Some properties of exact geodesics on triangular mesh surfaces

Yong-Jin Liu[1], Wen-Qi Zhang
Department of Computer Science and Technology,
Tsinghua University,
Beijing, China
[1]Corresponding author: liuyongjin@tsinghua.edu.cn

Kai Tang
Department of Mechanical Engineering,
The Hong Kong University of Science and Technology,
Hong Kong

Abstract

Computing exact geodesic paths on triangular mesh surfaces is an important operation in many geometric algorithms. In this paper we show several particular examples that reveal four interesting properties and three observations of exact geodesic computations on triangular mesh surfaces. The properties are proved and observations are drawn from experimental results using practical models. With these properties and observations, we present a new perspective of the classical MMP algorithm and the readers may find a clearer picture of how to code the MMP algorithm.

Keywords

Exact geodesic computation; triangular mesh surfaces

1. Introduction

In many geometric problems in industry, solution spaces (such as object spaces or configuration spaces) usually are given in the form of three-dimensional (3D) polygonal surfaces. On these 3D surfaces, the distance metric is built on geodesics. Computing the geodesic paths is a fundamental tool in many practical applications, including robot motion planning [1, 2], shape analysis and classification [3, 4] and shape tracking in vision [5, 6, 7].

Many works have been reported for computing approximate geodesics on triangular mesh surfaces, leading to both numerical algorithms [8, 9] and computational geometry algorithms [10, 11]. A comprehensive survey on approximate geodesic computation is presented in [12]. Exact geodesic algorithm was first studied by Mitchell et al. [13] in which an $O(n^2 \log n)$ algorithm is proposed. Later, several authors improved this bound to $O(n^2)$ [14, 15] and $O(n \log^2 n)$ [16]. Surazhsky et al. [17] presented a novel implementation of the MMP algorithm of [13] and showed that, in practice, it ran much faster than the other algorithms.

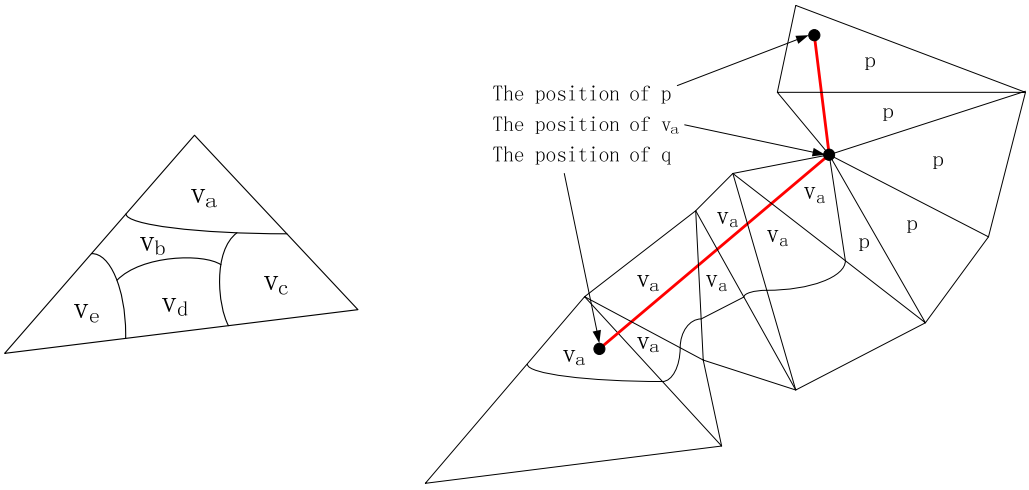
In this note, we illustrate with real examples of geodesic computation and prove four interesting properties. Three observations are also concluded from experimental results. Supported by these properties and observations, a new perspective of the MMP algorithm is presented, with which the user may find a clearer picture of how to code the MMP algorithm.

2. A new perspective of the MMP algorithm

The triangular mesh surfaces studied in this paper are compact piecewise flat surfaces. The Hopf-Rinow theorem ensures that a minimal geodesic exists for arbitrary two points on this kind of surfaces [18]. Denote the topology of a triangular mesh surface T by (V, E, F), where V, E, F are the vertex, edge and face sets, respectively. Given a single source point p on the surface T, the MMP algorithm computes a subdivision structure $\{D_1, D_2, \cdots, D_n\}$ on T that satisfies $\bigcup_{i=1}^{n} D_i = \bigcup_k F_k$, $F_k \in F$, and $D_i \cap D_j = \emptyset$, $i \neq j$, $i, j = 1,2, \cdots, n$, where n is the number of subdivision cells. Refer to Fig. 1(a). Each subdivision D_i has a corresponding pseudo-source v_i. v_i is unfolded and stored in a 2D coordinate on the local plane defined by D_i. Given an arbitrary target position q on T, the geodesic path between p and q is computed as follows.

(1) Find the subdivision D_q containing q. Set $D_l = D_q$, $r = q$.
(2) Connect r and the 2D position of v_l by a line l, in the plane defined by D_l.
(3) If $v_l \neq p$, find the intersection x of the ray l with the boundary of D_l; Otherwise stop.
(4) Find the adjacent subdivision D_j of D_l along the intersection x. Set $D_l = D_j$, $r = x$. Go back to (2).

The procedure is sketched in Fig. 1(b). The illustrative subdivision structure in Fig. 1 is from a real example as Example 4 in Section 3.



(a) The subdivision of a triangle with the corresponding pseudo-vertices

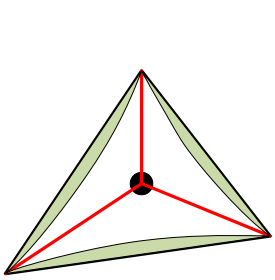
(b) Tracing the geodesic path using the subdivision and the pseudo-sources

Figure 1. The subdivision structure resulted from the MMP algorithm and the associated geodesic computation. A real example is presented in Example 4 and is shown in Fig. 15.

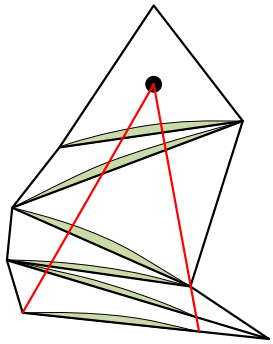
Due to usually the extreme complexity of the 2D subdivision structure with curved boundaries on T, it is impractical to build and store this 2D structure. Mitchell et al. [13] circumvented this difficulty by only building a 1D subdivision structure on the edge set E explicitly. The MMP algorithm can be stated as follows.

Inside a triangle, the geodesic paths must be straight lines. When crossing a triangle edge e, a geodesic must also be a straight line if the previous triangle is unfolded along e into the plane

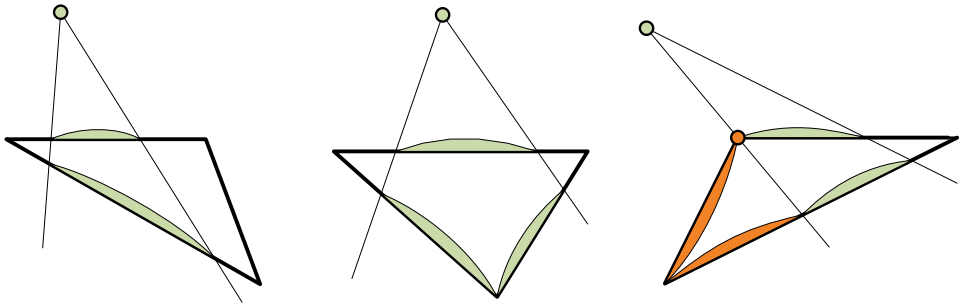
containing the next triangle. As shown in Fig. 2a, from the triangle containing the source *p*, a set of initial visibility wedges can be identified. Then these wedges are propagated until all the edges *E* in *T* are covered (ref. Fig. 2b). During the wedge propagation, three different cases as shown in Fig. 2c could arise. It is shown in [13] that the pseudo-source of each wedge can only be the concave vertices in *V*.



