p} on the 3D mesh and a radius r . We then applied our method to compute 60 geodesic paths passing through \mathbf{p} and having length r . Each geodesic path uniquely corresponds to a radial line of a polar coordinate system defined on $T_{\mathbf{p}}$, which allows us to build a polar coordinate system on 3D meshes. We visualized the parameterization by using the checkerboard texture mapping on a rectangular patch. The bottom row shows the parameterized patch on \mathbb{R}^2 .

Definition (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 Fig. 14(a).

A discrete geodesic γ containing no surface vertex 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 Fig. 14(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 Fig. 14(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 (see Fig. 15).
- The shortest geodesic distances converge to geodesic distances on smooth surfaces at a quadratic rate [11], whereas the straightest geodesics do not converge [31].

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

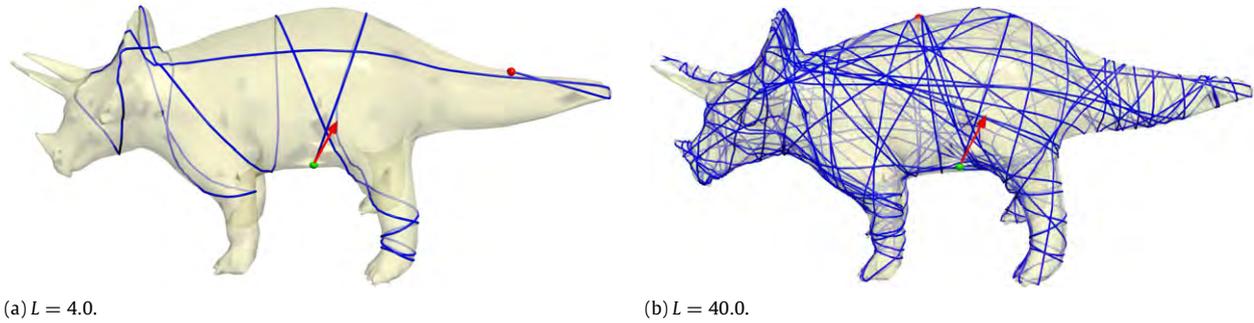


Fig. 12. Solving the initial value problem on the Rhino model with two different length constraints. The source point is located at the belly button.

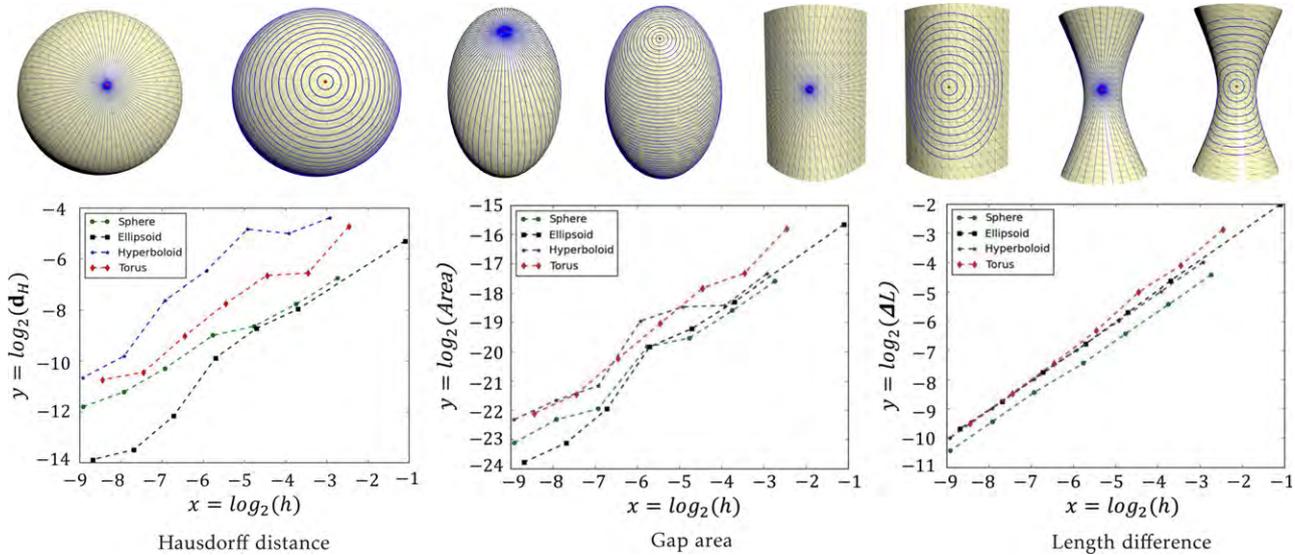


Fig. 13. Convergence test on four synthetic models. Sphere $(\cos u \sin v, \sin u \sin v, \cos v)$, $u \in [0, 2\pi]$, $v \in [0, \pi]$; Ellipsoid $(2 \cos u \sin v, 2 \sin u \sin v, 3 \cos v)$, $u \in [0, 2\pi]$, $v \in [0, \pi]$; Paraboloid $(2v^2, 4v, 6u)$, $u, v \in [-1, 1]$; Hyperboloid $(2 \cos v \sqrt{1+u^2}, 2 \sin v \sqrt{1+u^2}, 4u)$, $u \in [-1, 1]$, $v \in [0, 2\pi]$; 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 \bar{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 $\gamma \in \bar{M}$ converges to the geodesic $\hat{\gamma} \in \bar{M}$, since both the length difference $|\text{len}(\gamma) - \text{len}(\hat{\gamma})|$ and the area of the gap between γ and $\hat{\gamma}$ approach zero. The horizontal axis h is the average edge length of M .

to a triangle, must be *straight line*. We observe that straightest geodesics are too “rigid” to approximate analytic geodesics on low-resolution meshes. As Fig. 16 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. 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.

6.3. 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 [44], we adopted the following heuristic to determine the step length in our implementation. If

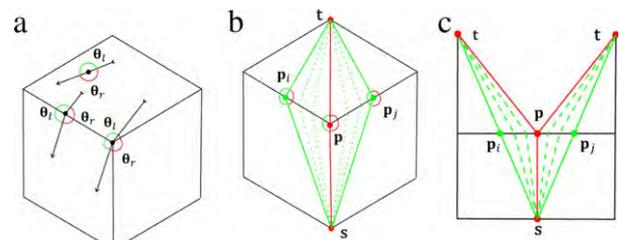


Fig. 14. (a) A straightest geodesic has equal left and right curve angles $\theta_l = \theta_r$. (b) The shortest geodesic between s and t cannot be extended through a spherical vertex p , since making it off p shortens the length. (c) Unfold the spherical vertex p to \mathbb{R}^2 .

the length of dN is less than a user-specified tolerance tol , we simply computed the locally straightest geodesic for this iteration. Otherwise, we adjusted the tangent direction with an adaptive step length δ_a

$$\delta_a = 0.9 \times \delta \times \frac{tol}{\|dN\|}. \tag{10}$$

Empirically, we set the local error tolerance $tol = \frac{h}{2^6}$. Table 1 shows the performance and quality measurements of our algorithm with fixed step length and adaptive step length.