(a) The initial visibility wedges in the triangle containing the source *p*



(b) The visibility wedge propagation.



(c) Three cases in visibility wedge propagation. The colors of pseudo-sources are the same with the colors of the wedges.

Figure 2. Visibility wedge propagation

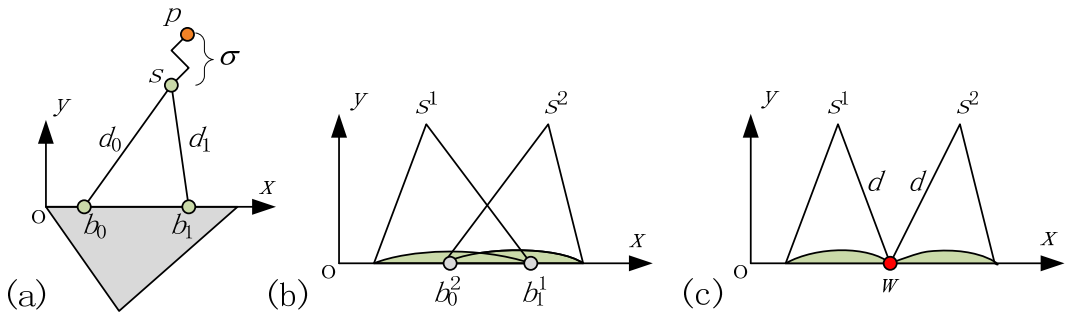


Figure 3. The encoding of the visibility wedge and the intersection of two wedges.

Surazhsky et al. [17] proposed a novel structure to store the wedge information in the local 2D plane, as follows.

Definition 1. Each wedge is encoded by a 6-tuple $(b_0, b_1, d_0, d_1, \sigma, \tau)$. Refer to Fig. 3a. The terms b_0, b_1 are parameters measuring distance along the edge. The 2D unfolded position of the pseudo-source *s* is encoded by its distances d_0, d_1 to the endpoints b_0, b_1. A binary direction τ is used to specify the side of edge on which the source lies. The term σ is the length of the geodesic path from *s* back to the source *p*.

In the following, we make no difference between the source and pseudo-sources if it is clear in context, and denote it by *s*. Refer to Figs. 3a and 3b. During the wedge propagation, the new resulting wedges may intersect some existing wedges. Surazhsky et al. [17] update the two intersected wedges $(b_0^i, b_1^i, d_0^i, d_1^i, \sigma^i, \tau^i)$, $i = 1, 2$, by solving an equation

$$\sqrt{(w_x - s_{1x})^2 + s_{1y}^2} + \sigma^1 = \sqrt{(w_x - s_{2x})^2 + s_{2y}^2} + \sigma^2 \tag{1}$$

The solution is the intersection points of a branch of hyperbola with the *x* axis. Equation (1) can be reduced to a quadratic equation:

$$Ap_x^2 + Bp_x + C = 0 \tag{2}$$

with

$$A = \alpha^2 - \beta^2, \qquad B = 2s_{2x}\beta^2 + \alpha\gamma, \qquad C = \frac{1}{4}\gamma^2 - \|s_2\|^2\beta^2$$

and

$$\alpha = s_{2x} - s_{1x}, \qquad \beta = \sigma^2 - \sigma^1, \qquad \gamma = \|s_1\|^2 - \|s_2\|^2 - \beta^2$$

Given the structure of 6-tuple $(b_0, b_1, d_0, d_1, \sigma, \tau)$, a wedge can be efficiently represented by an interval on an edge. Surazhsky et al. [17] use a priority queue to propagate visibility wedges in a continuous Dijkstra fashion. The queue is initialized with the wedges shown in Fig. 2(a). All wedges are sorted based on the value of minimal geodesic distance in the wedge. When a wedge is popped off the queue, it is propagated across a face. After the propagation as shown in Fig. 2c and the intersection testing as shown in Figs. 3b and 3c, new wedges may appear and get inserted into the queue. Mitchell et al. [13] prove that:

(a) The algorithm will generate correct solutions that all the edges *E* of *T* are completely covered by the intervals of the visibility wedges.

(b) The total number of intervals in *E* are bounded by $O(n^2)$, n is the number of edges in *E*.

In the following section, we prove four properties that may bring more clarity of the original MMP algorithm.

3. Some properties of exact geodesics in MMP algorithm

In this section, we present and prove four interesting properties for the exact geodesic in the MMP algorithm.

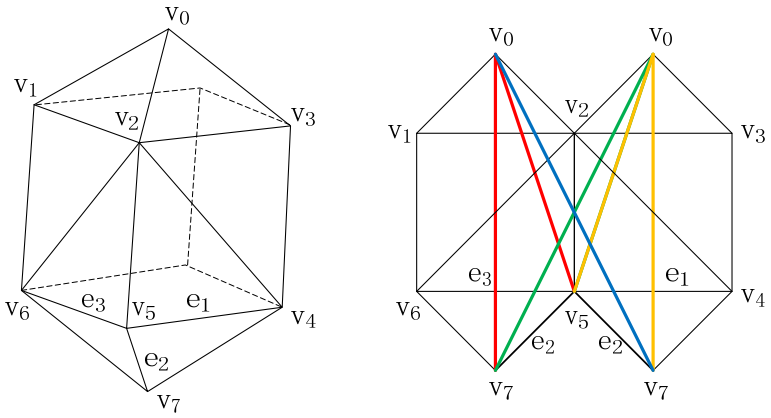


Figure 4. Example 1: on edge e_2, source v_0 contributes four visibility wedges w_1, w_2, w_3 and w_4, showing red, green, yellow and blue colors, respectively. w_1 goes through $\Delta v_0v_1v_2$, $\Delta v_1v_6v_2$, $\Delta v_2v_6v_5$, $\Delta v_5v_6v_7$. w_2 goes through $\Delta v_0v_2v_3$, $\Delta v_2v_4v_3$, $\Delta v_2v_5v_4$, $\Delta v_2v_6v_5$, $\Delta v_5v_6v_7$, w_3 and w_4 are symmetric counterparts of w_1 and w_2, respectively.

Example 1. Figure 4 shows a symmetric mesh model. The positions of the vertices are: $v_0(0.5,0.5,1.5)$, $v_1(0,0,1)$, $v_2(1,0,1)$, $v_3(1,1,1)$, $v_4(1,1,0)$, $v_5(1,0,0)$, $v_6(0,0,0)$ and $v_7(0.5,0.5,-0.5)$. The source point is at v_0. Let Δijk denote the triangle formed by vertices i,j,k. Consider edge $e_2(v_5,v_7)$. On e_2 source v_0 contributes four visibility wedges w_1, w_2, w_3 and w_4. w_1 goes through $\Delta v_0v_1v_2$, $\Delta v_1v_6v_2$, ,$\Delta v_2v_6v_5$, ,$\Delta v_5v_6v_7$. w_2 goes through $\Delta v_0v_2v_3$, $\Delta v_2v_4v_3$, $\Delta v_2v_5v_4$, $\Delta v_2v_6v_5$, $\Delta v_5v_6v_7$. w_3 is a symmetric counterpart of w_1 and goes through $\Delta v_0v_2v_3$, $\Delta v_2v_4v_3$, $\Delta v_2v_5v_4$, $\Delta v_4v_5v_7$. w_4 is a symmetric counterpart of w_2. It is readily seen that (a) the intersection of w_1 and w_2 has no solution inside the intersected range; (b) due to symmetry, for the intersection of w_1 and w_3, $\alpha=\beta=\gamma=0$, $A=B=C=0$ and the quadratic equation (2) has an infinite number of solutions inside the intersected range.

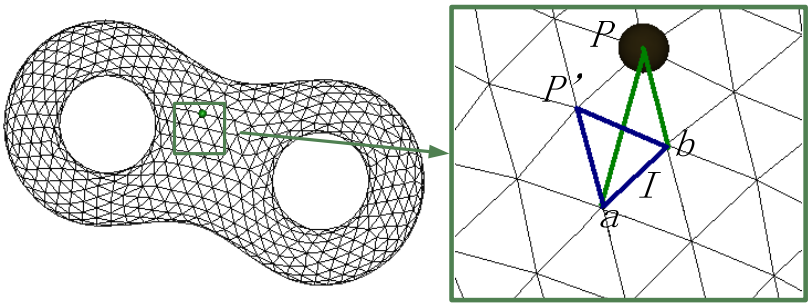


Figure 5. Example 2: the source point P and the pseudo-source P' contribute two visibility wedges on the edge $I(a,b)$, in blue and green colors, respectively.

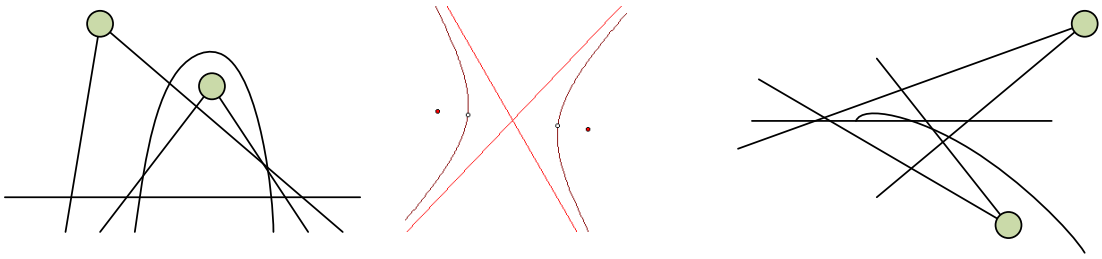
Example 2. Figure 5 shows an 8-shape model. In the zoomed view, the source point P and the pseudo-source P' contribute two visibility wedges on the edge $I(a,b)$, respectively. The geodesic distance from P' to P is $\|P-P'\|$. The vectors $P-P'$ and $b-a$ are parallel (in the

tolerance of floating point arithmetic). In this case, $A = B = 0,\ C \neq 0$, there is no solution in Equation (2).

For the number of solutions of Equation (2) in the shared range (b_0^2, b_1^1) of two intersected wedges w_1 and w_2 (ref. Fig. 3b), we have the following property:

Property 1. In the shared range (b_0^2, b_1^1) of two intersected wedges w_1 and w_2, the number of solutions in Equation (2) can be zero, one, two or infinite. If the number of solutions is two, the two (pseudo-) sources must be in the different sides of the edge.

Proof. The existence of one single solution is given in [17]. The existence of no solution is shown in Example 2. The existence of infinite solutions is shown in Example 1. There are two possibilities of existence of exactly two solutions in (b_0^2, b_1^1). The first one is that two pseudo-sources lie in the same side of the edge. The second is two sources lying in the different sides of the edge. Below we prove that the first case does not exist. First, by Property 2 (to be given next), one source cannot be in the wedge of another source. So if two sources lie in the same side, they can only be in the configuration shown in Fig. 6a. However, since the line connecting any two points in a branch of hyperbola lies between the two asymptotes of the hyperbola, this case cannot exist. So If the number of solutions is two in the range (b_0^2, b_1^1), the two sources must be in the different sides of the edge (cf. Fig. 6b). ■



(a) In the same side of the edge (b) In different sides of the edge

Figure 6. Proof of Property 1: relationships between the two sources of intersected wedges and the involved edge

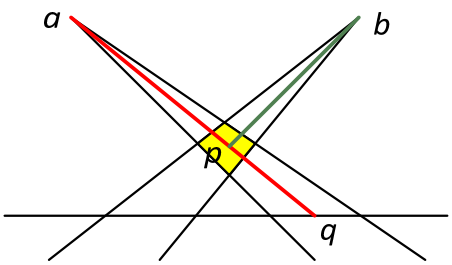


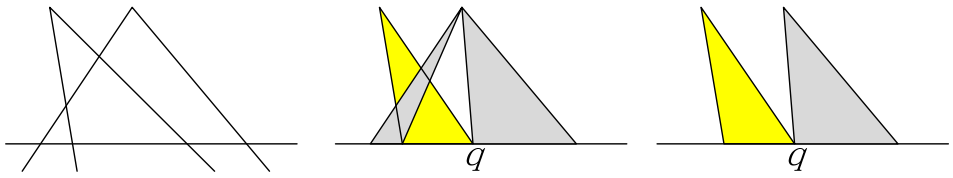
Figure 7. Proof of Property 2.

In the proof of Property 1, we use the following property:

Property 2. The updated visibility wedges cannot intersect.

Proof. Let a, b be two (pseudo-) sources of the wedges w_a and w_b, respectively. If w_a and w_b intersect, let p be a point in the intersected area as shown in Fig. 7. Denote the geodesic distance from source to p through a by $p(a)$, and let $p(b)$ be the one similarly defined for b. Without loss of generality, assume $p(a) > p(b)$ (no point in the intersection area could have equal geodesic distances to a and to b; otherwise it would violate the fact that the bisector is a branch of hyperbola). In the wedge w_a, extend the line from a to p to a point q at the edge. Then the inequality $p(a) + \overline{pq} > p(b) + \overline{pq}$ means $q(a)$ in w_a is not a geodesic distance, a contradiction. ■

Using Property 2, a frequently appeared configuration of wedge intersection in practical models, as shown in Fig. 8, can be correctly characterized. In Figure 8(a), the two wedges intersect each other and, at the edge, the range of one wedge completes cover that of the other range. Fig. 8(b) is an incorrectly updated result according to Property 2; Fig. 8(c) gives the correct update.



(a) (b) (c)

Figure 8. (a) Two intersected wedges to be updated: q is the solution of equation (2) inside the intersected range. Given Property 2, (b) is an incorrect updated result and (c) is correct.

Refer to Fig. 6b again. A natural question is: can two visibility wedges meet at an edge from its different sides? This is equivalent to asking, is it necessary to set the Boolean value τ in the 6-tuple $(b_0, b_1, d_0, d_1, \sigma, \tau)$ of a wedge? Our answer is yes and we use the following example to verify our claim.