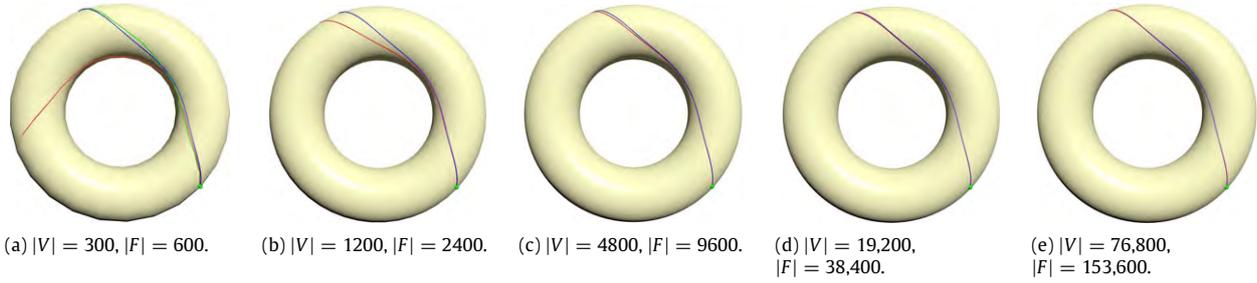


Fig. 15. The straightest geodesic [24] (red) converges to the ground truth (blue) only when the mesh resolution is sufficiently high. Our method with step length $\delta = \frac{h}{16}$ produces very good result (green) even when the mesh resolution is very low. Note that all geodesics are given the same initial condition. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

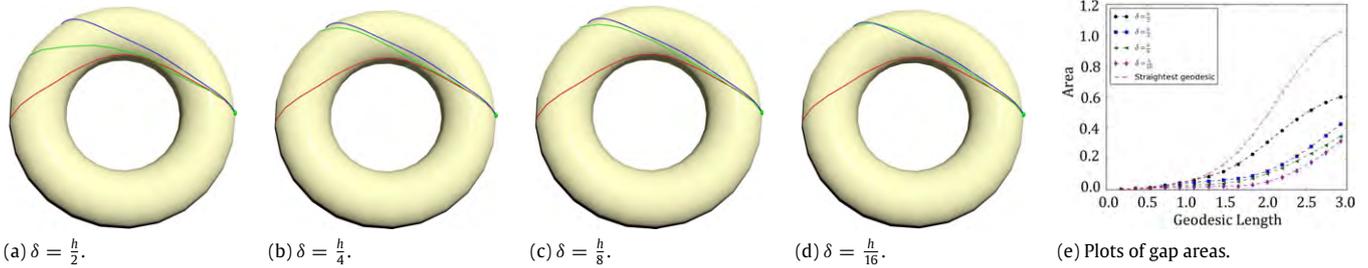


Fig. 16. Comparison of our method (green) to straightest geodesic (red) and analytic geodesic (blue) on the torus model with 600 faces. Note that all geodesics are with the same initial condition. We set the step length $\delta = \frac{h}{2}, \frac{h}{4}, \frac{h}{8}$ and $\frac{h}{16}$, respectively, 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). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1

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

Surface	(V , F)	h	Fixed step length (ΔL , t)				Adaptive step length (ΔL , 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)	(0.002584, 0.277009)
Ellipsoid	(382, 760)	0.467991	(0.040078, 0.161148)	(0.019068, 0.260773)	(0.018068, 0.485002)	(0.004623, 1.005054)	(0.004917, 0.244246)
Hyperboloid	(465, 900)	0.132355	(0.031381, 0.229931)	(0.015646, 0.258773)	(0.007801, 0.460958)	(0.003913, 0.958326)	(0.005089, 0.248684)
Torus	(300, 600)	0.182588	(0.057618, 0.139749)	(0.030948, 0.206672)	(0.012446, 0.425581)	(0.005751, 0.888576)	(0.006547, 0.213495)

7. Conclusion

This paper presents a method to solve the initial value problem of discrete geodesics. Since the commonly used shortest geodesic [8] does not produce a unique solution, we solve the problem by computing a geodesic curve on a piecewise smooth surface \tilde{M} , which well approximates M and can be constructed at little cost. Each patch in \tilde{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 the geodesic curve reaches the boundary of the current patch, its tangent can be transported to the neighboring patch, thanks to the G^1 -continuity along the common boundary of two adjacent patches. We project the geodesic curve from \tilde{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.

Acknowledgments

This project was partially supported by the National Research Foundation, Prime Minister’s Office, Singapore under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office. P. Cheng is supported by the Ph.D. scholarship from the Institute for Media Innovation, NTU. Y.-J. Liu is partially supported by the Natural Science Foundation of China (61432003, 61322206) and the TNList Cross-discipline Foundation. C. Tu is partially supported by the Natural Science Foundation of China 61332015 and 61272242. Y. He is partially supported by RG40/12 and MOE2013-T2-2-011.