Example 3. Refer to the mesh model shown in Figure 9. The positions of vertices are $v_0(0.5,0.5,1)$, $v_1(0,0,1)$, $v_2(1,0,1)$, $v_3(1,1,1)$, $v_4(0,1,1)$, $v_5(0,0,0)$, $v_6(1,0,0)$, $v_7(1,1,0)$, $v_8(0,1,0)$. The source point is at v_0. Consider edge $e_0(v_5, v_7)$. Source v_0 contributes two wedges w_1, w_2 on e_0. w_1 goes through $\Delta v_0 v_1 v_2$, $\Delta v_1 v_6 v_2$, $\Delta v_1 v_5 v_6$, $\Delta v_5 v_7 v_6$. w_2 goes through $\Delta v_0 v_3 v_4$, $\Delta v_3 v_7 v_4$, $\Delta v_4 v_7 v_8$, $\Delta v_5 v_8 v_7$. The intersection of w_1 and w_2 has one solution in the intersected range and the final updated wedges, lying in different sides of edge e_0, are shown rightmost in Fig. 9.

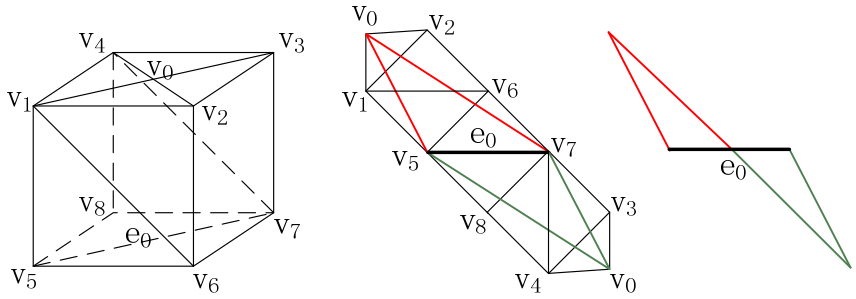


Figure 9. The model used in Example 3.

Property 3. The final visibility wedges at an edge may come from different sides of the edge. Furthermore, the visibility wedges may be sorted at an edge in an interlaced manner, i.e., odd wedges from one side and even wedges from the other side of the edge.

Proof. To see the interlaced wedges from different sides of the edge, consider the case shown in Fig. 6b. Figure 10(a) redraws the configuration that the intersection of two wedges has two solutions in the intersected range. The final updated wedges are shown in Fig. 10(b). If the wedges at the edge are paired as in the Fig. 10(a) configuration, the final wedges appear in an interlaced manner. ■

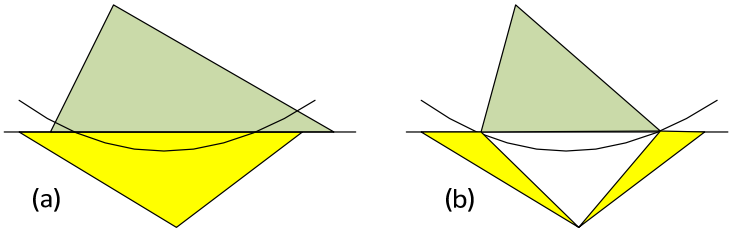


Figure 10. Proof of Property 3.

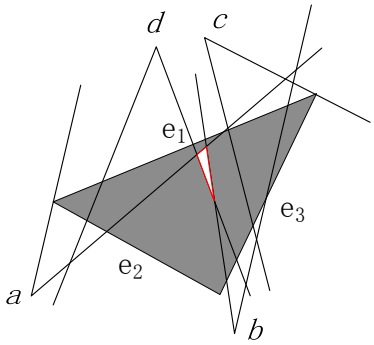


Figure 11. A paradox of 2D face subdivision from 1D edge subdivision.

As stated in Section 2, the MMP algorithm of [13] can be interpreted as building a 1D subdivision on the edge set E of the mesh T; however, the geodesic computation needs a 2D subdivision on the face set F of T as shown in Fig. 1a. Since the wedges can reach an edge from its different sides, one may ask whether the 1D subdivision on the edges can induce a correct 2D

subdivision in the faces. Figure 11 illustrates such a puzzle. Let w_a, w_b, w_c, w_d denote the wedges with pseudo-sources a, b, c, d, respectively. For 1D subdivision on the edges, the edge e_1 is completely covered by wedges w_a, w_b, w_c, w_d, e_2 by w_a, w_d and e_3 by w_b, w_c, w_d. For the induced 2D subdivision inside the triangle by w_a, w_b, w_c, w_d, there is a vacuum area (shown in white color in Fig. 11) which is not covered by any wedges. With the following property, we prove that this case cannot happen. Furthermore, a real example is presented in Example 4 that demonstrates the illustrative subdivision structure shown in Fig. 1a.

Property 4. The 1D subdivision on edges *E*, resulted from the MMP algorithm, induce a correct 2D subdivision inside all the faces *F* of *T*. Furthermore, each planar 2D subdivision area is star-shaped with respect to the 2D unfolded position of its corresponding pseudo-source in the same plane.

Proof. Assume that a vacuum area exists inside the triangle *t* from the induced 2D subdivision, as shown in Fig. 11. Let *p* be a point in this area. The geodesic path from the source to *p* must intersect the boundary edges of *t*. Let *q* be the intersection point. Since the edges are completely covered by wedges, point *q* must be in a wedge w_a with pseudo-source *a*. Then $q(a)$ is the geodesic distance of *q* and by assumption, *p* cannot be in the extended line from *a* to *q*. Two possibilities exist. First, *q* is inside the wedge w_a as shown in Fig. 12a. In this case, the line segment from a point (shown in green color) near *q* in line segment $\overline{aq}$ to a point (shown also in green) near *q* in segment $\overline{pq}$ contributes a shorter path than the geodesic path to *p*, a contradiction. In the second case, *q* is at the boundary of a wedge w_a as shown in Fig. 12b. In this case, the geodesic path to *p* must go through the interior of wedge w_a. Let *r* be a point on the path which is also in the interior of w_a. Then a similar shortcut exists around point *r* as shown in Fig. 12b. So it shows that any point inside triangle must be in some wedge and then the 2D subdivision is star-shaped with respect to the wedge's pseudo-source. ■

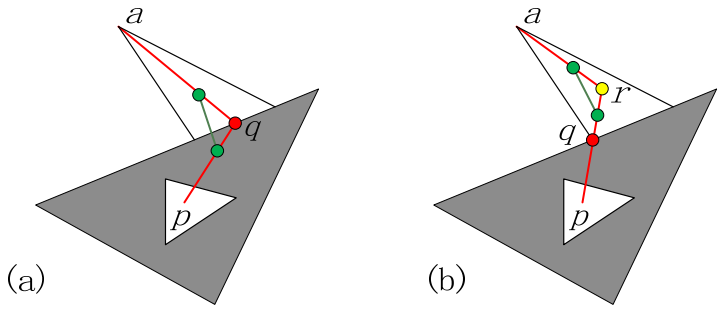


Figure 12. Proof of Property 4: (a) *q* lies inside a wedge with pseudo-vertex *a*; (b) *q* is at the boundary of a wedge.

For the illustrative 2D subdivision structure shown in Fig. 1a, it helps to provide a real example of 2D subdivision induced from 1D edge subdivision; thus the following example is in order. At the end, we will also draw an observation from this example.