References

- [1] Sánchez-Reyes J, Dorado R. Constrained design of polynomial surfaces from geodesic curves. *Comput-Aided Des* 2008;40(1):49–55.
- [2] Dorst L, Mandhyan I, Trovato K. The geometrical representation of path planning problems. *Robot Auton Syst* 1991;7(2–3):181–95.
- [3] Sloan P-PJ, Rose III CF, Cohen MF. Shape by example. In: *Proceedings of the 2001 symposium on interactive 3D graphics*. ACM; 2001. p. 135–43.
- [4] Peyré G, Cohen L. Geodesic computations for fast and accurate surface remeshing and parameterization. In: *Elliptic and parabolic problems*. Springer; 2005. p. 157–71.
- [5] Floater MS, Hormann K, Reimers M. Parameterization of manifold triangulations. In: *Approximation theory X: abstract and classical analysis*. Nashville: Vanderbilt University Press; 2002. p. 197–209.

- [6] Koschan A. Perception-based 3D triangle mesh segmentation using fast marching watersheds. In: Proceedings of CVPR'03, Vol. 2, 2003. p. II–27.
- [7] Liu Y-J. Semi-continuity of skeletons in 2-manifold and discrete Voronoi approximation. *IEEE Trans Pattern Anal Mach Intell* 2015; <http://dx.doi.org/10.1109/TPAMI.2015.2430342>.
- [8] Mitchell JS, Mount DM, Papadimitriou CH. The discrete geodesic problem. *SIAM J Comput* 1987;16(4):647–68.
- [9] Chen J, Han Y. Shortest paths on a polyhedron. In: Proceedings of the sixth annual symposium on computational geometry. ACM; 1990. p. 360–9.
- [10] Kimmel R, Sethian JA. Computing geodesic paths on manifolds. *Proc Natl Acad Sci* 1998;95(15):8431–5.
- [11] Crane K, Weischedel C, Wardetzky M. Geodesics in heat: a new approach to computing distance based on heat flow. *ACM Trans Graph* 2013;32(5):152.
- [12] Campen M, Heistermann M, Kobbelt L. Practical anisotropic geodesy. In: Computer graphics forum. Vol. 32. Wiley Online Library; 2013. p. 63–71.
- [13] Ying X, Wang X, He Y. Saddle vertex graph (SVG): a novel solution to the discrete geodesic problem. *ACM Trans Graph* 2013;32(6):170.
- [14] Ying X, Xin S-Q, He Y. Parallel Chen–Han (PCH) algorithm for discrete geodesics. *ACM Trans Graph* 2014;33(1):9.
- [15] Xu C-X, Wang TY, Liu Y-J, Liu L, He Y. Fast wavefront propagation (FWP) for computing exact discrete geodesics on meshes. *IEEE Trans Vis Comput Graphics* 2015;21(7):822–34.
- [16] Xin S-Q, Wang G-J. Improving Chen and Han's algorithm on the discrete geodesic problem. *ACM Trans. Graphics* 2009;28(4):104.
- [17] Surazhsky V, Surazhsky T, Kirsanov D, Gortler SJ, Hoppe H. Fast exact and approximate geodesics on meshes. In: *ACM Transactions on Graphics*, Vol. 24, 2005, p. 553–60.
- [18] Liu Y-J. Exact geodesic metric in 2-manifold triangle meshes using edge-based data structures. *Comput-Aided Des* 2013;45(3):695–704.
- [19] Liu Y-J, Zhou Q-Y, Hu S-M. Handling degenerate cases in exact geodesic computation on triangle meshes. *Vis Comput* 2007;23(9–11):661–8.
- [20] Bommes D, Kobbelt L. Accurate computation of geodesic distance fields for polygonal curves on triangle meshes. In: Proceedings of Vision Modeling and Visualization. 2007, p. 151–60.
- [21] Xin S-Q, Ying X, He Y. Efficiently computing geodesic offsets on triangle meshes by the extended Xin–Wang algorithm. *Comput-Aided Des* 2011;43(11):1468–76.