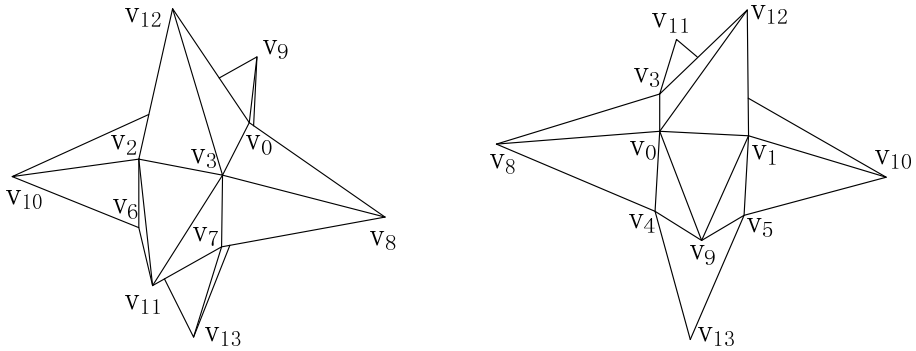


Figure 13. A triangular mesh surface model used in Example 4.

Example 4. Fig. 13 shows a mesh model consisting of 14 vertices and 24 triangles. Its geometric data is as follows.

(1) Vertex positions:
$v_0 = (0,0,0)$, $v_1 = (0,1,0)$, $v_2 = (1,1,0)$, $v_3 = (1,0,0)$, $v_4 = (0,0,1)$, $v_5 = (0,1,1)$, $v_6 = (1,1,1)$, $v_7 = (1,0,1)$, $v_8 = (0.5,-1.8,0.5)$, $v_9 = (-1.8,0.5,0.5)$, $v_{10} = (0.8,2.6,0.8)$, $v_{11} = (2.9,0.2,0.2)$, $v_{12} = (0.1,0.9,-1.5)$, $v_{13} = (0.5,0.5,2.8)$

(2) Triangle indices:
$f_0 = (v_0, v_3, v_8)$, $f_1 = (v_3, v_7, v_8)$, $f_2 = (v_7, v_4, v_8)$, $f_3 = (v_4, v_0, v_8)$, $f_4 = (v_1, v_0, v_9)$, $f_5 = (v_5, v_1, v_9)$, $f_6 = (v_4, v_5, v_9)$, $f_7 = (v_0, v_4, v_9)$, $f_8 = (v_2, v_1, v_{10})$, $f_9 = (v_6, v_2, v_{10})$, $f_{10} = (v_5, v_6, v_{10})$, $f_{11} = (v_1, v_5, v_{10})$, $f_{12} = (v_2, v_6, v_{11})$, $f_{13} = (v_6, v_7, v_{11})$, $f_{14} = (v_7, v_3, v_{11})$, $f_{15} = (v_3, v_2, v_{11})$, $f_{16} = (v_0, v_1, v_{12})$, $f_{17} = (v_1, v_2, v_{12})$, $f_{18} = (v_2, v_3, v_{12})$, $f_{19} = (v_3, v_0, v_{12})$, $f_{20} = (v_5, v_4, v_{13})$, $f_{21} = (v_6, v_5, v_{13})$, $f_{22} = (v_7, v_6, v_{13})$, $f_{23} = (v_4, v_7, v_{13})$.

The exact geodesic algorithm is run on this model with the source point locating at $v_2 = (1,1,0)$. For a better illustration, we sort and represent the wedges at each edge by a 7-tuple $(b_0, b_1, d_0, d_1, \tau, s, \sigma)$: b_0, b_1, d_0, d_1 have the same meaning as in Definition 1, σ is the geodesic distance from pseudo-vertex s to the source $p = v_2 = (1,1,0)$, τ is true (false) if the visibility wedge comes from the outside (inside) of the face. If the wedge is degenerated into a line and coincides with the edge, τ is false. The 1D edge subdivision on three edges of Face 20 is shown in Fig. 14 and the 2D subdivision structure of Face 20 is shown in Fig. 15.

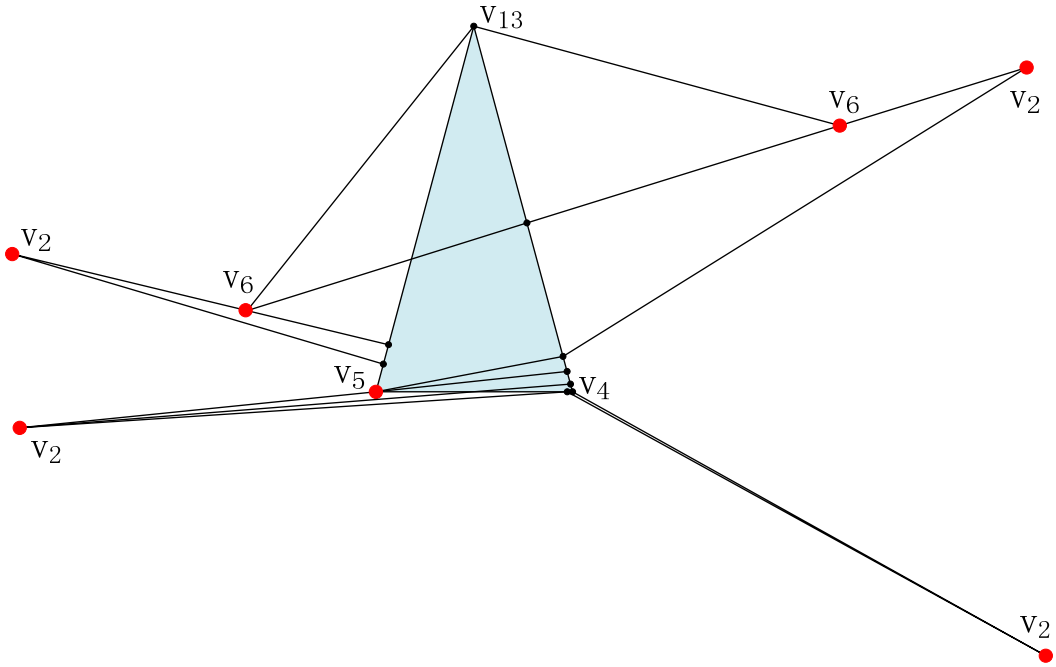


Figure 14. The 1D edge subdivision on three edges of Face 20 in the model shown in Fig. 13.

Face 20: $f_{20} = (v_5, v_4, v_{13})$

There are two wedges at edge (v_5, v_4):

$$(0.0, 0.976, 1.826, 2.799, true, v_2, 0.0)$$
$$(0.976, 1.0, 2.799, 2.778, true, v_2, 0.0)$$

There are five wedges at edge (v_4, v_{13}):

$$(0.0, 0.038, 0.0, 0.038, false, v_4, 2.778)$$
$$(0.038, 0.104, 2.816, 2.804, false, v_2, 0.0)$$
$$(0.104, 0.188, 0.978, 0.968, false, v_5, 1.826)$$
$$(0.188, 0.891, 2.794, 2.675, true, v_2, 0.0)$$
$$(0.891, 1.934, 1.675, 1.934, true, v_6, 1.0)$$

There are three wedges at edge (v_{13}, v_5):

$$(0.0, 1.690, 1.934, 0.966, true, v_6, 1.0)$$
$$(1.690, 1.790, 1.966, 1.970, true, v_2, 0.0)$$
$$(1.790, 1.934, 0.144, 0.0, false, v_5, 1.826)$$

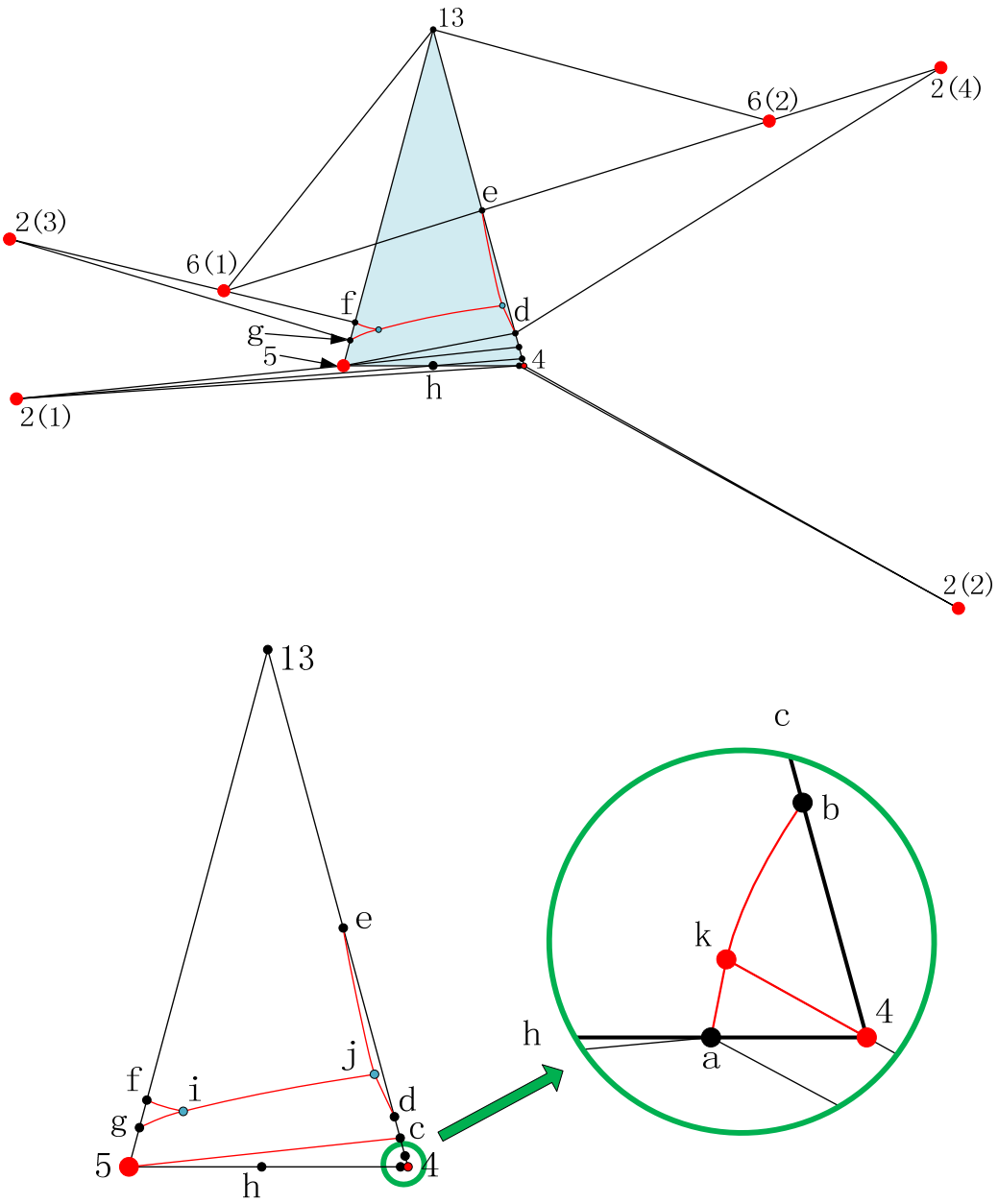


Figure 15. The 2D subdivision inside the Face 20: the vertex indices are bracketed with numbers if the index is duplicated for different locations of the same vertex.

The 2D subdivision inside the Face 20 is shown in Fig. 15. There are totally 7 subdivisions inside the Face 20 $f_{20} = (v_5, v_4, v_{13})$:

Subdivision 1: area $\{f, i, j, e, 13, f\}$, the corresponding pseudo-vertex is $6(1)$.
Subdivision 2: area $\{f, g, i, f\}$, corresponding to $2(3)$.
Subdivision 3: area $\{j, d, e, j\}$, corresponding to $2(4)$.
Subdivision 4: area $\{g, 5, c, d, j, i, g\}$, corresponding to 5.
Subdivision 5: area $\{5, h, a, k, b, c, 5\}$, corresponding to $2(1)$.

Subdivision 6: area $\{k, 4, b, k\}$, corresponding to 4.
Subdivision 7: area $\{a, 4, k, a\}$, corresponding to $2(2)$.

The meaning of the above letters and numbers is illustrated in Fig. 15. All these subdivisions are star shaped with respect to their corresponding pseudo-vertices: this demonstrates the Property 4.

Refer to Fig. 15. The line $(13, e)$ is the bisector of vertices $\{6(1), 6(2)\}$. Akin to Example 1, this gives another symmetric configuration so that the intersection of wedges $\{6(1), 13, e\}$ and $\{6(2), 13, e\}$ has an infinite number of solutions of Equation (2) in the range $(13, e)$. The other bisectors' information is as follows.

Hyperbolic curve (e, j): the bisector of pseudo-vertices $\{6(1), 2(4)\}$.
Line segment (f, i): the bisector of pseudo-vertices $\{2(3),6(1)\}$.
Hyperbolic curve (g, i): the bisector of pseudo-vertices $\{2(3),5\}$.
Hyperbolic curve (i, j): the bisector of pseudo-vertices $\{5,6(1)\}$.
Line segment $(5, c)$: the bisector of pseudo-vertices $\{5,2(1)\}$.
Line segment (a, k): the bisector of pseudo-vertices $\{2(1),2(2)\}$.
Line segment $(4, k)$: the bisector of pseudo-vertices $\{4,2(2)\}$.
Hyperbolic curve(k, b): the bisector of pseudo-vertices $\{4,2(1)\}$.

Note that after the intersection of bisectors (f, i) and (g, i), a new bisector (i, j) grows up. All above listed bisectors are trimmed properly.

Besides the Face 20, all the edge subdivisions are illustrated in the supplemental material submitted together with this paper. From the Example 4 and the supplemental material, we have the following observation.

Observation 1. In the 1D edge subdivision structure in exact geodesic computation, the number of wedges on different edges in the triangle model varies severely and the distribution of wedges on each edge is highly non-uniform.

4. An efficient solution to wedge intersection

Proper handling of wedge intersection during wedge propagation process is the crux of the MMP algorithm. In Property 1 we have proven that the intersection can lead to complicated situations: the Equation (2) can have solutions of zero, one, two or infinite. The users have to deal with these situations with many branches in coding. Here we propose a practical solution to wedge intersection. The following pseudo-code is in C++ style and the wedge w is implemented in a class with class members $(b_0, b_1, d_0, d_1, \sigma, \tau)$. All the computations below are in the local 2D plane defined by the edge that the wedge belongs to.

Algorithm: Wedge_intersection
Input. Two wedges w_1, w_2 at an edge
Output. The updated wedges at that edge

1. Set $inter.b_0 = \max(w_1.b_0, w_2.b_0)$, $inter.b_1 = \min(w_1.b_1, w_2.b_1)$.
2. **If** $inter.b_0 < inter.b_1$ //There is an overlapped range between two wedges

2.1. Let $w_1\_\sigma\_b_0$ be the distance from the pseudo-source of w_1 to the point $inter.b_0$ at the edge, and set $geo\_dist\_b_0\_w_1 = w_1.\sigma + w_1\_\sigma\_b_0$. The values $w_j\_\sigma\_b_i$, $geo\_dist\_b_i\_w_j$, $i = 0,1$, $j = 1,2$ are defined similarly.