- [22] Xin S-Q, He Y, Fu C-W. Efficiently computing exact geodesic loops within finite steps. *IEEE Trans Vis Comput Graphics* 2012;18(6):879–89.
- [23] Xin S-Q, Ying X, He Y. Constant-time all-pairs geodesic distance query on triangle meshes. In: Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games, 2012, p. 31–8.
- [24] Polthier K, Schmies M. Straightest geodesics on polyhedral surfaces. *ACM*; 2006.
- [25] Polthier K, Schmies M. Geodesic flow on polyhedral surfaces. In: *Data visualization*. Springer; 1999. p. 179–88.
- [26] Liu Y-J, Chen Z, Tang K. Construction of iso-contours, bisectors, and Voronoi diagrams on triangulated surfaces. *IEEE Trans Pattern Anal Mach Intell* 2011;33(8):1502–17.
- [27] Ravi Kumar G, Srinivasan P, Devaraja Holla V, Shastry K, Prakash B. Geodesic curve computations on surfaces. *Comput Aided Geom Design* 2003;20(2):119–33.
- [28] Martínez D, Velho L, Carvalho PC. Computing geodesics on triangular meshes. *Comput Graph* 2005;29(5):667–75.
- [29] Hildebrandt K, Polthier K, Wardetzky M. On the convergence of metric and geometric properties of polyhedral surfaces. *Geom. Dedicata* 2006;123(1):89–112.
- [30] Dai J, Luo W, Jin M, Zeng W, He Y, Yau S-T, et al. Geometric accuracy analysis for discrete surface approximation. *Comput Aided Geom Design* 2007;24(6):323–38.
- [31] Lieutier A, Thibert B. Convergence of geodesics on triangulations. *Comput Aided Geom Design* 2009;26(4):412–24.
- [32] Beck JM, Farouki RT, Hinds JK. Surface analysis methods. *IEEE Comput Graph Appl* 1986;6(12):18–36.
- [33] Patrikalakis N, Bardis L. Offsets of curves on rational B-spline surfaces. *Eng Comput* 1989;5(1):39–46.
- [34] Maekawa T. Computation of shortest paths on free-form parametric surfaces. *J Mech Des* 1996;118(4):499–508.
- [35] Patrikalakis NM, Maekawa T. Shape interrogation for computer aided design and manufacturing. Springer Science & Business Media; 2009.
- [36] Chen X, He S-Y, Yu D-F, Yin H-C, Hu W-D, Zhu G-Q. Geodesic computation on NURBS surfaces for UTD analysis. *IEEE Antennas Wirel Propag Lett* 2013;12:194–7.
- [37] do Carmo MP. *Differential geometry of curves and surfaces*. Pearson; 1976.
- [38] Spivak M. *A comprehensive introduction to differential geometry*, vol. 2. I, Boston, Mass., 1970.
- [39] Meyer M, Desbrun M, Schröder P, Barr AH. Discrete differential-geometry operators for triangulated 2-manifolds. In: *Visualization and mathematics III*. Springer; 2003. p. 35–57.
- [40] Rusinkiewicz S. Estimating curvatures and their derivatives on triangle meshes. In: Proceedings of symposium on 3D data processing, visualization and transmission. IEEE; 2004. p. 486–93.
- [41] Vlachos A, Peters J, Boyd C, Mitchell JL. Curved PN triangles. In: Proceedings of the 2001 symposium on interactive 3D graphics, 2001, p. 159–66.
- [42] Fünfzig C, Müller K, Hansford D, Farin G. PNG1 triangles for tangent plane continuous surfaces on the GPU. In: Proceedings of graphics interface. GI'08, 2008, p. 219–26.
- [43] Leung Y-S, Wang CC, Zhang Y. Localized construction of curved surfaces from polygon meshes: A simple and practical approach on GPU. *Comput-Aided Des* 2011;43(6):573–85.
- [44] Press WH. *Numerical recipes 3rd edition: the art of scientific computing*. Cambridge university press; 2007.