2.2. **If** ($geo\_dist\_b_0\_w_1 < geo\_dist\_b_0\_w_2$ and $geo\_dist\_b_1\_w_1 > geo\_dist\_b_1\_w_2$) or ($geo\_dist\_b_0\_w_1 > geo\_dist\_b_0\_w_2$ and $geo\_dist\_b_1\_w_1 < geo\_dist\_b_1\_w_2$)
//exclude the cases that eq. (2) have zero or infinite solutions

2.2.1. Find the solutions of equation (2).

2.2.2. **If** there is one solution q in the range $(inter.b_0, inter.b_1)$

2.2.2.1. Update w_1 and w_2 in the way as shown in Fig. 3c.

2.2.3. **Else** //Two solutions in the range $(\text{inter.b}_0, \text{inter.b}_1)$

2.2.3.1. Update w_1 and w_2 in the way as shown in Fig. 10.

2.3. **Else** // Eq. (2) have zero or infinite solutions

2.3.1. Compute the values of A, B, C in equation (2)

2.3.2. **If** $A = B = C = 0$ // Eq. (2) have infinite solutions

2.3.2.1. Delete w_2;

2.3.3. **Else** // Eq. (2) have zero solution

2.3.3.1. **If** $geo\_dist\_b_0\_w_1 < geo\_dist\_b_0\_w_2$

2.3.3.1.1. Delete w_2;

2.3.3.2. **Else**

2.3.3.2.1. Delete w_1;

5. Benchmark test model

Result verification is critical for the coding of a geodesic computation algorithm. When finishing code programming, the users need test the code on diverse models to examine the robustness and stability of the code. However, without a benchmark model that can output correct geodesics for comparison, the users cannot know whether or not their computed geodesics are correct.

Here we propose a benchmark model that utilizes the visibility graphs [19, 20]. Refer to Figs. 16a and 16b. Given a set *S* of disjoint polygonal obstacles in a plane, the visibility graph *VG*(*S*) of *S* is defined as follows: its nodes are the vertices of *S* and there is a line between vertices *v* and *w* if they can see each other. Given two points *p*, *q* and *S* with *n* vertices, $VG(S \cup \{p, q\})$ can be constructed in $O(n^2 \log n)$ [19]. Given the visibility graph, the shortest path between *p*, *q* can be found in $O(n \log n)$ [21].

We use the shortest path computed from the visibility graph as the ground truth to construct the benchmark model. Let the obstacles *S* be bounded in a rectangle *R* that is large enough. A constrained Delaunay triangulation (CDT) [22] is constructed with *p*, *q* and the boundaries of *S* and *R*. (ref. Fig. 16c; actually any constrained triangulation can be applied here: we choose CDT due to its nice property of maxmin angle.) At the place of each obstacle, a polygonal cylinder is lifted and sewed with the planar triangulation along the obstacle's boundary (ref. Fig. 16d). It is readily seen that if the height of cylinders is large enough, the geodesic between *p* and *q* will not climb the cylinders; instead, the geodesic will go around the boundary of obstacles (ref. Fig. 16e

and 16f). For generalization, we use both convex and concave polygon obstacles in the benchmark model. By using mesh refinement, the complexity of the benchmark model can be controlled by the user at wills.

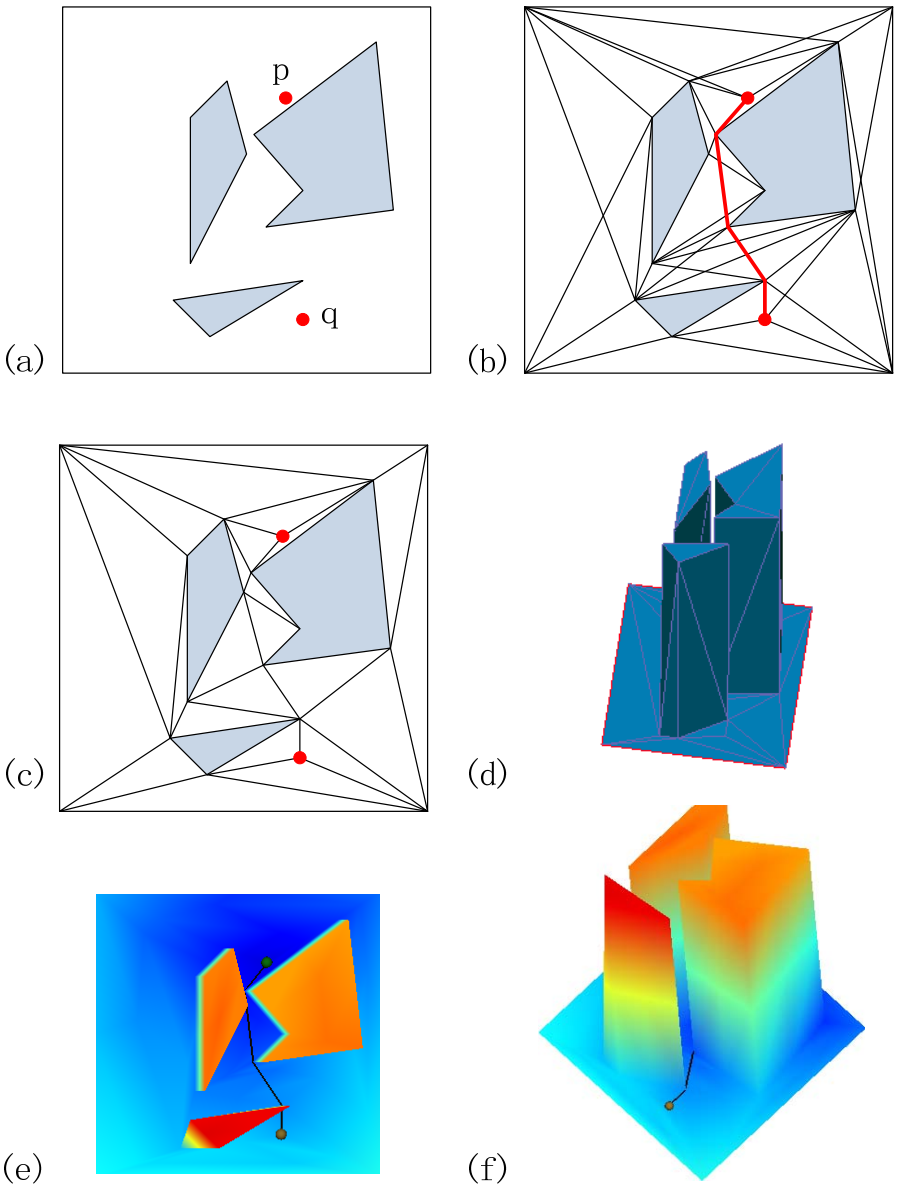


Figure 16. The benchmark model utilizing visibility graphs: (a) A set of obstacles and two points *p*, *q* inside a rectangle; (b) The visibility graph of (a) and the shortest path between *p* and *q*; (c) The constrained Delaunay triangulation of (a); (d) The benchmark model made by lifting each polygonal obstacle as a polygonal cylinder whose height is large enough; (e, f) The geodesic between *p* and *q* in the benchmark model, showing from different views; the distance field on the model is colored by one-to-one mapping the geodesic distance to an index color map.

6. Experimental results

We have implemented the exact geodesic computation algorithm for triangular mesh surfaces, on

the platform of Visual C++.net and with Microsoft Windows XP operating system. Some test models are shown in Figs. 17 and 18. The performance data is given in Tables 1 and 2. All the tests are performed with an off-the-shelf PC with Intel(R) Core(TM)2 Duo CPU T7700, running at 2.40GHz with 2.00GB RAM. The authors of the work [17] have published an implementation *GeodesicLib* of their method. The *GeodesicLib* restrictedly output geodesic paths in a text file without wedge information. We compare our implementation to *GeodesicLib* side by side and the results are the same, subject to the floating-point arithmetic.

For revealing more properties of exact geodesic computation, we also used our implementation to collect wedge information. We choose the test models according to the following criteria. First, as shown in Fig. 17, the models of sphere, eight and teapot models are presented in two formats: non-uniform and uniform. The uniformness is judged by the maximum of minimal internal angles of all triangles in the model. The non-uniform models have irregular triangle shapes that are also irregularly distributed in the models. We apply the feature-sensitive remeshing technique in [23] to obtain uniform models with almost the same number of triangles. For each model, the number $w(e)$ of wedges at each edge is regarded as a random variable. In Table 1, the mean $\overline{w(e)}$ of $w(e)$ (i.e., the average number of wedges per edge), the standard deviation $\sigma[w(e)]$, the maximal number of wedges at one edge $max_e w(e)$, and the running time are listed. From these data, we have obtained the following observation:

Observation 2. Compared to non-uniform models, uniform models have statistically less wedges on the edges, whose distribution of wedges on each edge is more concentrated near the mean, and the running time is generally smaller.

Secondly, as shown in Fig. 18, we test the code on two large models. All the models shown in Figs. 17 and 18 are normalized in a $1.0 \times 1.0 \times 1.0$ cube. So models with larger number of edges have smaller edge lengths than those on models with smaller number of edges. In the case of existence of very short edges, degeneration frequently happens due to floating-point arithmetic. In our code, all degenerate cases are handled using the technique in [24]. For each model, the length of each edge is denoted by $l(e)$ and is regarded as a random variable. In Table 2, the mean $\overline{l(e)}$ of $l(e)$ (i.e., the average edge length), the standard deviation $\sigma[l(e)]$ (which gives another measure of the uniformness of the model), the maximal edge length $max_e l(e)$ in the model, the minimal edge length $min_e l(e)$ in the model, the average number of wedges per unit edge length $\overline{w(e)}/\overline{l(e)}$, are listed. From these data, we have got our final observation:

Observation 3. If floating-point arithmetic is used, the average number of wedges per unit edge length is scale-dependent, i.e., it is inversely proportional to the average edge length of the model.

7. Conclusions

In this paper, we have reported several interesting properties and observations of exact geodesic computation on triangular mesh surfaces. A benchmark test model with ground truth of exact geodesic paths is also presented for test of the code. Particular real examples and experimental results are depicted, demonstrating that these properties may help users gain better understanding

of the classical MMP algorithm from a new perspective. The presented properties are also helpful in classifying several ambiguous cases that may crash the generic geometric algorithm.

Reference

[1] Bajaj C, Kim M. Compliant motion planning with geometric models. *Proceedings of the 3rd Annual Symposium on Computational Geometry*, pp. 171-180, 1987.

[2] Wilmarth SA, Amato NM, Stiller PF. Motion planning for a rigid body using random networks on the medial axis of the free space. *Proceedings of the 15th Annual Symposium on Computational Geometry*, pp. 173-180, 1999.

[3] Hilaga M, Shinagawa Y, Kohmura T, Kunii TL. Topology matching for fully automatic similarity estimation of 3d shapes. *In Proc. of ACM SIGGRAPH'01*, pp. 203-212, 2001.

[4] Hamza AB, Krim H. Geodesic matching of triangulated surfaces. *IEEE Transactions on Image Processing*, 15(8): 2249-2258, 2006.

[5] Caselles V, Kimmel R, Sapiro G. Geodesic Active Contours. *International Journal of Computer Vision*, 22(1): 61-79, 1997.

[6] Goldenberg R, Kimmel R, Rivlin E, Rudzsky M. Fast geodesic active contours. *IEEE Transactions on Image Processing*, 10(10): 1467-1475, 2001.

[7] Paragios N, Deriche R. Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects. *IEEE Transactions on Pattern Analysis and Machine*, 22(3): 266-280, 2000.

[8] Kimmel R., Sethian JA. Computing geodesic paths on manifolds. *Proc. of National Academy of Sci.* 95(15): 8431-8435, 1998.

[9] Kirsanov D. *Minimal Discrete Curves and Surfaces*. PhD thesis, Applied Math., Harvard University, 2004.

[10] Lanthier M, Maheshwari A, Sack JR. Approximating weighted shortest paths on polyhedral surfaces. *Proceedings of the 13th Annual Symposium on Computational Geometry*, pp. 274-283, 1997.

[11] Kanai T, Suzuki H. Approximate shortest path on a polyhedral surface and its applications. *Computer-Aided Design*, 33(11): 801-811, 2001.

[12] Mitchell J. Geometric shortest paths and network optimization. *In Handbook of Computational Geometry*. JR. Sack, J. Urrutia, eds., Elsevier Science, pp. 633-702, 2000.

[13] Mitchell JSB, Mount DM, Papadimitriou CH. The discrete geodesic problem. *SIAM J. of Computing*, 16(4): 647-668, 1987.

[14] Chen J, Han Y. Shortest paths on a polyhedron: part I: computing shortest paths. *Int. J. Comp. Geom. & Appl.* 6(2): 127-144, 1996.

[15] Kaneva B, o'Rourke J. An implementation of Chen & Han's shortest paths algorithm. *In Proc. of the 12th Canadian Conf. on Comput. Geom.*, pp. 139-146, 2000.

[16] Kapoor S. Efficient computation of geodesic shortest paths. *In Proc. 32nd Annual ACM Symp. Theory Comput.,* pp. 770-779, 1999.

[17] Surazhsky V, Surazhsky T, Kirsanov D, Gortler S, Hoppe H. Fast exact and approximate geodesics on meshes. *ACM SIGGRAPH'05*, pp. 553-560, 2005.

[18] do Carmo M.P. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.
[19] Lee D.T.. Proximity and reachability in the plane. *Report R-831*, Dept. Elect. Engrg., Univ. Illinois, Urbana, IL, 1978.
[20] Mitchell J. Shortest paths among obstacles in the plane. *In Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pp. 308-317, 1993.
[21] Hershberger J, Suri S. Efficient computation of Euclidean shortest paths in the plane. *In Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pp. 508-517, 1993.
[22] Lee D.T., Lin A.K. Generalized Delaunay triangulation for planar graphs. *Discrete Comput. Geom.* 1: 201-217, 1986.
[23] Lai Y, Zhou Q.Y., Hu S.M., Wallner J, Pottmann H. Robust Feature Classification and Editing. *IEEE Transaction on Visualization and Computer Graphics*, Vol. 13, pp. 34-45, 2007.
[24] Liu Y.J., Zhou Q.Y., Hu S.M. Handling degenerate cases in exact geodesic computation on triangle meshes. *The Visual Computer*, 23(9-11): 661-668, 2007.